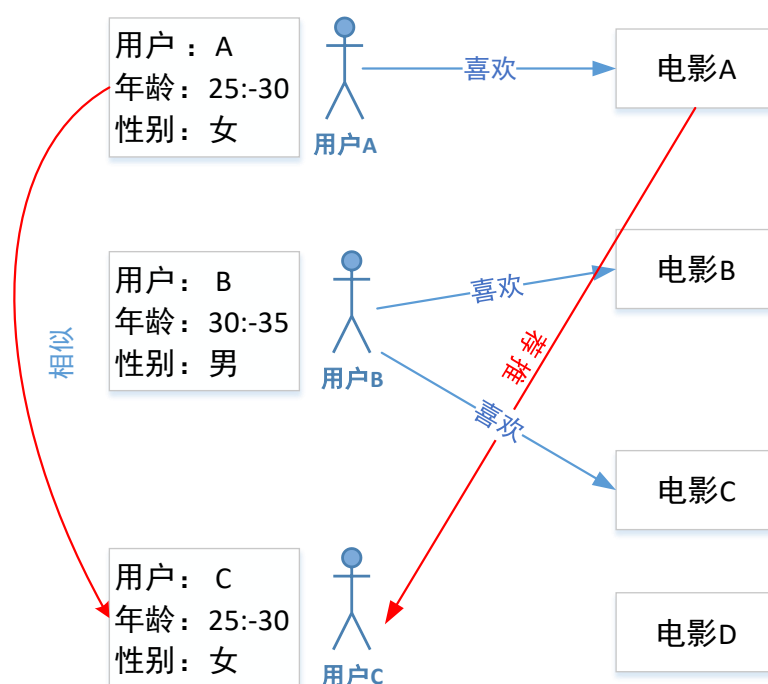


1、基于人口统计学的推荐算法



基于人口统计学的推荐机制（Demographic-based Recommendation）是一种最易于实现的推荐方法，它只是简单的根据系统用户的基本信息发现用户的相关程度，然后将相似用户喜爱的其他物品推荐给当前用户。系统根据用户的属性建模，比如用户的性别，年龄，兴趣等信息。根据这些特征计算用户间的相似度，比如上图系统发现用户A和C比较相似，就会把A喜欢的物品推荐给C。

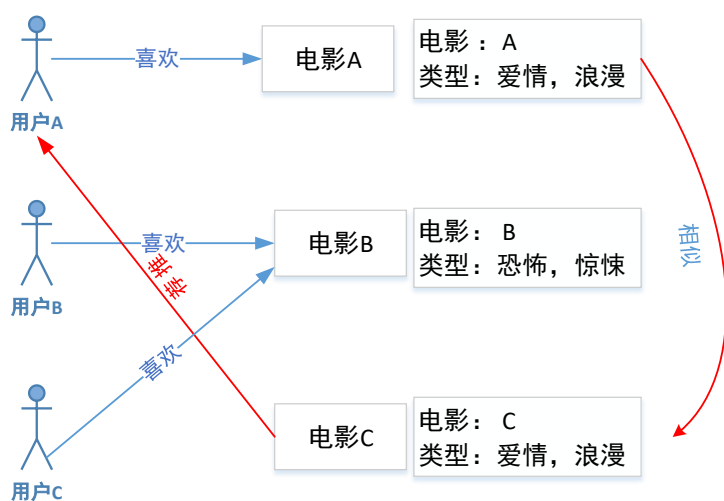
优势：

- a 不需要历史数据，没有冷启动问题
- b 不依赖于物品的属性，因此其他领域的问题都可无缝接入。

不足：

算法比较粗糙，效果很难令人满意，只适合简单的推荐

2、基于内容的推荐算法



基于内容的推荐是在推荐引擎出现之初应用最为广泛的推荐机制，它的核心思想是根据推荐物品或内容的元数据，发现物品或者内容的相关性，然后基于用户以往的喜好记录，

推荐给用户相似的物品。这种推荐系统多用于一些资讯类的应用上，针对文章本身抽取一些 tag 作为该文章的关键词，继而可以通过这些 tag 来评价两篇文章的相似度。系统根据电影的属性建模，比如电影的类型信息。根据这些特征计算电影间的相似度，比如上图系统发现电影 A 和 C 比较相似，系统还会发现用户 A 喜欢电影 A，由此得出结论，用户 A 很可能对电影 C 也感兴趣，于是将电影 C 推荐给 A。

优势：

对用户兴趣可以很好的建模，并通过对物品属性维度的增加，获得更好的推荐精度

不足：

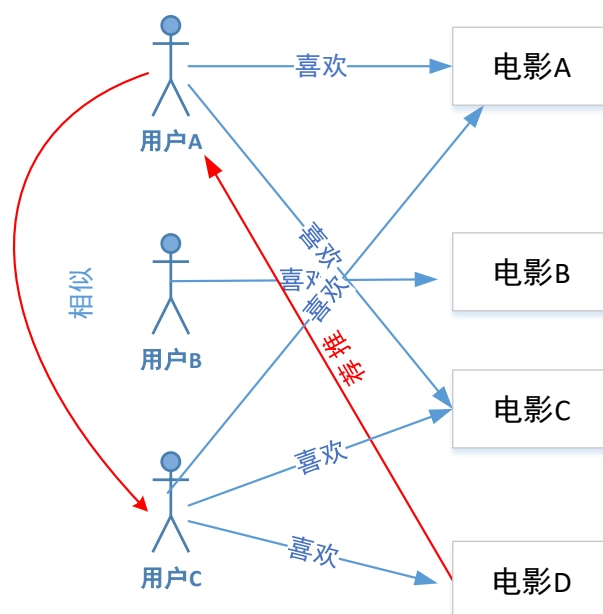
- a 物品的属性有限，很难有效的得到更多数据
- b 物品相似度的衡量标准只考虑到了物品本身，有一定的片面性
- c 需要用户的物品的历史数据，有冷启动的问题

3、协同过滤推荐算法

协同过滤（CF）推荐算法会寻找用户的行为模式，并据此创建用户专属的推荐内容。这种算法会根据系统中的用户使用数据——比如用户对读过书籍的评论来确定用户对其喜爱程度。关键概念在于：如果两名用户对于某件物品的评分方式类似，那么他们对于某个新物品的评分很可能也是相似的。值得注意的是：这种算法无需再额外依赖于物品信息（比如描述、元数据等）或者用户信息（比如感兴趣的物品、统计数据等）。协同过滤推荐算法可分为两类：基于邻域的与基于模型的。在前一种算法（也就是基于内存的协同过滤推荐算法）中，用户-物品评分可直接用以预测新物品的评分。而基于模型的算法则通过评分来研究预测性的模型，再根据模型对新物品作出预测。大致理念就是通过机器学习算法，在数据中找出模式，并将用户与物品间的互动方式模式化。

3.1 基于用户的协同过滤推荐算法

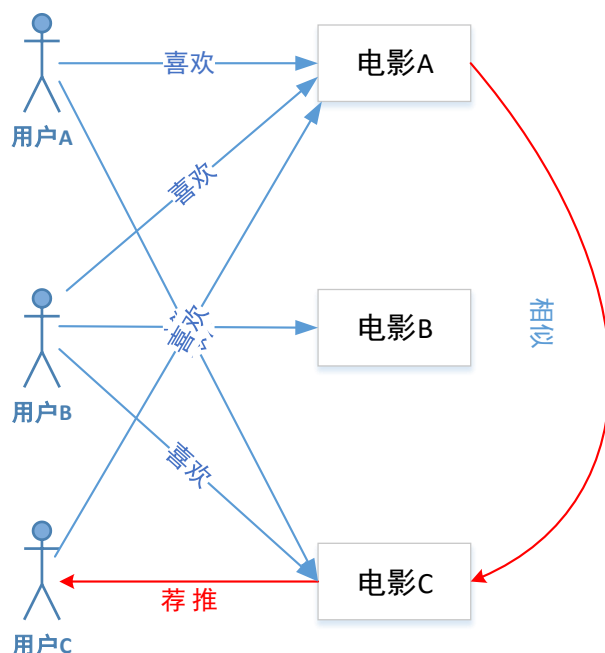
用户/物品	电影 A	电影 B	电影 C	电影 D
用户 A	√		√	推荐
用户 B		√		
用户 C	√		√	√



基于用户的 CF 的基本思想是基于用户对物品的偏好找到相邻邻居用户，然后将邻居用户喜欢的推荐给当前用户。计算上，将一个用户对所有物品的偏好作为一个向量来计算用户之间的相似度，找到 K 个邻居后，根据邻居的相似度权重以及他们对物品的偏好，预测当前用户没有偏好的未涉及物品，计算得到一个排序的物品列表作为推荐。下图给出了一个例子，对于用户 A，根据用户的历史偏好，这里只计算得到一个邻居 - 用户 C，然后将用户 C 喜欢的物品 D 推荐给用户 A。

3.2 基于物品的协同过滤推荐算法

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

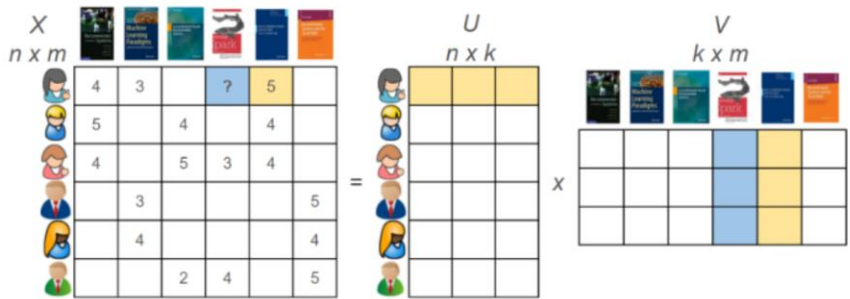


基于物品的 CF 的原理和基于用户的 CF 类似，只是在计算邻居时采用物品本身，而不是从用户的角度，即基于用户对物品的偏好找到相似的物品，然后根据用户的历史偏好，推荐相似的物品给他。从计算的角度看，就是将所有用户对某个物品的偏好作为一个向量来计算物品之间的相似度，得到物品的相似物品后，根据用户历史的偏好预测当前用户还没有表示偏好的物品，计算得到一个排序的物品列表作为推荐。上图例子表示对于物品 A，根据所有用户的历史偏好，喜欢物品 A 的用户都喜欢物品 C，得出物品 A 和物品 C 比较相似，而用户 C 喜欢物品 A，那么可以推断出用户 C 可能也喜欢物品 C。

3.3 基于模型的协同过滤推荐算法

基于模型的协同过滤方式可以克服基于邻域方法的限制。与使用用户-物品评分直接预测新物品评分的邻域方式不同，基于模型的方法则使用评分来研究预测性模型，并根据模型来预测新物品。大致理念就是通过机器学习算法，在数据中找出模式，并将用户与物品间的互动方式模式化。总体来讲，基于模型的协同过滤方式是构建协同过滤更高级的算法。很多不同的算法都能用来构建模型，以进行预测；例如贝叶斯网络、集群、分类、回归、矩阵因式分

解、受限波尔兹曼机等，这些技术其中有些在获得 Netflix Prize 奖项时起到了关键性作用。Netflix 在 2006 年到 2009 年间举办竞赛，当时还为能够生成准确度超过其系统 10%的推荐系统制作团队提供 100 万美元的大奖。胜出的解决方案是一套综合了逾 100 种不同算法模型，并在生产环境中采用了矩阵因式分解与受限玻尔兹曼机的方法。



矩阵因式分解（比如奇异值分解、SVD++）将物品与用户都转化为同一个隐空间，表现了用户与物品间的底层互动（上图）。矩阵因式分解背后的原理在于：其潜在特性代表了用户如何对物品进行评分。根据用户与物品的潜在表现，我们就可以预测用户对未评分的物品的喜爱程度。