# 推荐系统算法综述

从上个世纪90年代中期协同过滤的文章出现开始，推荐系统成为一个重要的研究领域。至今在工业界和学术界已有大量成果。在信息爆炸的时代，推荐系统能够有效地帮助user选取有效和有趣的内容，因此至今相关研究仍然火热。本篇文章对已有的推荐系统算法做了调研，并针对不同的方法判断其局限性，讨论扩展方法。

## 问题定义

推荐系统领域和认知科学，近似理论、信息检索等都有关系，但在1990s就成为了独立的学科。早期的推荐系统主要是评分，被形式化为根据user对其他item的打分来估算对其从来没有见过的item的打分，然后向user推荐所估计打分比较高的item。

给定一个集合。。。

通过已知的分数推算出未知的分数通常有两种思路，1.启发式算法，定义效用函数，根据经验验证其性能；2. 定义一种性能标准，例如平均平方误差，然后进行优化。

一旦能够预测到未知评分，我们可以选择评分最高的item推荐给user，也可以选择Top-N的item推荐给user或者Top-N的user给一件商品。

推荐系统的可以分成以下三类：

1. 基于内容的推荐（content-based recommendations）
2. 协同过滤
3. 混合策略（Hybrid approaches）:结合协同过滤和基于内容的推荐。

## Content-based Methods

Content-based Methods 是向user推荐与该user喜欢的item相似的item。例如像user推荐电影，Content-based methods 是试图找出user喜欢电影的共同点，例如导演、类型、演员等特征，然后向该user推荐拥有这些特征的电影。Content-based Methods只能向user推荐与user喜欢的item有高度相似性的item。

很多Content-based Methods 是基于文本信息推荐。根据user的个人资料(profile)可以容易地获得user的喜好和需求，user的个人资料可以通过user填写问卷表单等方式直接获取，也可以通过分析user的行为间接获取。对于item的资料(item profile)，一般是从item资料的内容中提取特征，item资料一般通过文本描述，内容特征可以通过关键字来描述。然后通过TF-IDF对关键字构成向量，再通过cosine similarity计算相似度，可以得到item之间的相似度，选取与user喜好相似度高的item进行推荐。

###什么是推荐系统

* Item
* User
* Transaction

### 推荐系统的历史

推荐系统和信息检索、预测理论等研究有非常密切的相关性，前者可以看成是后者的延伸研究。推荐系统成为一个独立的学科一般认为从1994年明尼苏达大学GroupLens研究组的 GroupLens 研究系统。该系统有两大重要贡献：一是首次ᨀ出了基于协同过滤 (Collaborative Filtering)来完成推荐任务的思想，二是为推荐问题建立了一个形式化的模型(见 1.4)。基于该模型的协同过滤推荐引领了之后推荐系统在今后十几年的发展方向。

### 推荐系统评测指标

推荐系统可以用多个指标进行描述，有些可以定量计算，有些只能定性描述。

1. 用户满意度。主要通过用户调查或在线实验获得。如GroupLens曾设计调查问卷调查用户对推荐论文结果的感受。电子商务网站可以通过推荐物品点击率、是否购买等信息获得用户满意度。视频网站可以通过点击率、用户停留时间等信息评价用户满意度。
2. 预测准确度。最重要的推荐系统离线评测指标。将数据集分成训练集和测试集，计算推荐算法在测试集上的预测准确度。

* 评分预测 评分预测通过 user 以往对 item 的打分情况，预测对新的 item 的打分情况。评分预测的预测准确度一般通过均方差误差(RMSE)和平均局对误差(MAE)计算，对于测试集中的一个一个 user 对 item 打分，令是用户对物品的世纪评分，而是推荐算法给出的预测评分，则RMSE的定义为:

用绝对值计算预测误差，它的定义为：

* TopN 预测

很多网站的推荐服务是提供给用户一个个性化 item 列表，如电影列表，选取 user 最有可能感兴趣的 N 个 item 进行推荐，这种推荐叫做 TopN 推荐。TopN推荐一般通过 准确率(precision)/召回率(recall)度量。

令是根据用户在训练集上的行为给用户作出的推荐列表，而是用户在测试集上的行为列表，召回率为：

准确率定义为：

为了全面评测 TopN 推荐的准确率和召回率，会选取不同的推荐列表长度计算出多组准确率/召回率，然后画出 准确率/召回率 曲线(P-R曲线)。

1. 覆盖率 覆盖率描述推荐系统对物品长尾的挖掘能力，如淘宝的“去爆款”化，覆盖率有不同的定义，最简单的是推荐物品的数量占总数量的比例。设系统的用户集合为,为每个用户推荐一个长度为的物品列表，推荐系统覆盖率为：

这样计算覆盖率最简单，但很粗略，覆盖率相同但流行度分布有无数的可能性。另外两个指标可以克服这个问题，分别是信息熵和基尼系数。

信息熵计算方法如下：

$$
H=-\Sum{i=1}^{n}p(i)\log{p(i)}
$$

其中是物品的流行度除以所有物品流行度之和。

基尼系数计算方法如下：

其中,是按照物品流行度 p()从小到大排序的物品列表中第个物品。

1. 多样性

用户的兴趣是广泛的，我们希望推荐系统推荐的物品是多样的，增加用户找到感兴趣物品的概率。多样性描述了推荐列表中物品两两之间的不相似性，因此多样性与相似性是对应的。假设定义了物品和之间的相似度，那么用户的推荐列表的多样性定义如下：

系统整体的多样性可以定义为所有用户推荐列表多样性的平均值

1. 新颖性 新颖的推荐是指给用户推荐之前没有听说过的物品，网站实现新颖性可以将用户有过行为的物品过滤掉。推荐越不热门的物品，用户越可能觉得新颖。
2. 惊喜度(serendipity)

推荐系统中的惊喜度和新颖度有区别，举例说明，如果一个用户喜欢周星驰的电影，系统给他推荐一部叫做《临歧》的电影（该电影是1983年由刘德华、周星驰、梁朝伟合作演出的，很少有人知道这部有周星驰出演的电影），而该用户不知道这部电影，那么这个推荐具有新颖性，但是没有惊喜度，因为该用户一旦了解了这个电影的演员，就不会觉得特别奇怪。但如果我们给用户推荐张艺谋导演的《红高粱》，假设这名用户没有看过这部电影，那么他看完这部电影后可能会觉得很奇怪，因为这部电影和他的兴趣一点关系也没有，但如果用户看完电影后觉得这部电影很不错，那么就可以说这个推荐是让用户惊喜的。

惊喜度没有固定的衡量标准，但首先需要衡量推荐 item 和 user 的历史兴趣的相似度，然后衡量用户对推荐系统的满意度。

1. 信任度

在推荐系统中，认为同样的推荐结果，如果是用户信任的方式推荐则用户会更加有购买欲。增加信任的方式一是可以增加推荐系统的透明性，提供推荐介事，二是可以考虑用户的社交网络，通过用户的好友做推荐。

此外，衡量推荐系统还有实时性、健壮性、考虑商业目标等方面。

### 推荐系统的关键技术

* content-based，基于内容的推荐。主要思想是认为 user 会喜欢和自己喜欢的 item 类似的 item. 两个 item 是否相似通过比较 item 的内容特征。如一个用户在某电影网站上给喜剧1评价很高，则推荐系统会倾向给用户推荐喜剧2，因为它们都是喜剧。content-based 关于item 特征的提取很多时候基于自然语言处理的相关技术，例如，分词等。
* Collaborative filtering, 协同过滤，是最简单也是最原始的推荐方法。基本思想是向 user A 推荐与其喜好相似的 user B 所喜欢的 item。 主要通过两个 user 的历史打分记录来判断两个用户的喜好是否相同，称为 user-user approach。 同理，通过 item 的历史被打分记录可以判断两个 item 之间的相似性，从而向 user 推荐和其评分比较高的 item 相似的 item, 称为 item-item approach. 这两种方法都是基于近邻思想寻找相似的 item 和 user，同时由于计算简单，可以直接在内存中完成，被归为基于内存(memory-based)的协同过滤方法一类中。随着机器学习方法的发展，越来越多的机器学习模型也被用于推荐系统，与基于内存的方法不同，基于模型(model-based)的方法一般需要实现训练好保存下来，然后进行推荐。常用的model-based方法有矩阵分解等。
* 用户特征(Demographic): 此类方法主要运用 user 的信息进行个性化推荐，例如根据 user 的语言或者年龄向用户推荐不同的信息。
* Knowledge-based，基于专家知识，协同过滤和基于内容的推荐方法可以以相对较小的代价获取推荐内容，但是在一些场景下，比如房屋、汽车、计算机等商品，需要一些专业知识，并且可能会由于评分数据少而效果不好，这些场景下则需要一些专家知识辅助推荐。基于知识的推荐需要用户指定需求，共分为基于约束推荐和基于实例推荐。两者的共同点事都需要用户指定需求，然后系统给出解决方案，不同点是基于约束推荐将用户给出的需求条件作为约束直接过滤，基于实例推荐则通过不同的相似度衡量方法检索出相似的物品。
* Community-based, 基于社团的推荐。根据 user 的社交好友关系进行推荐，本方法的基本思想是比起喜好相同的陌生人，user 更容易和自己的好友形成相似的喜好。腾讯社交广告第二届题目即是通过好友关系推荐广告。
* Hybrid recommender system, 混合推荐系统。多种推荐系统组合。

其中 content-based 和 collaborative filtering 两种是最基础和最常用的方法，本调研主要围绕这两种方法展开。

## Content-based

Content-based 推荐主要是通过分析 user 之前打过分的 item, 并基于这些 item 的特征挖掘 user 的兴趣，基于挖掘的兴趣对 user 进行其他 item 的推荐，总体来说，Content-based 推荐系统一般分为三步：content analyzer, profile learner, filter component。

* Content analyzer, 主要是对无结构的 item 的内容信息进行预处理，例如对描述 item 的文本，网页等进行处理和表示，以便能够输入到下一个阶段。
* profile learner, 收集到 user 打分过的 item 的内容表示之后，对其进行特征分析和提取，以便挖掘 user 的兴趣点。
* filter component, 通过获得的 user 的兴趣点，对其他 item 过滤并推荐。

### Content-based 方法的优缺点

与协同过滤相比， content-based方法有以下优点：

* User independence，content-based 方法为每一个用户建立一个独立的 profile， 而协同过滤则需要找与其相似的用户，只有相似的用户打分高的 item 才会被推荐。
* transparency，推荐内容可解释，由于推荐的item都是根据某些特征而来，与协同过滤黑盒不同，content-based拥有更好的可解释性。
* New item, content-based 对新的 item 是非常友好的，即使一个 item 没有被其他用户打过分，但content-based 算法仍然可以将其推荐给其他用户，协同过滤做不到。

content-based 也有缺点：

* Limited content analysis，无论是自动构造特征还是人工构造特征，特征的类型和数量都是有限的。一旦没有从内容中分析出有效的特征信息，则无法做出有效地推荐。
* Over-spacialization, content-based 推荐会非常集中于某一类别，无法给用户推荐惊喜的item。越完美的content-based推荐系统越无法向用户推荐惊喜的 item 。
* New user，新用户由于没有打分的 item, 因此也就没有挖掘兴趣点的 content， 因此无法进行有效推荐。

### content-based 推荐关键技术

基于Content-based的推荐大多是通过分析网页上获取的文本来获得信息和特征，因此需要自然语言处理相关技术。Content-based的关键技术可以分为 item representation 和 user profile learning 两部分。

#### Item representation

Item representation 一般是分析从网页上获取的文本，常常遇到的问题有一词多义（一个词表示多个意思）、同义词（多个词表示同个意思）等问题。常用的 item representation 表示方式有基于关键字的向量空间模型、通过专业词库辅助分析语义等方法。

常用的基于关键字的向量空间模型有 TF-IDF、Word2Vector等方法。

##### TF-IDF

##### Word2Vector

通过专业词库辅助分析语义是为了给系统提供一个额外的专业知识背景，例如电影网站提供导演、演员、编剧姓名列表等信息。

#### User profile learning

User profile learning 可以看做一个文本二分类问题，每个 item 被分为感兴趣和不感兴趣两类，常用的方法有贝叶斯分类等方法。

贝叶斯公式目标是估计一个 item 属于哪个分类的概率比较高，二分类的先验概率已知，在分类中观察到 item 的概率为， item 的概率为，则后验概率 item 属于分类的概率为：

最后选择概率较大的类别，即为分类结果：

$$
c=\argmax{c\_j}\frac{P(c\_j)P(i|c\_j)}{P(i)}
$$

此外，决策树、KNN等数据挖掘算法也常用在 content-based推荐应用上。随着 User generated content 的增多，如今日头条等平台，基于自然语言处理相关技术的文档推荐、与社交网络相结合的标签推荐等方法越来越多。另外关于content-based serendipity 相关的研究也越来越多。

## Collaborative filtering

与content-based 只用了推荐 user 自身的历史的喜好数据不同，协同过滤也利用了其他用户的历史信息。协同过滤总体来说可以分为基于近邻和基于模型两类，其中基于近邻的方法也被叫做基于内存(memory-based)或者启发式(heuristic-based)。基于近邻的方法分为 user-based 和 item-based 两种。其中 User-based 方法是通过发现与之相似的 user 近邻，将近邻的喜好 item 推荐给 该user。而item-based模型则是通过发现与 user 历史上喜欢的 item 相似的 item，并将其推荐给该 user，关于相似度的衡量有多种方式。基于近邻的方法是直接利用保存的打分情况计算并推荐，与之不同的是 model-based 方法，是建模 user 和 item 之间的关系，学习到一个可以用来预测的模型，常用的用来推荐的模型有很多种，例如贝叶斯聚类、 Latent Semantic Analysis， Latent Dirichlet Allocation, 最大熵， Boltzman机，SVM，SVD分解等等。

基于近邻的协同过滤是最早的协同过滤，拥有简单、有效、稳定等优点。

### 基于近邻的协同过滤

User-based 协同过滤与 item-based 协同过滤同理，此处以 user-based 为例，介绍基于近邻的协同过滤。

常见的推荐系统应用场景为 rating 预测（连续），和分类(like & dislike，离散评分，离散问题)。

#### Rating 预测

User-based 协同过滤是通过找到与 user 最相近的 user 集合，利用他们对 item 的打分，预测 对的打分。假设 user 集合是给打分的 user 中与 user 最相近的 个用户，相似度为， 则我们估计的 对的打分情况如下：

其中我们把每个用户的打分都看成了一样的权重，但相似度高的用户显然应该比相似度低的用户影响更大，因此加权改进如下：

此外加权还可以采用 等方式。

此时计算方法中未考虑用户个人打分习惯，有的用户习惯打高分，有的则习惯打低分，可以做 normalize 。

#### 分类问题

将 rating 离散开，可以将问题看成一个分类问题，分类结果估计如下:

对于所有可能的评分，例如1、2、3、4、5,, if ,否则等于0。计算出得分最大的rating，则为最后的估计评分。

这种计算方法兼顾了相似度加权，同理考虑到不同人的打分习惯，也是可以先 normalize 一下。

将问题看成回归比看成分类更加保守，但分类更有可能带来惊喜度。

同理，user-based 换成 item，就形成了 Item-based 方法。

#### 相似度衡量

##### 归一化方法

* Mean-centering
* Z-score normalization h(r\_{ui})=

##### 相似度计算

* Cosine similarity
* Pearson Correlation
* Other similarity 还有一些其他的相似度度量方法，例如Adjusted Cosine, Mean Square Difference， Spearman Rank Correlation 等

#### 近邻选择

近邻选择有不同的方法，常用的几种方法如下：

* Top-N 选择， 选择相似度最高的N个近邻。
* Threshold，根据阈值选择N个近邻
* Negative filtering, 对 accuracy 没什么影响，是否使用根据数据集情况确定。

#### 近邻算法的问题

近邻算法存在一些问题，例如覆盖度低、对数据稀疏敏感等问题。

* 覆盖度低，近邻是基于两个用户有共同打分的item上进行的，因此具有一定的局限性，即使两个用户没有共同的item 打分，他们的喜好仍然可能相同，因此推荐 item 的覆盖度会有一些局限性。
* 对稀疏数据敏感，对于新加入的商品和用户，打分记录比较少，在基于紧邻的方法中存在冷启动问题。

为了解决这些问题，可以采用一些其他的方法避免，例如降维技术，基于图理论的方法等。

##### 降维技术

降维常用方法为隐空间特征分解，SVD分解等矩阵分解相关技术，可以直接将打分矩阵分解或者相似度矩阵分解得到隐空间表示，然后再通过内积计算两者的相似度。

##### 基于图的方法

基于图的方法主要将 user 和 item 表示成二分图，如下： [./images/bipartite-graph.png]

基于图的方法主要有基于路径的方法（如最短路径、路径数量等）、随机游走方法（Itemrank, 平均最先到达时间等）方法。

### Model-based

基于近邻的算法存在一些问题

#### 矩阵分解

##### SVD分解

##### SVD++

##### SVD free

##### NMF

##### PMF

#### 深度神经网络

##### MLP

##### AE

##### CNN

##### DNN

## 推荐系统的评价

## 总结