**[推荐系统的常用算法概述](http://www.cnblogs.com/luchen927/archive/2012/02/04/2334316.html)**

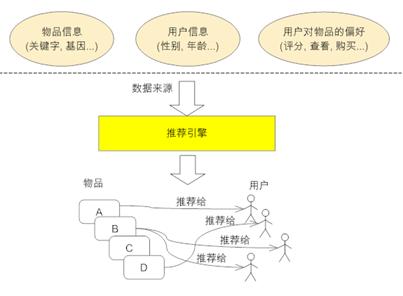
　　前一阵子准备毕业论文的开题，一直在看推荐系统相关的论文。对推荐系统有了一个更加清晰和理性的认识，也对推荐算法有了深入了解。借此机会总结分享一下，大家多多拍砖。

**推荐系统的出现**

　　随着互联网的发展，人们正处于一个信息爆炸的时代。相比于过去的信息匮乏，面对现阶段海量的信息数据，对信息的筛选和过滤成为了衡量一个系统好坏的重要指标。一个具有良好用户体验的系统，会将海量信息进行筛选、过滤，将用户最关注最感兴趣的信息展现在用户面前。这大大增加了系统工作的效率，也节省了用户筛选信息的时间。

　　搜索引擎的出现在一定程度上解决了信息筛选问题，但还远远不够。搜索引擎需要用户主动提供关键词来对海量信息进行筛选。当用户无法准确描述自己的需求时，搜索引擎的筛选效果将大打折扣，而用户将自己的需求和意图转化成关键词的过程本身就是一个并不轻松的过程。

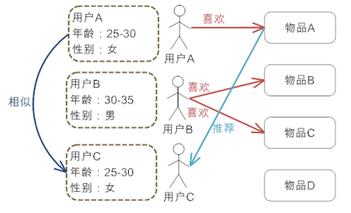
　　在此背景下，推荐系统出现了，推荐系统的任务就是解决上述的问题，联系用户和信息，一方面帮助用户发现对自己有价值的信息，另一方面让信息能够展现在对他感兴趣的人群中，从而实现信息提供商与用户的双赢。



**推荐算法介绍**

　　1 基于人口统计学的推荐

　　这是最为简单的一种推荐算法，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。



　　系统首先会根据用户的属性建模，比如用户的年龄，性别，兴趣等信息。根据这些特征计算用户间的相似度。比如系统通过计算发现用户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

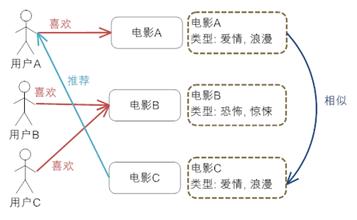
　　b 不依赖于物品的属性，因此其他领域的问题都可无缝接入。

　　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

　　与上面的方法相类似，只不过这次的中心转到了物品本身。使用物品本身的相似度而不是用户的相似度。



　　系统首先对物品（图中举电影的例子）的属性进行建模，图中用类型作为属性。在实际应用中，只根据类型显然过于粗糙，还需要考虑演员，导演等更多信息。通过相似度计算，发现电影A和C相似度较高，因为他们都属于爱情类。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

　　协同过滤是推荐算法中最经典最常用的，具体的思路我已经在之前的[一篇文章](http://www.cnblogs.com/luchen927/archive/2012/02/01/2325360.html)中提到过，大家可以去看一下。

　　这里只想介绍一下协同过滤中的方法和刚才提到方法的区别。

　　大家都知道协同过滤分为基于用户的协同过滤和基于物品的协同过滤。那么他们和基于人口学统计的推荐和基于内容的推荐有什么区别和联系呢？下面简单介绍一下。

　　基于用户的协同过滤——基于人口统计学的推荐

　　基于用户的协同过滤推荐机制和基于人口统计学的推荐机制都是计算用户的相似度，并基于“邻居”用户群计算推荐，但它们所不同的是如何计算用户的相似度，基 于人口统计学的机制只考虑用户本身的特征，而基于用户的协同过滤机制可是在用户的历史偏好的数据上计算用户的相似度，它的基本假设是，喜欢类似物品的用户 可能有相同或者相似的口味和偏好。

　　基于物品的协同过滤——基于内容的推荐

　　基于项目的协同过滤推荐和基于内容的推荐其实都是基于物品相似度预测推荐，只是相似度计算的方法不一样，前者是从用户历史的偏好推断，而后者是基于物品本身的属性特征信息。

　　协同过滤的优势：

　　a 它不需要对物品或者用户进行严格的建模，而且不要求物品的描述是机器可理解的，所以这种方法也是领域无关的。

　　b 这种方法计算出来的推荐是开放的，可以共用他人的经验，很好的支持用户发现潜在的兴趣偏好

　　不足：

　　a 方法的核心是基于历史数据，所以对新物品和新用户都有“冷启动”的问题。

　　b 推荐的效果依赖于用户历史偏好数据的多少和准确性。

　　c 在大部分的实现中，用户历史偏好是用稀疏矩阵进行存储的，而稀疏矩阵上的计算有些明显的问题，包括可能少部分人的错误偏好会对推荐的准确度有很大的影响等等。

　　d 对于一些特殊品味的用户不能给予很好的推荐。

　　e 由于以历史数据为基础，抓取和建模用户的偏好后，很难修改或者根据用户的使用演变，从而导致这个方法不够灵活。

　　4 混合推荐算法

　　以上介绍的方法是推荐领域最常见的几种方法。但是可以看出，每个方法都不是完美的。因此实际应用中，向Amazon这样的系统都是混合使用各种推荐算法，各取所长。因此我们在使用时，也可以多考虑一下什么情况下更适合使用哪种算法，来提高我们系统的效率。

**推荐系统的发展方向**

　　1 需要考虑在保证推荐准确性的同时增加推荐的多样性，使特殊喜好的用户得到推荐

　　2 需要考虑用户对系统的实时反馈，更好的利用反馈数据强化推荐算法

　　3 考虑时间，空间，任务等因素，来更好的完成推荐