推荐系统

2018.11.17

协同过滤

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

用 Pearson 比较多

相似度计算

- 欧几里德距离（Euclidean Distance）

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

- 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）

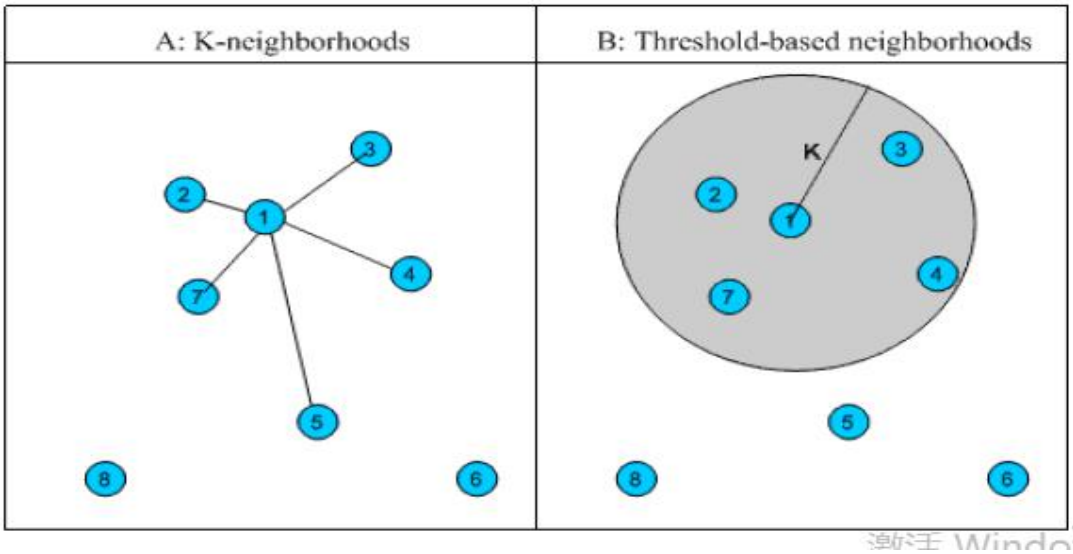
$$T(x,y)=\frac{x\bullet y}{\|x\|^2\times\|y\|^2}=\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

邻居的选择

A. 固定数量的邻居

B. 基于相似度门槛的邻居

最好用 B



一、协同过滤

基于用户的协同过滤

协同过滤是基于统计的，需要两两计算，计算量很大

在用户中找相似度，A,B,C 写成向量，计算相似度，[1,0,1,0];[0,1,0,0];[1,0,1,1].

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	√		√	√

问题：

用户评分矩阵 R 稀疏；

需要推断矩阵中空格处的值

对于新用户，很难找到邻居用户，冷启动问题

对于一个物品，所有最近邻居都在其上没有多少打分，那么这个物品永远不会被推荐给邻居用户

方案：

相似度计算 最好用皮尔逊相似度

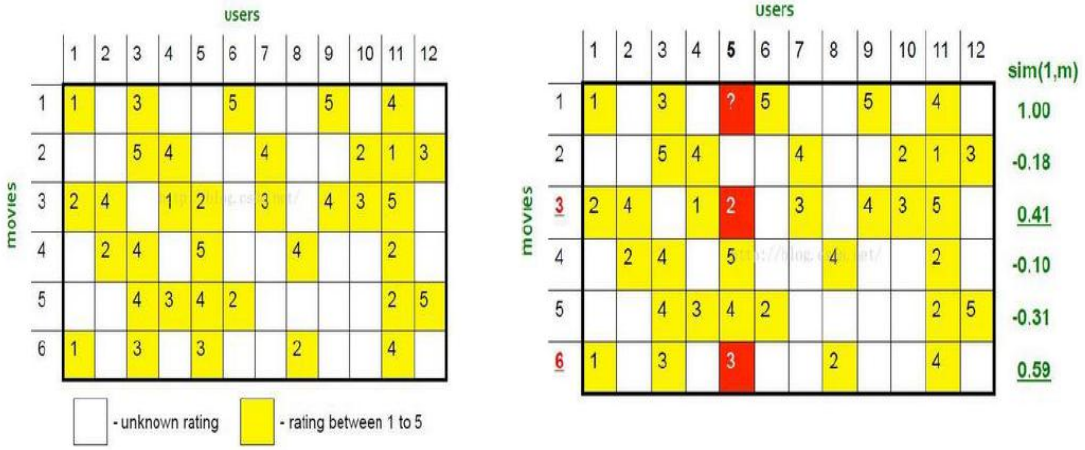
考虑共同打分物品的数目，如乘上 min(n,N)/N　n:共同打分数 N：指定阈值
对打分进行归一化处理
设置一个相似度阈值，用阈值做一个半径

基于用户的协同过滤，这个算法不太常用，表太大，用户太多，买的东西相对于所有物品很少，人的喜好容易变化

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

计算性能高，物品好打标签，能用的信息多
e.g.



$$r\_51 = (0.41*2 + 0.59*3)/(0.41 + 0.59) = 2.6$$

Sim(1,m)分别是 1 和 1 的相似度，1 和 2 的相似度...
找出相似度高的 3 号和 6 号
相似度作为权重参数
设定一个阈值为 3.5 因为 2.6<3.5 所以就不推荐了

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

物品冷启动问题

文本分析
主题模型
打标签
推荐排行榜单

两种方法比较

	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

应用场景

基于用户的推荐
实时新闻、突然情况

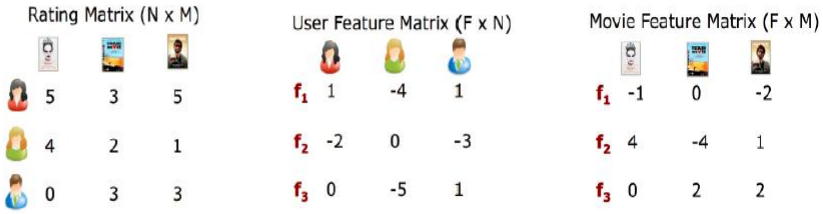
基于物品的推荐
图书、电子商务、电影

二、隐语义模型

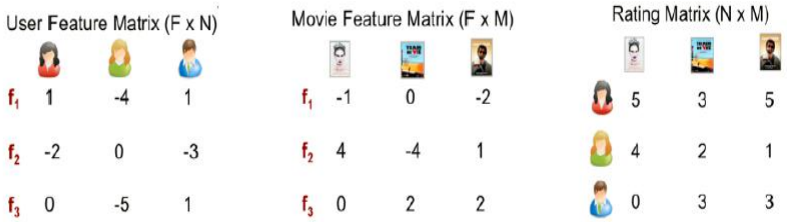
用户和物品之间有着隐含的关系
隐含因子让计算机理解就好
将用户和物品通过中间隐含因子联系起来

隐语义模型

分解



组合



N 个用户，M 个商品

矩阵分解 N F　F M，F 个隐藏因子与 SVD 差不多，但是比 SVD 简单

隐藏因子要有价值

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
user 1	R11	R12	R13	R14
user 2	R21	R22	R23	R24
user 3	R31	R32	R33	R34

R

=

	class 1	class 2	class 3
user 1	P11	P12	P13
user 2	P21	P22	P23
user 3	P31	P32	P33

P

×

	item 1	item 2	item 3	item 4
class 1	Q11	Q12	Q13	Q14
class 2	Q21	Q22	Q23	Q24
class 3	Q31	Q32	Q33	Q34

Q

P,Q 是参数，所以要放在正则化惩罚项中

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：

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

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

协同过滤：统计建模思想
隐语义：机器学习思想

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，则为 r_ui=1，负样本则为 r_ui=0

隐语义模型参数选择

隐特征的个数 F，通常 F=100
学习率 alpha,别太大　0.01 交叉验证
正则化参数 lambada，别太大　0.01　0.1　交叉验证
正负样本比例　负样本/正样本比例 ratio　5-10

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%

三、推荐系统评估标准

✔ 评估标准：

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

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

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

R，是推荐上的列表
T，是行为上的列表，比如 test

评估标准：

覆盖率：

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

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

多样性：

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

覆盖率可以按照覆盖种类算，也可按照熵值来算，熵值越大，覆盖率越大尽可能广的推荐东西