链滴

数据挖掘算法初窥门庭--协同过滤

作者：Zing
原文链接：https://ld246.com/article/1457699957238
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

协同过滤(CF)有两种base:

- Memory-based:
- Model-based:
  - 用户一般会喜欢与自己喜欢物品相似的物品。（item-based）
  - 用户一般会喜欢与自己相似的其他用户喜欢的物品。（user-based）

##基于用户的协同过滤

一些网购网站的“看了该物品的用户还看了以下物品”，采用的是user-based的思想。

###算法过程

- 找到与目标用户兴趣相似的用户集合

通常使用Jaccard或余弦相似度计算两个用户之间的相似度。

为了一下子计算所有用户之间的相似度：建立“物品-用户”的倒排，然后建立“用户-用户矩阵”，阵中值为用户喜欢相同物品的个数。

最后用余弦相似度计算用户的相似度：w(A,B)=A和B喜欢相同物品的个数/（根号）(A喜欢物品个数*喜欢物品个数)

- 推荐物品

从上述相似用户矩阵中，找到和目标用户u最相似的k个用户，将这k个用户喜欢的物品全部提取出来并去除u已经喜欢的物品，计算u到每个候选物品的感兴趣程度。

基本算法是：p(u,i)=（用户u和v的相似程度*用户v对物品i的感兴趣程度）对每个喜欢i且和用户u相的用户v求和。

###特点:

- 算法过程简单，结果易于解释
- 数据稀疏，导致算法很难需找偏好相似的用户。
- 算法扩展性差，若用户及物品数量增加，导致矩阵变大，复杂度高

##基于物品的协同过滤

网易云音乐采用的就是item-based，会根据你听过的音乐，推荐相似的音乐。

###算法过程

- 计算物品之间的相似度

一般采用的方法有：基于余弦的相似度、基于关联的相似度、调整余弦相似度（考虑打分）

- 预测值计算

一般采用的方法有：加权求和（已打分物品分数*和目标物品相似度）、回归

###特点:

- Item-based算法预测结果要比user-based算法质量高一些

- 由于Item-based可以离线计算物品的相似度
- 使用数据的小部分子集也可以得到高质量的结果

原文链接：数据挖掘算法初窥门庭--协同过滤