I'm a DMer & MLer & NLPer & IRer
博文有原创，有转载。均为分享学习之用。P.S. 部分转载可能未注明出处。若有冒昧，请与我联系:(ILoveDataMining AT gmail DOT com)

非常好的协同过滤入门文章

“探索推荐引擎内部的秘密”系列将带领读者从浅入深的学习探索推荐引擎的机制，实现方法，其中还涉及一些基本的优化方法，例如聚类和分类的应用。同时在理论讲解的基础上，还会结合 Apache Mahout 介绍如何在大规模数据上实现各种推荐策略，进行策略优化，构建高效的推荐引擎的方法。本文作为这个系列的第一篇文章，将深入介绍推荐引擎的工作原理，和其中涉及的各种推荐机制，以及它们各自的优缺点和适用场景，帮助用户清楚的了解和快速构建适合自己的推荐引擎。

信息发现

如今已经进入了一个数据爆炸的时代，随着 Web 2.0 的发展， Web 已经变成数据分享的平台，那么，如何让人们在海量的数据中想要找到他们需要的信息将变得越来越难。

在这样的情形下，搜索引擎（Google，Bing，百度等等）成为大家快速找到目标信息的最好途径。在用户对自己需求相对明确的时候，用搜索引擎很方便的通过关键字搜索很快的找到自己需要的信息。但搜索引擎并不能完全满足用户对信息发现的需求，那是因为在很多情况下，用户其实并不明确自己的需要，或者他们的需求很难用简单的关键字来表述。又或者他们需要更加符合他们个人口味和喜好的结果，因此出现了推荐系统，与搜索引擎对应，大家也习惯称它为推荐引擎。

随着推荐引擎的出现，用户获取信息的方式从简单的目标明确的数据的搜索转换到更高级更符合人们使用习惯的信息发现。

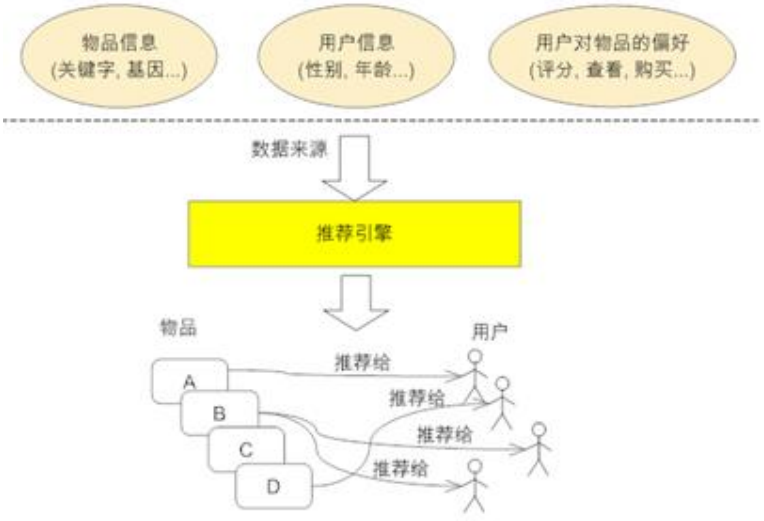
如今，随着推荐技术的不断发展，推荐引擎已经在电子商务 (E-commerce，例如 Amazon，当当网) 和一些基于 social 的社会化站点 (包括音乐，电影和图书分享，例如豆瓣，Mtime 等) 都取得很大的成功。这也进一步的说明了，Web2.0 环境下，在面对海量的数据，用户需要这种更加智能的，更加了解他们需求，口味和喜好的信息发现机制。

回页首

推荐引擎

前面介绍了推荐引擎对于现在的 Web2.0 站点的重要意义，这一章我们将讲讲推荐引擎到底是怎么工作的。推荐引擎利用特殊的信息过滤技术，将不同的物品或内容推荐给可能对它们感兴趣的用户。

图 **1.** 推荐引擎工作原理图



2014年5月 (1)
2014年3月 (2)
2014年1月 (1)
2013年7月 (1)
2013年6月 (1)
2013年5月 (1)
2013年2月 (1)
2013年1月 (5)
2012年12月 (3)
2012年11月 (2)
2012年9月 (2)
2012年7月 (1)
2012年6月 (11)
2012年5月 (21)
2012年4月 (19)
2012年3月 (33)
2012年2月 (7)
2012年1月 (5)
2011年12月 (61)
2011年10月 (1)
2011年9月 (2)

About Me
Homepage
Micro-Blog (Sina)

BLOG
A Link Collection

StuDy
Course
朝花夕拾

最新评论
1. Re:BP神经网络模型与学习算法
这个BP神经网络模型与学习算法PPT三层BP网络模型是不是错误的?
--ly382075924
2. Re:了解信息增益和决策树
例子太好了
--紫魂的海角
3. Re:了解信息增益和决策树
很赞 很赞
--残阳飞雪
4. Re:非常好的协同过滤入门文章
讲解的很有条理，赞~
--tinavalue
5. Re:大数据分析之——基于模型的复杂数据多维聚类分析
不错，值得一学
--飞哥007
6. Re:了解信息增益和决策树
因此特征T给系统带来的信息增益就可以写成系统原本的熵与固定特征T后的条件熵之差....好像没有什么因果关系
--noobpythoner
7. Re:转：关键词抽取（key words extraction）的相关研究
总结的不错，不知道博主试了这些方法哪个提取效果比较

图 1 给出了推荐引擎的工作原理图，这里先将推荐引擎看作黑盒，它接受的输入是推荐的数据源，一般情况下，推荐引擎所需要的数据源包括：

- 要推荐物品或内容的元数据，例如关键字，基因描述等；
- 系统用户的基本信息，例如性别，年龄等
- 用户对物品或者信息的偏好，根据应用本身的不同，可能包括用户对物品的评分，用户查看物品的记录，用户的购买记录等。其实这些用户的偏好信息可以分为两类：

- 显式的用户反馈：这类是用户在网站上自然浏览或者使用网站以外，显式的提供反馈信息，例如用户对物品的评分，或者对物品的评论。
- 隐式的用户反馈：这类是用户在使用网站是产生的数据，隐式的反应了用户对物品的喜好，例如用户购买了某物品，用户查看了某物品的信息等等。

显式的用户反馈能准确的反应用户对物品的真实喜好，但需要用户付出额外的代价，而隐式的用户行为，通过一些分析和处理，也能反映用户的喜好，只是数据不是很精确，有些行为的分析存在较大的噪音。但只要选择正确的行为特征，隐式的用户反馈也能得到很好的效果，只是行为特征的选择可能在不同的应用中有很大的不同，例如在电子商务的网站上，购买行为其实就是一个能很好表现用户喜好的隐式反馈。

推荐引擎根据不同的推荐机制可能用到数据源中的一部分，然后根据这些数据，分析出一定的规则或者直接对用户对其他物品的喜好进行预测计算。这样推荐引擎可以在用户进入的时候给他推荐他可能感兴趣的物品。

推荐引擎的分类

推荐引擎的分类可以根据很多指标，下面我们一一介绍一下：

1. 推荐引擎是不是为不同的用户推荐不同的数据

   根据这个指标，推荐引擎可以分为基于大众行为的推荐引擎和个性化推荐引擎

   - 根据大众行为的推荐引擎，对每个用户都给出同样的推荐，这些推荐可以是静态的由系统管理员人工设定的，或者基于系统所有用户的反馈统计计算出的当下比较流行的物品。
   - 个性化推荐引擎，对不同的用户，根据他们的口味和喜好给出更加精确的推荐，这时，系统需要了解需推荐内容和用户的特质，或者基于社会化网络，通过找到与当前用户相同喜好的用户，实现推荐。

   这是一个最基本的推荐引擎分类，其实大部分人们讨论的推荐引擎都是将个性化的推荐引擎，因为从根本上说，只有个性化的推荐引擎才是更加智能的信息发现过程。

2. 根据推荐引擎的数据源

   其实这里讲的是如何发现数据的相关性，因为大部分推荐引擎的工作原理还是基于物品或者用户的相似集进行推荐。那么参考图 1 给出的推荐系统原理图，根据不同的数据源发现数据相关性的方法可以分为以下几种：

   - 根据系统用户的基本信息发现用户的相关程度，这种被称为基于人口统计学的推荐（Demographic-based Recommendation）
   - 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，这种被称为基于内容的推荐（Content-based Recommendation）
   - 根据用户对物品或者信息的偏好，发现物品或者内容本身的相关性，或者是发现用户的相关性，这种被称为基于协同过滤的推荐（Collaborative Filtering-based Recommendation）。

3. 根据推荐模型的建立方式

   可以想象在海量物品和用户的系统中，推荐引擎的计算量是相当大的，要实现实时的推荐务必需要建立一个推荐模型，关于推荐模型的建立方式可以分为以下几种：

   - 基于物品和用户本身的，这种推荐引擎将每个用户和每个物品都当作独立的实体，预测每个用户对于每个物品的喜好程度，这些信息往往是用一个二维矩阵描述的。由于用户感兴趣的物品远远小于总物品的数目，这样的模型导致大量的数据空置，即我们得到的二维矩阵往往是一个很大的稀疏矩阵。同时为了减小计算量，我们可以对物品和用户进行聚类， 然后记录和计算一类

用户对一类物品的喜好程度，但这样的模型又会在推荐的准确性上有损失。

- 基于关联规则的推荐（Rule-based Recommendation）：关联规则的挖掘已经是数据挖掘中的一个经典的问题，主要是挖掘一些数据的依赖关系，典型的场景就是“购物篮问题”，通过关联规则的挖掘，我们可以找到哪些物品经常被同时购买，或者用户购买了一些物品后通常会购买哪些其他的物品，当我们挖掘出这些关联规则之后，我们可以基于这些规则给用户进行推荐。
- 基于模型的推荐（Model-based Recommendation）：这是一个典型的机器学习的问题，可以将已有的用户喜好信息作为训练样本，训练出一个预测用户喜好的模型，这样以后用户在进入系统，可以基于此模型计算推荐。这种方法的问题在于如何将用户实时或者近期的喜好信息反馈给训练好的模型，从而提高推荐的准确度。

其实在现在的推荐系统中，很少有只使用了一个推荐策略的推荐引擎，一般都是在不同的场景下使用不同的推荐策略从而达到最好的推荐效果，例如 Amazon 的推荐，它将基于用户本身历史购买数据的推荐，和基于用户当前浏览的物品的推荐，以及基于大众喜好的当下比较流行的物品都在不同的区域推荐给用户，让用户可以从全方位的推荐中找到自己真正感兴趣的物品。

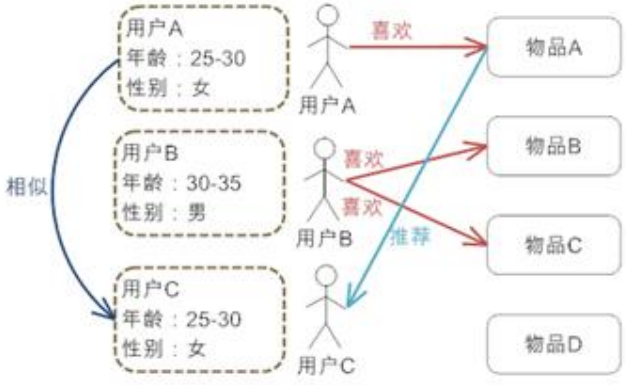
深入推荐机制

这一章的篇幅，将详细介绍各个推荐机制的工作原理，它们的优缺点以及应用场景。

基于人口统计学的推荐

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，图 2 给出了这种推荐的工作原理。

图 **2.** 基于人口统计学的推荐机制的工作原理



从图中可以很清楚的看到，首先，系统对每个用户都有一个用户 Profile 的建模，其中包括用户的基本信息，例如用户的年龄，性别等等；然后，系统会根据用户的 Profile 计算用户的相似度，可以看到用户 A 的 Profile 和用户 C 一样，那么系统会认为用户 A 和 C 是相似用户，在推荐引擎中，可以称他们是“邻居”；最后，基于“邻居”用户群的喜好推荐给当前用户一些物品，图中将用户 A 喜欢的物品 A 推荐给用户 C。

这种基于人口统计学的推荐机制的好处在于：

1. 因为不使用当前用户对物品的喜好历史数据，所以对于新用户来讲没有“冷启动（Cold Start）”的问题。
2. 这个方法不依赖于物品本身的数据，所以这个方法在不同物品的领域都可以使用，它是领域独立的（domain-independent）。

那么这个方法的缺点和问题是什么呢？这种基于用户的基本信息对用户进行分类的方法过于粗糙，尤其是对品味要求较高的领域，比如图书，电影和音乐等领域，无法得到很好的推荐效果。可能在一些电子商务的网站中，这个方法可以给出一些简单的推荐。另外一个局限是，这个方法可能涉及到一些与信息发现问题本身无关却比较敏感的信息，比如用户的年龄等，这些用户信息不是很好获取。

基于内容的推荐

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。图 3 给出了基于内容推荐的基本原理。

图 **3.** 基于内容推荐机制的基本原理

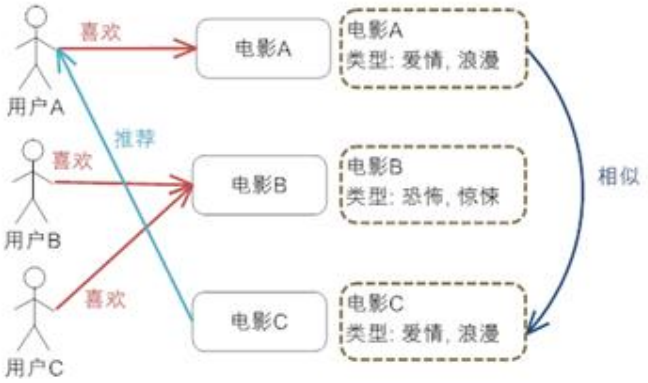


图 3 中给出了基于内容推荐的一个典型的例子，电影推荐系统，首先我们需要对电影的元数据有一个建模，这里只简单的描述了一下电影的类型；然后通过电影的元数据发现电影间的相似度，因为类型都是“爱情，浪漫”电影 A 和 C 被认为是相似的电影（当然，只根据类型是不够的，要得到更好的推荐，我们还可以考虑电影的导演，演员等等）；最后实现推荐，对于用户 A，他喜欢看电影 A，那么系统就可以给他推荐类似的电影 C。

这种基于内容的推荐机制的好处在于它能很好的建模用户的口味，能提供更加精确的推荐。但它也存在以下几个问题：

1. 需要对物品进行分析和建模，推荐的质量依赖于对物品模型的完整和全面程度。在现在的应用中我们可以观察到关键词和标签（Tag）被认为是描述物品元数据的一种简单有效的方法。
2. 物品相似度的分析仅仅依赖于物品本身的特征，这里没有考虑人对物品的态度。
3. 因为需要基于用户以往的喜好历史做出推荐，所以对于新用户有“冷启动”的问题。

虽然这个方法有很多不足和问题，但他还是成功的应用在一些电影，音乐，图书的社交站点，有些站点还请专业的人员对物品进行基因编码，比如潘多拉，在一份报告中说道，在潘多拉的推荐引擎中，每首歌有超过 100 个元数据特征，包括歌曲的风格，年份，演唱者等等。

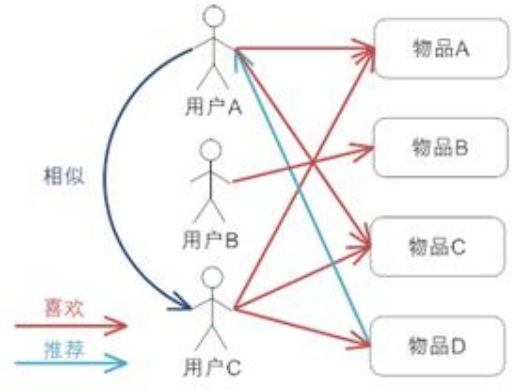
基于协同过滤的推荐

随着 Web2.0 的发展，Web 站点更加提倡用户参与和用户贡献，因此基于协同过滤的推荐机制因运而生。它的原理很简单，就是根据用户对物品或者信息的偏好，发现物品或者内容本身的相关性，或者是发现用户的相关性，然后再基于这些关联性进行推荐。基于协同过滤的推荐可以分为三个子类：基于用户的推荐（User-based Recommendation），基于项目的推荐（Item-based Recommendation）和基于模型的推荐（Model-based Recommendation）。下面我们一个一个详细的介绍着三种协同过滤的推荐机制。

基于用户的协同过滤推荐

基于用户的协同过滤推荐的基本原理是，根据所有用户对物品或者信息的偏好，发现与当前用户口味和偏好相似的“邻居”用户群，在一般的应用中是采用计算“K- 邻居”的算法；然后，基于这 K 个邻居的历史偏好信息，为当前用户进行推荐。下图 4 给出了原理图。

图 **4.** 基于用户的协同过滤推荐机制的基本原理



上图示意出基于用户的协同过滤推荐机制的基本原理，假设用户 A 喜欢物品 A，物品 C，用户 B 喜欢物品 B，用户 C 喜欢物品 A，物品 C 和物品 D；从这些用户的历史喜好信息中，我们可以发现用户 A 和用户 C 的口味和偏好是比较类似的，同时用户 C 还喜欢物品 D，那么我们可以推断用户 A 可能也喜欢物品 D，因此可以将物品 D 推荐给用户 A。

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。

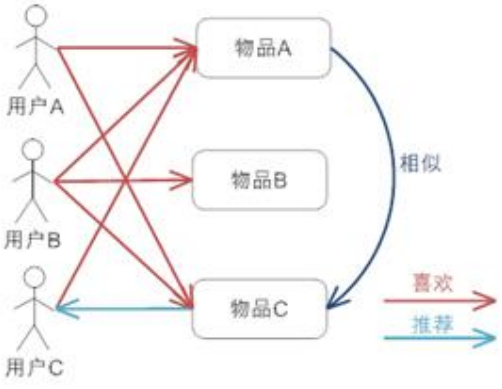
基于项目的协同过滤推荐

基于项目的协同过滤推荐的基本原理也是类似的，只是说它使用所有用户对物品或者信息的偏好，发现物品和物品之间的相似度，然后根据用户的历史偏好信息，将类似的物品推荐给用户，图 5 很好的诠释了它的基本原理。

假设用户 A 喜欢物品 A 和物品 C，用户 B 喜欢物品 A，物品 B 和物品 C，用户 C 喜欢物品 A，从这些用户的历史喜好可以分析出物品 A 和物品 C 时比较类似的，喜欢物品 A 的人都喜欢物品 C，基于这个数据可以推断用户 C 很有可能也喜欢物品 C，所以系统会将物品 C 推荐给用户 C。

与上面讲的类似，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。

图 **5.** 基于项目的协同过滤推荐机制的基本原理



同时协同过滤，在基于用户和基于项目两个策略中应该如何选择呢？其实基于项目的协同过滤推荐机制是 Amazon 在基于用户的机制上改良的一种策略，因为在大部分的 Web 站点中，物品的个数是远远小于用户的数量的，而且物品的个数和相似度相对比较稳定，同时基于项目的机制比基于用户的实时性更好一些。但也不是所有的场景都是这样的情况，可以设想一下在一些新闻推荐系统中，也许物品，也就是新闻的个数可能大于用户的个数，而且新闻的更新程度也有很快，所以它的形似度依然不稳定。所以，其实可以看出，推荐策略的选择其实和具体的应用场景有很大的关系。

基于模型的协同过滤推荐

基于模型的协同过滤推荐就是基于样本的用户喜好信息，训练一个推荐模型，然后根据实时的用户喜好的信息进行预测，计算推荐。

基于协同过滤的推荐机制是现今应用最为广泛的推荐机制，它有以下几个显著的优点：

1. 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。
2. 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

而它也存在以下几个问题：

1. 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。
2. 推荐的效果依赖于用户历史偏好数据的多少和准确性。
3. 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。
4. 对于一些特殊品味的用户不能给予很好的推荐。
5. 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。

混合的推荐机制

在现行的 Web 站点上的推荐往往都不是单纯只采用了某一种推荐的机制和策略，他们往往是将多个方法混合在一起，从而达到更好的推荐效果。关于如何组合各个推荐机制，这里讲几种比较流行的组合方法。

1. 加权的混合（Weighted Hybridization）: 用线性公式（linear formula）将几种不同的推荐按照一定权重组合起来，具体权重的值需要在测试数据集上反复实验，从而达到最好的推荐效果。
2. 切换的混合（Switching Hybridization）：前面也讲到，其实对于不同的情况（数据量，系统运行状况，用户和物品的数目等），推荐策略可能有很大的不同，那么切换的混合方式，就是允许在不同的情况下，选择最为合适的推荐机制计算推荐。
3. 分区的混合（Mixed Hybridization）：采用多种推荐机制，并将不同的推荐结果分不同的区显示给用户。其实，Amazon，当当网等很多电子商务网站都是采用这样的方式，用户可以得到很全面的推荐，也更容易找到他们想要的东西。
4. 分层的混合（Meta-Level Hybridization）: 采用多种推荐机制，并将一个推荐机制的结果作为另一个的输入，从而综合各个推荐机制的优缺点，得到更加准确的推荐。

推荐引擎的应用

介绍完推荐引擎的基本原理，基本推荐机制，下面简要分析几个有代表性的推荐引擎的应用，这里选择两个领域：Amazon 作为电子商务的代表，豆瓣作为社交网络的代表。

推荐在电子商务中的应用 **– Amazon**

Amazon 作为推荐引擎的鼻祖，它已经将推荐的思想渗透在应用的各个角落。Amazon 推荐的核心是通过数据挖掘算法和比较用户的消费偏好于其他用户进行对比，借以预测用户可能感兴趣的商品。对应于上面介绍的各种推荐机制，Amazon 采用的是分区的混合的机制，并将不同的推荐结果分不同的区显示给用户，图 6 和图 7 展示了用户在 Amazon 上能得到的推荐。

图 **6. Amazon** 的推荐机制 **-** 首页

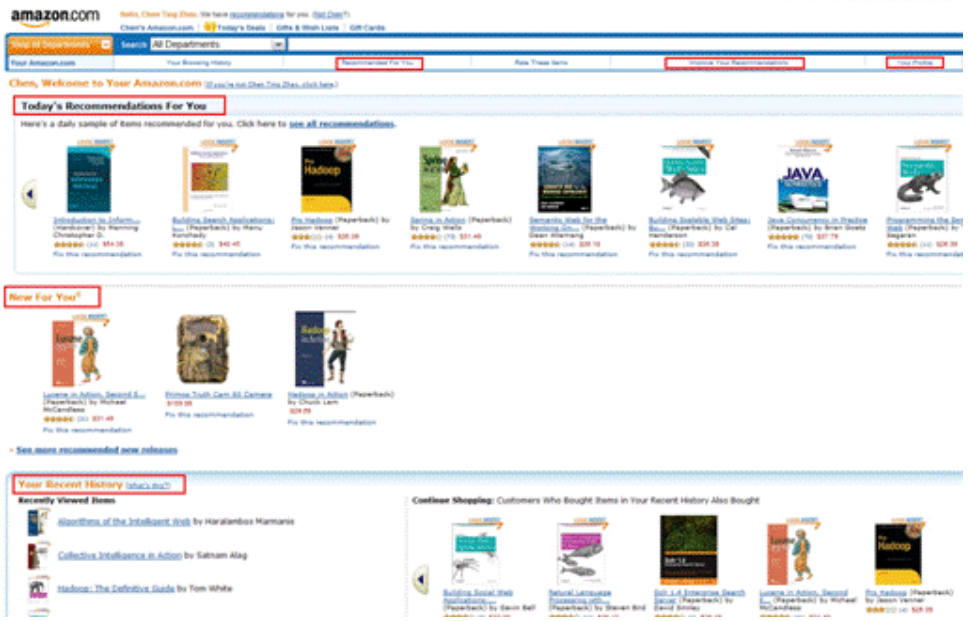
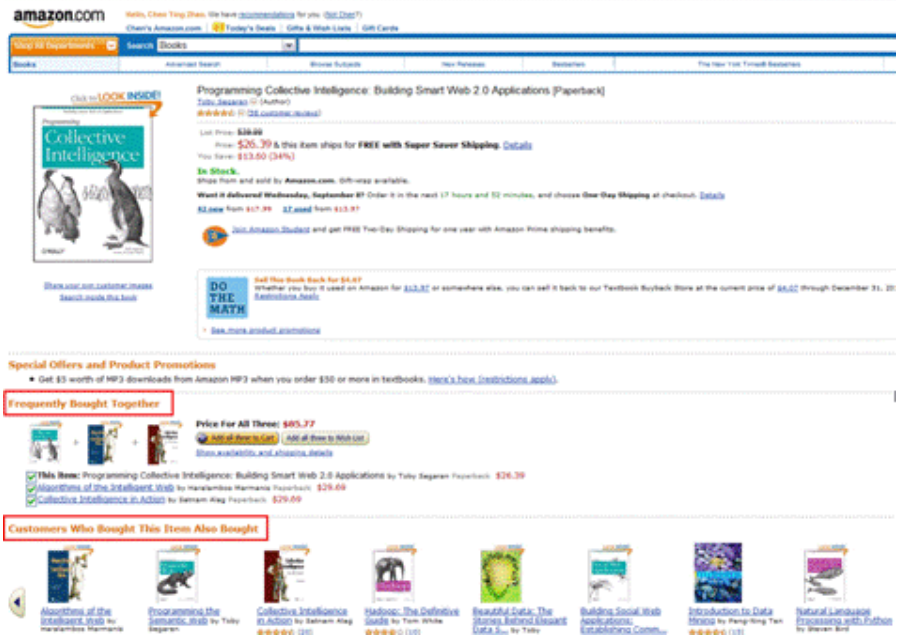


图 **7. Amazon** 的推荐机制 **-** 浏览物品



Amazon 利用可以记录的所有用户在站点上的行为，根据不同数据的特点对它们进行处理，并分成不同区为用户推送推荐：

- 今日推荐 (Today's Recommendation For You): 通常是根据用户的近期的历史购买或者查看记录，并结合时下流行的物品给出一个折中的推荐。
- 新产品的推荐 (New For You): 采用了基于内容的推荐机制 (Content-based Recommendation)，将一些新到物品推荐给用户。在方法选择上由于新物品没有大量的用户喜好信息，所以基于内容的推荐能很好的解决这个“冷启动”的问题。
- 捆绑销售 (Frequently Bought Together): 采用数据挖掘技术对用户的购买行为进行分析，找到经常被一起或同一个人购买的物品集，进行捆绑销售，这是一种典型的基于项目的协同过滤推荐机制。
- 别人购买 / 浏览的商品 (Customers Who Bought/See This Item Also Bought/See): 这也是一个典型的基于项目的协同过滤推荐的应用，通过社会化机制用户能更快更方便的找到自己感兴趣的物品。

值得一提的是，Amazon 在做推荐时，设计和用户体验也做得特别独到：

Amazon 利用有它大量历史数据的优势，量化推荐原因。

- 基于社会化的推荐，Amazon 会给你事实的数据，让用户信服，例如：购买此物品的

用户百分之多少也购买了那个物品；

- 基于物品本身的推荐，Amazon 也会列出推荐的理由，例如：因为你的购物框中有 ***，或者因为你购买过 ***，所以给你推荐类似的 ***。

另外，Amazon 很多推荐是基于用户的 profile 计算出来的，用户的 profile 中记录了用户在 Amazon 上的行为，包括看了那些物品，买了那些物品，收藏夹和 wish list 里的物品等等，当然 Amazon 里还集成了评分等其他的用户反馈的方式，它们都是 profile 的一部分，同时，Amazon 提供了让用户自主管理自己 profile 的功能，通过这种方式用户可以更明确的告诉推荐引擎他的品味和意图是什么。

推荐在社交网站中的应用 – 豆瓣

豆瓣是国内做的比较成功的社交网站，它以图书，电影，音乐和同城活动为中心，形成一个多元化的社交网络平台，自然推荐的功能是必不可少的，下面我们看看豆瓣是如何推荐的。

图 **8** . 豆瓣的推荐机制 **-** 豆瓣电影

当你在豆瓣电影中将一些你看过的或是感兴趣的电影加入你看过和想看的列表里，并为它们做相应的评分，这时豆瓣的推荐引擎已经拿到你的一些偏好信息，那么它将给你展示如图 8 的电影推荐。

图 **9** . 豆瓣的推荐机制 **-** 基于用户品味的推荐

豆瓣的推荐是通过“豆瓣猜”，为了让用户清楚这些推荐是如何来的，豆瓣还给出了“豆瓣猜”的一个简要的介绍。

“你的个人推荐是根据你的收藏和评价自动得出的，每个人的推荐清单都不同。你的收藏和

评价越多，豆瓣给你的推荐会越准确和丰富。
每天推荐的内容可能会有变化。随着豆瓣的长大，给你推荐的内容也会越来越准。”

这一点让我们可以清晰明了的知道，豆瓣必然是基于社会化的协同过滤的推荐，这样用户越多，用户的反馈越多，那么推荐的效果会越来越准确。

相对于 Amazon 的用户行为模型，豆瓣电影的模型更加简单，就是“看过”和“想看”，这也让他们的推荐更加专注于用户的品味，毕竟买东西和看电影的动机还是有很大不同的。

另外，豆瓣也有基于物品本身的推荐，当你查看一些电影的详细信息的时候，他会给你推荐出“喜欢这个电影的人也喜欢的电影”， 如图 10，这是一个基于协同过滤的应用。

图 **10** . 豆瓣的推荐机制 **-** 基于电影本身的推荐

总结

在网络数据爆炸的年代，如何让用户更快的找到想要的数据，如何让用户发现自己潜在的兴趣和需求，无论是对于电子商务还是社会网络的应用都是至关重要的。推荐引擎的出现，使得这个问题越来越被大家关注。但对大多数人来讲，也许还在惊叹它为什么总是能猜到你到底想要些什么。推荐引擎的魔力在于你不清楚在这个推荐背后，引擎到底记录和推理了些什么。

通过这篇综述性的文章，你可以了解，其实推荐引擎只是默默的记录和观察你的一举一动，然后再借由所有用户产生的海量数据分析和发现其中的规律，进而慢慢的了解你，你的需求，你的习惯，并默默的无声息的帮助你快速的解决你的问题，找到你想要的东西。

其实，回头想想，很多时候，推荐引擎比你更了解你自己。

通过第一篇文章，相信大家对推荐引擎有一个清晰的第一印象，本系列的下一篇文章将深入介绍基于协同过滤的推荐策略。在现今的推荐技术和算法中，最被大家广泛认可和采用的就是基于协同过滤的推荐方法。它以其方法模型简单，数据依赖性低，数据方便采集，推荐效果较优等多个优点成为大众眼里的推荐算法“No.1”。本文将带你深入了解协同过滤的秘密，并给出基于 Apache Mahout 的协同过滤算法的高效实现。Apache Mahout 是 ASF 的一个较新的开源项目，它源于 Lucene，构建在 Hadoop 之上，关注海量数据上的机器学习经典算法的高效实现。

感谢大家对本系列的关注和支持。

分类: Data Mining,Recommendation System

« 上一篇：位置+推荐

» 下一篇：麦包包也看到了个性化推荐：数据驱动销售——个性化推荐引擎

posted on 2011-12-16 11:21 wentingtu 阅读(25090) 评论(5) 编辑 收藏

Feedback

#1楼 2012-04-07 14:10 陈艺勇

毕业设计是基于SNS网站的个性化推荐，正在学些数据挖掘的东西，博主这篇文章立马让我对其有了个初步的了解，很有用，很给力，顶一下！

支持(0) 反对(0)

#2楼 2012-11-19 20:10 davidrui

阅读后帮助很大，谢谢！

支持(0) 反对(0)

#3楼 2014-11-03 16:12 猫二爷

写的很好！很有忙著 谢谢

支持(0) 反对(0)

#4楼 2016-01-27 15:43 xx ee

好文章

支持(0) 反对(0)

#5楼 2017-07-25 14:39 tinavalue

讲解的很有条理，赞~

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新**IT**新闻：

· 当心！穿"外卖服"的不一定是外卖小哥

· AI校招程序员最高薪酬曝光！腾讯80万年薪领跑，还送北京户口

· 王健林讲话完整版曝光：万达苏宁明年将在资本方面有动作

· 比特币网创始人卖掉所有比特币：投资风险太高

· 支付宝福利：免费扫码领红包 赏金翻倍
» 更多新闻...

最新知识库文章：
· 以操作系统的角度述说线程与进程
· 软件测试转型之路
· 门内门外看招聘
· 大道至简，职场上做人做事做管理
· 关于编程，你的练习是不是有效的？
» 更多知识库文章...