机器学习-协同过滤

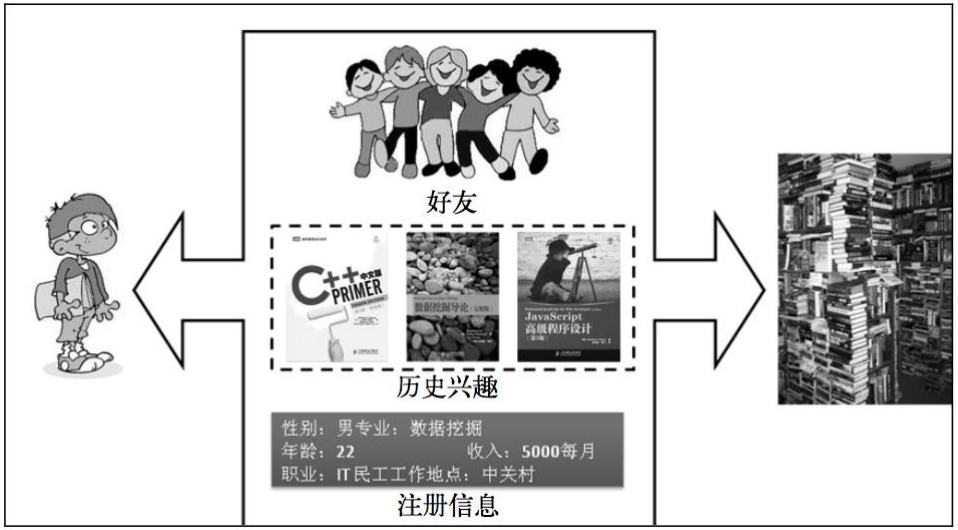
目录 Contents

协同过滤要解决的问题

协同过滤算法主要用于推荐系统，推荐系统是信息过载所采用的措施，面对海量的数据信息，从中快速推荐出符合用户特点的物品。一些人的“选择恐惧症”、没有明确需求的人。

解决如何从大量信息中找到自己感兴趣的信息。

解决如何让自己生产的信息脱颖而出，受到大众的喜爱。



协同过滤要解决的问题

	I1	I2	I3	I4	I5	I6	I7	I8	I9
U1		1		5	3			2	
U2			2				5		4
U3	3	5		2		4			
U4			3		5		2		1
U5		2		1	2			5	
U6	5		3			5		1	2

推荐数据的准备

用户ID、物品ID、偏好值

偏好值就是用户对物品的喜爱程度，推荐系统所做的事就是根据这些数据为用户推荐他还没有见过的物品，并且猜测这个物品用户喜欢的概率比较大。

用户ID和物品ID一般通过系统的业务数据库就可以获得，偏好值的采集一般会有很多办法，比如评分、投票、转发、保存书签、页面停留时间等等，然后系统根据用户的这些行为流水，采取减噪、归一化、加权等方法综合给出偏好值。一般不同的业务系统给出偏好值的计算方法不一样。

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
点击流（查看）	隐式	一组用户的点击，用户对物品感兴趣，需要进行分析，得到偏好	用户的点击一定程度上反映了用户的注意力，所以它也可以从一定程度上反映用户的喜好。
页面停留时间	隐式	一组时间信息，噪音大，需要进行去噪，分析，得到偏好	用户的页面停留时间一定程度上反映了用户的注意力和喜好，但噪音偏大，不好利用。
购买	隐式	布尔量化的偏好，取值是 0 或 1	用户的购买是很明确的说明这个项目它感兴趣。

协同过滤的意思

协同是什么意思？

过滤是什么意思？

相似度度量

皮尔逊相关系数是介于1到-1之间的数，他衡量两个一一对应的序列之间的线性相关性。也就是两个序列一起增大或者一起减小的可能性。两个序列正相关值就趋近1，否者趋近于0。

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

数学含义：两个序列协方差与二者方差乘积的比值

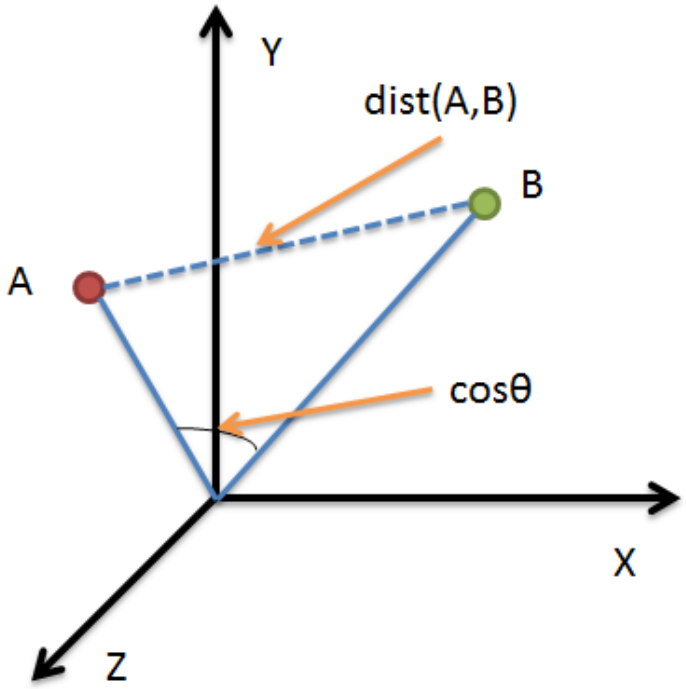
如果比较两个人的相似度，那么他们所有共同评价过的物品可以看做两个人的特征序列，这两个特征序列的相似度就可以用皮尔逊相关系数去衡量。物品的相似度比较也是如此。

皮尔逊对于稀疏矩阵表现不好，可以通过引入权重进行优化。

相似度度量

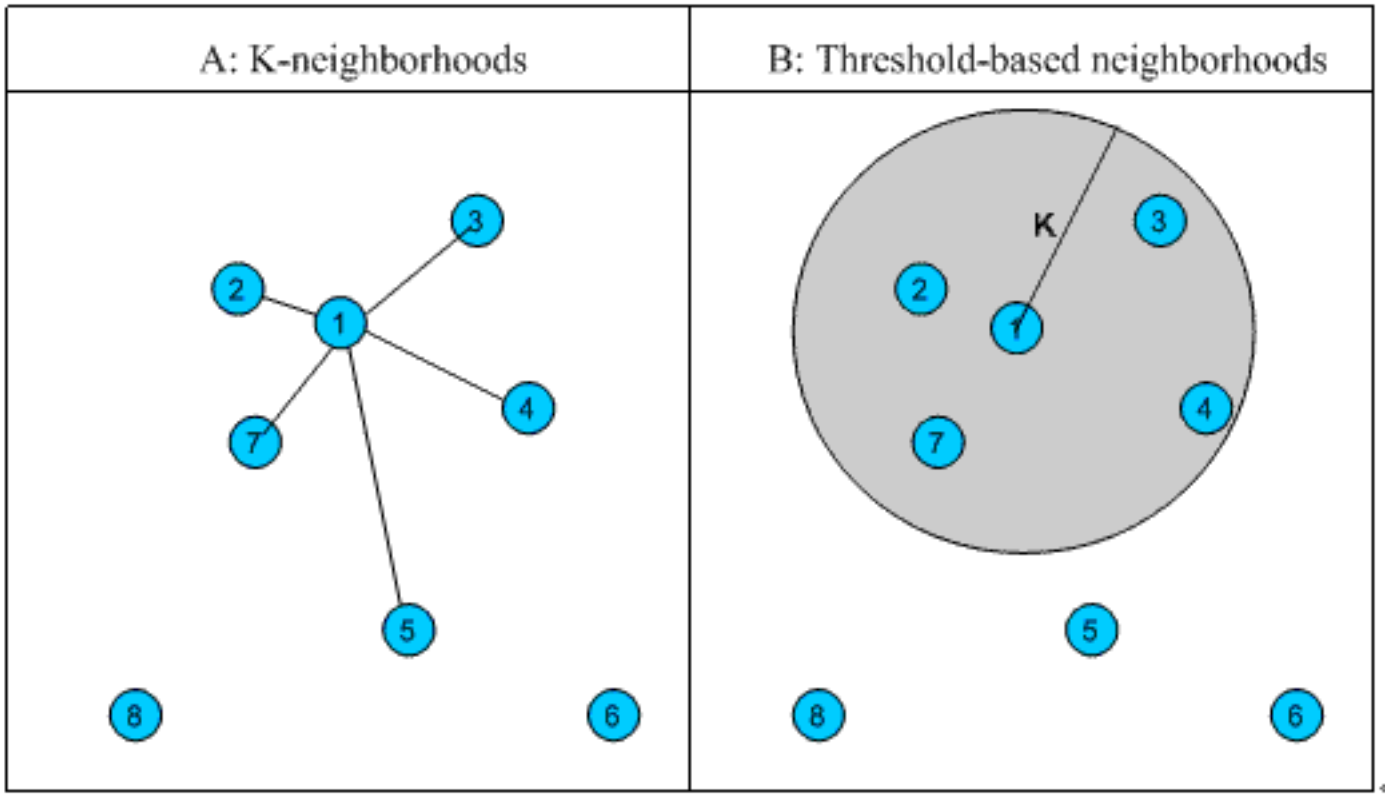
欧几里德距离，这个距离就是平时我们理解的距离，如果是两个平面上的点，也就是（X1，Y1），和（X2，Y2），那这俩点距离就是√((x1-x2)^2+(y1-y2)^2)，如果是三维空间中呢？√((x1-x2)^2+(y1-y2)^2+(z1-z2)^2；推广到高维空间公式就以此类推。可以看出，欧几里德距离真的是数学加减乘除算出来的距离，因此这就是只能用于连续型变量的原因。

$$d(x,y)=\sqrt{(\sum(x_i-y_i)^2)}$$



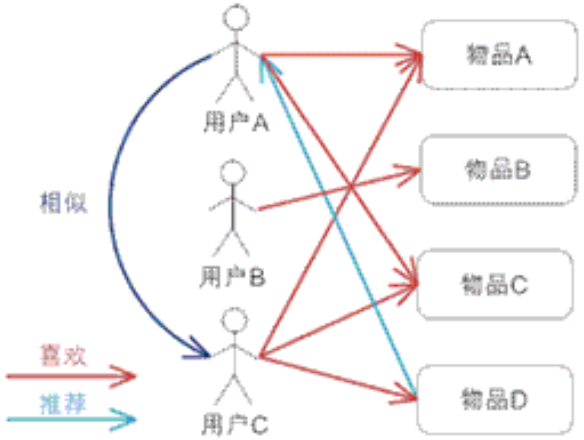
协同过滤要解决的问题

有了相似度的比较，那么比较多少个用户或者物品为好呢？一般会有基于**固定大小的邻域**以及基于**阈值的领域**。具体的数值一般是通过对模型的评比分数进行调整优化。



基于用户的CF

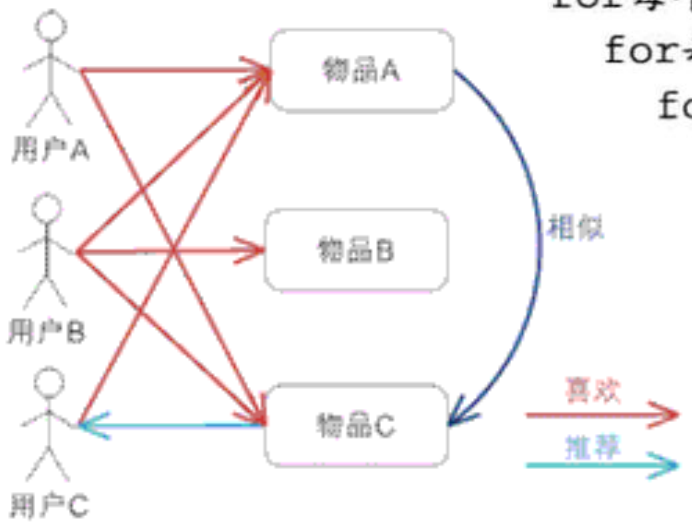
用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√



for每个其他用户w
  计算用户u和用户w的相似度s
  按相似度排序后，将位置靠前的用户作为邻域n
for（n中用户有偏好，而u中用户无偏好的）每个物品i
    for（n中用户对i有偏好的）每个其他用户v
  计算用户u和用户v的相似度s
    按权重s将v对i的偏好并入平均值

基于物品的CF

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



for每个物品**i**
  for每个其他物品**j**
    for对**i**和**j**均有偏好的每个用户**u**
      将物品对（**i**与**j**）间的偏好值差异加入**u**的偏好

for用户**u**未表达过偏好的每个物品**i**
  for用户**u**表达过偏好的每个物品**j**
    找到**j**与**i**之间的平均偏好值差异
    添加该差异到**u**对**j**的偏好值
    添加其至平均值
Return值最高的物品（按平均差异排序）

ALS优化

V / U	v1	v2	v3	v4	v5	v6	v7
u1	3	?	5	5	?	4	?
u2	5	?	?	4	?	?	?
u3	5	5	?	?	5	5	?
u4	?	4	4	?	2	5	?
u5	?	5	?	5	3	3	?

既然要打分，那么?

	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：用户是不是有一些隐含的特征?

Q：物品是不是有一些隐含的特征?

如果Match?

ALS优化

V / U	v1	v2	v3	v4	v5	v6	v7
u1	3	?	5	5	?	4	?
u2	5	?	?	4	?	?	?
u3	5	5	?	?	5	5	?
u4	?	4	4	?	2	5	?
u5	?	5	?	5	3	3	?

既然要打分，那么？

损失函数：吉洪诺夫正则化

$$\min_{x_*,y_*} \sum_{u,i \text{ is known}} (r_{ui} - x_u^T y_i)^2 \quad \Rightarrow \quad \min_{x_*,y_*} L(X,Y) = \min_{x_*,y_*} \sum_{u,i \text{ is known}} (r_{ui} - x_u^T y_i)^2 + \lambda(|x_u|^2 + |y_i|^2)$$

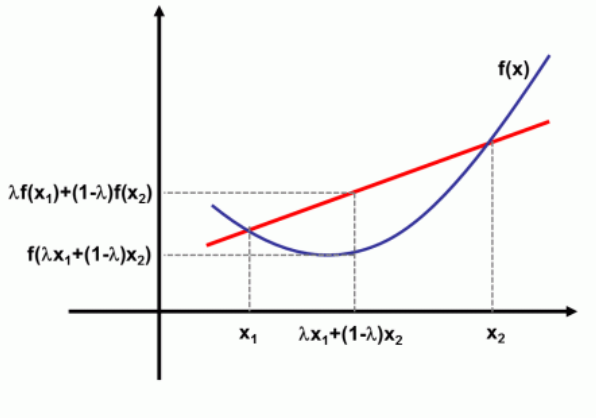
$$\frac{\partial C}{\partial v_j} = \frac{\partial}{\partial v_j}\left[\sum_{i=1}^{m}\left[(a_{i,j} - (u_i^{(0)})^T v_j)^2 + \lambda(\left\|u_i^{(0)}\right\|^2 + \left\|v_j\right\|^2)\right]\right]$$

$$= \sum_{i=1}^{m}\left[2(a_{i,j} - (u_i^{(0)})^T v_j)(-u_i^{(0)}) + 2\lambda v_j\right]$$

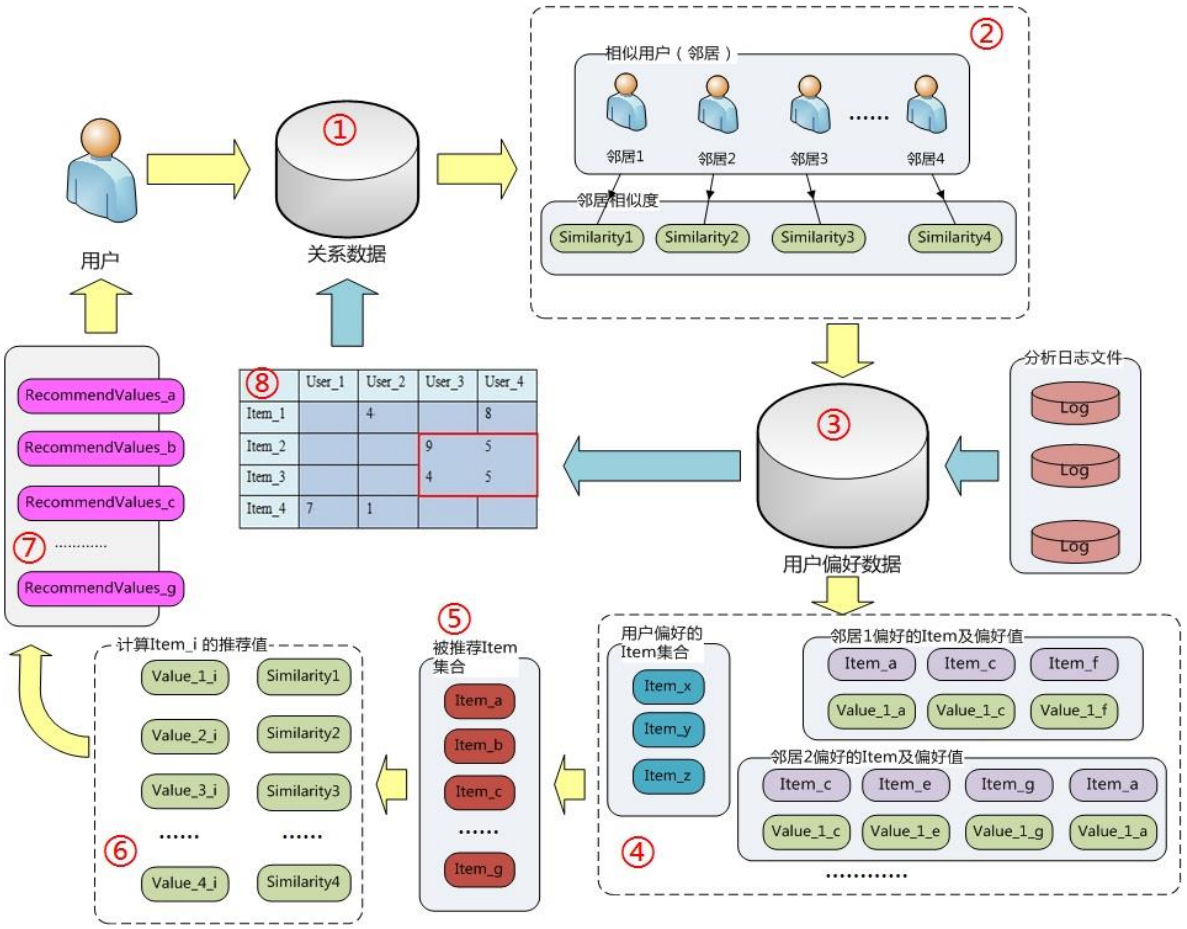
$$= 2\sum_{i=1}^{m}\left[((u_i^{(0)})^T u_i^{(0)} + \lambda)v_j - a_{i,j}u_i^{(0)}\right]$$

$$\frac{\partial C}{\partial v_j} = 0$$

损失函数是一个凸函数



协同过滤推荐架构



①. 查询的是与该用户相似的用户，所以一来直接查了关系数据源。以及相似用户与该用户的相似度。

②. 对数据集进行优化，得到相似用户和相似度。

③. 查询关系数据源，得到相似用户即邻居偏好过的物品；如步骤④；图中由于空间小，没有把所有邻居的偏好关系都列出来，用……表示。其次还要得到该用户偏好过的物品集合。

④. 被推荐的Item集合是由该用户的所有邻居的偏好过的物品的并集，同时再去掉该用户自己偏好过的物品。作用就是得到你的相似用户喜欢的物品，而你还没喜欢过的。

⑤. 集合优化同基于物品的协同过滤算法的步骤②。

⑥. 也是对应类似的，依次计算被推荐集合中Item_i 的推荐值，计算的方式略有不同，Value_1_i表示邻居1对，Item_i的偏好值，乘以该用户与邻居1的相似度 Similarity1；若某个邻居对Item_i偏好过，就重复上述运算，然后取平均值；得到Item_i的推荐值。

⑦、⑧. 与上一个算法的最后两部完全类似，只是步骤 ⑧你竖着看，判断两个用户相似的法子和判断两个物品相似的法子一样。