**用户行为数据简介**

用户数据行为的存在形式——日志。

两种用户行为：

显性反馈行为：明确的表示喜欢/不喜欢，比如评分系统

隐形反馈行为：不能明确反应用户喜好，比如页面浏览行为

本章中，以无上下文信息的隐形反馈数据集为例，也就是每一条行为记录仅仅包含用户ID和物品id，来分析如何利用用户行为数据。

**用户行为分析**

**用户活跃度和物品流行度的分布**

幂律分布/长尾分布/二八定律/Zipf分布



**用户流行度和物品流行度的关系**

用户越活跃，越倾向于浏览冷门的物品

**基于邻域的算法**

**基于用户的协同过滤算法**

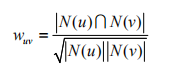
1. 基础算法
2. 找到和目标用户兴趣相似的用户集合
3. 找到这个集合中用户喜欢的，且目标用户没有听说过的物品推荐给目标用户

用行为的相似度代替兴趣的相似度，N(u)表示用户u曾经有过正反馈的物品集合。

Jaccard相似度：



余弦相似度：



train, test：字典形式，key是用户，value是临幸过的物品

实现余弦相似度的代码：

初级代码：遍历每对用户，计算相似度

复杂！O(n^2)

进阶代码：建立倒序表(新字典物品-用户)；扫描倒序表中的元素，每次扫描遍历用户对，建立用户爱好(字典)数量和用户对(二维字典)爱好数量；遍历字典求相似度。

用户u对物品i的感兴趣程度：

从最相似的K个用户中，选出对i有过行为的那些。

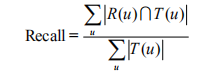
推荐的代码：

给定用户、训练集、用户相似度矩阵，遍历用户没有搞过的物品，计算该用户的潜在爱好。

准确率和召回率的计算代码：

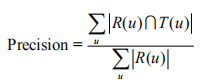
召回率：预测准了的物品/用户喜欢的全部物品

输入为train、test、N，遍历用户集合(要求是在训练集中出现的用户集合)，计算



准确率：预测准了的物品/预测的全部物品

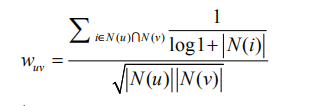
输入为train、test、N，遍历用户集合(要求是在训练集中出现的用户集合)，计算



**用户相似度计算的改进**

两个用户对冷门物品采取过同样的行为更能说明他们兴趣的相似度。

用户相似度重写为：



N(i)这个符号应该是排版或者拼写错误，它表示的是第i个物品有多少人用。该公式惩罚了用户u和用户v共同兴趣列表中热门物品对他们相似度的影响。

代码只需要对相似度计算的部分修改即可。

**基于物品的协同过滤算法**

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

用品相似度的定义：



归一化之后的定义



不同于之前的建立用户二维字典，这里建立物品二维字典。同样遍历训练矩阵，生成物品对共现次数(对应于分子)和物品共现次数(对应于分母)。书里给出的代码应该是有一点问题。

尽管在计算过程中没有利用任何内容属性，但计算结果却可以体现出相似度，这正是它和基于内容的推荐的区别。推荐算法的原则是，选择用户最喜欢的若干个物品，和用户历史上感兴趣的物品越相似的集合，越可能在用户的推荐列表中取得较高的排名。

代码的实现逻辑是，给定训练集、用户id、相似度矩阵w和比较数量k，对于用户临幸过的每一个物品，找出和它最相似的K个物品，累加相似度，最后返回推荐分数。

这个算法的好处是，可以提供解释，它不是一个 黑箱，我们很明确的知道，每一个推荐是从哪里来的。

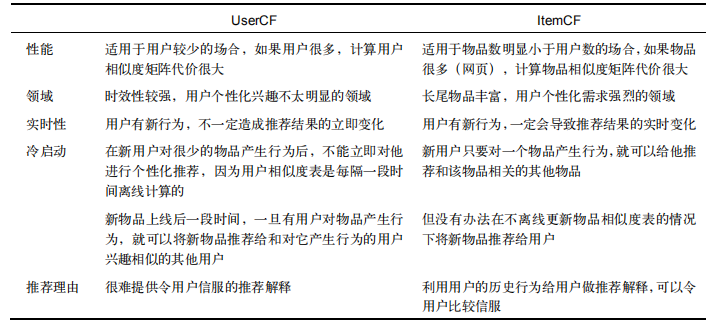
对于这种算法的修正：惩罚活跃用户/归一化物品相似度

**比较UserCF和ItemCF:**

UserCF的推荐结果着重于反映和用户兴趣相似的小群体的热点，而ItemCF的推荐结果着重于维系用户的历史兴趣。换句话说，UserCF的推荐更社会化，反映了用户所在的小型兴趣群体中物品的热门程度，而ItemCF的推荐更加个性化，反映了用户自己的兴趣传承。

UserCF的应用：新闻【侧重实时性和热门程度】

ItemCF的应用：图书，电商，电影【侧重个性推荐和个人爱好】



**哈利波特问题：**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。这个时候，仅仅靠用户行为数据是不能解决这个问题的，因为用户的行为表示这种物品之间应该相似度很高。此时，我们只能依靠引入物品的内容数据解决这个问题，比如对不同领域的物品降低权重等。这些就不是协同过滤讨论的范畴了。

**隐语义模型**

**基础算法**

通过隐含特征联系用户兴趣和物品，基于用户行为统计进行自动聚类。

LFM技术：



度量了用户u的兴趣和第k个隐类的关系，而度量了第k个隐类和物品i之间的关系。

——如何计算这两个参数？通过数据集，类似于机器学习中的“参数”。

在隐形反馈数据集上，我们只有用户-物品这样的正样本，而没有负样本。对负样本采样的原则：

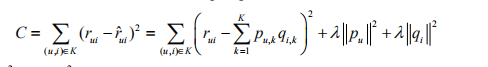
1.正负样本平衡

2.采样时，偏重采样热门的产品

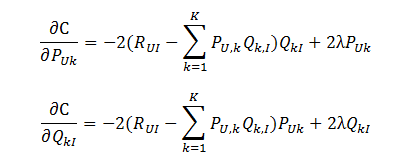
采样过程代码：

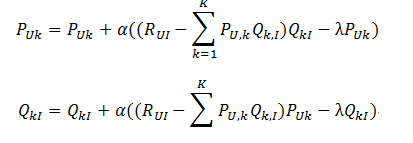
采负样本个数和正样本个数相同，每次从一个不平衡的随机池中选出一个样本，如果它是负样本，则丢开。

选择参数时，优化下面的损失函数：



使用随机梯度下降法。





带入公式即可得到最速下降的代码。

**LFM和基于邻域的方法的比较**

理论基础：LFM有较好的理论基础，邻域方法只是一种统计方法

空间复杂度：LFM算法的空间复杂度是O(F\*(M+N))，相对比较优秀

时间复杂度：LFM比较复杂

实时能力：LFM计算较复杂，实时性不太好，不太适合物品数非常大的系统

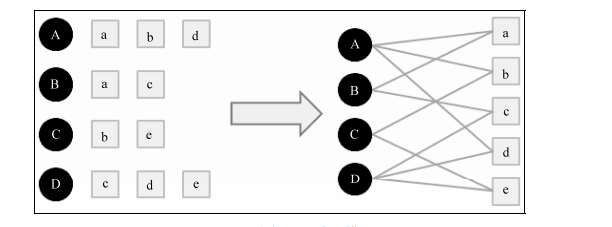
推荐解释：ItemCF以利用用户的历史行为解释推荐结果。LFM无法提供这样的解释

**基于图的模型**

**用户行为数据的二分图表示**

基于图的模型可以看作基于邻域模型的扩展。

就是下面这个图表示的意思。



**基于图的推荐算法**

推荐物品的任务转化为度量用户顶点*vu*和与*vu*没有边直接相连的物品节点在图上的相关性。

而相关性高的一对顶点一般具有如下特征：

 两个顶点之间有很多路径相连；

 连接两个顶点之间的路径长度都比较短；

 连接两个顶点之间的路径不会经过出度比较大的顶点。

随机游走算法公式：

