

# 推荐系统调研报告及综述

张永锋

清华大学计算机系人工智能研究所

zhangyf07@gmail.com

## 一、推荐系统概述

### 1.1 什么是推荐系统

推荐系统(Recommender System)的发展已经经历了近 20 年的时间,但是迄今为止仍没有人试图对推荐系统给出一个精确的定义。

广义上的推荐系统可以理解为是主动向用户推荐物品(Item)的系统,所推荐的物品可以是音乐、书籍、餐厅、活动、股票、数码产品、新闻条目等等,这依赖于具体的应用领域,推荐系统所推荐的物品或者对用户有帮助,或者用户可能感兴趣[1]。

随着电子商务规模的不断扩大,商品数量和种类不断增长,用户对于检索和推荐提出了更高的要求。由于不同用户在兴趣爱好、关注领域、个人经历等方面的不同,以满足不同用户的不同推荐需求为目的、不同人可以获得不同推荐为重要特征的个性化推荐系统(Personalized Recommender System)应运而生[1]。目前所说的推荐系统一般指个性化推荐系统。

### 1.2 推荐系统的发展历史

如果追根溯源,推荐系统的初端可以追溯到函数逼近理论、信息检索、预测理论等诸多学科中的一些延伸研究。

推荐系统成为一个相对独立的研究方向一般被认为始自 1994 年明尼苏达大学 GroupLens 研究组推出的 GroupLens 系统[2]。该系统有两大重要贡献:一是首次提出了基于协同过滤(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。

GroupLens 所提出的推荐算法实际上就是目前人们时常提及的基于用户的协同过滤推荐算法(User-based Collaborative Filtering Algorithms),虽然论文本身并没有使用这样一个名字。在之后的十几年中,其它一些著名的协同过滤算法逐渐被提出,主要的有基于物品的协同过滤算法(Item-based Collaborative Filtering Algorithms)[3],基于矩阵分解的协同过滤算法(SVD-based/NMF-based, etc.)等等。当然,基于其它方法而非协同过滤的推荐算法也在不断地发展,这些方法之间的互补、融合也成为重要的研究方向,这些会在本文第二部分(推荐方法的分类)和第三部分(典型推荐算法概述及优缺点)中详细阐述。

目前,推荐算法已经被广泛集成到了很多商业应用系统中,比较著名的有 Netflix 在线视频推荐系统、Amazon 网络购物商城等。实际上,大多数的电子商务平台尤其是网络购物平台,都不同程度地集成了推荐算法,如淘宝、京东商城等。Amazon 发布的数据显示,亚马逊网络书城的推荐算法为亚马逊每年贡献近三十个百分点的创收。

## 1.3 推荐系统的输入输出

### 1.3.1 推荐系统的输入数据

推荐系统可能的输入数据多种多样，但是归纳起来可以分为用户(User)、物品(Item)和评价(Review)三个层面，它们分别对应于一个矩阵中的行、列、值。

#### 1.3.1.1 物品(Item)

用来描述一个 Item 的性质，也经常被称为 Item Profile。根据 item 的不同，Item Profile 也是不尽相同的。比如对于图书推荐，Item Profile 有可能包括图书所属类别、作者、页数、出版时间、出版商等；对于新闻推荐，Item Profile 则有可能是新闻的文本内容、关键词、时间等；而对于电影，可以是片名、时长、上映时间、主演、剧情描述等。

#### 1.3.1.2 用户(User)

用来描述一个用户的“个性”，也就是 User Profile。根据不同的应用场景以及不同的具体算法，User Profile 可能有不同的表示方式。

一种直观且容易理解的表示方式与 Item Profile 类似，比如该用户的性别、年龄、年收入、活跃时间、所在城市等等。但是在推荐系统中，这样的 profile 很难集成到常见的算法中，也很难与具体的 item 之间建立联系(比如我们很难断定某商品一定不会被某年龄段的人喜欢，这样的判断过于粗糙)，因此这种 User Profile 在推荐系统中虽然也经常会被使用，但是很少直接用在推荐算法中，而是用于对推荐结果进行过滤和排序。

由于在很多推荐算法中，计算 User Profile 和 Item Profile 之间的相似度是一个经常会用到的操作，另一种使用更为广泛也更有实际意义的 User Profile 应运而生[4]。它的结构与该系统中的 Item Profile 的结构一样，为了更清楚地说明其结构，我们以一种典型的构建 User Profile 的方法为例来进行说明：考虑该 User 打过分的所有 Item，将这些 Item 的 Item Profile 的每一项分别进行加权平均，得到一个综合的 Profile，作为该用户的 User Profile。这种 User Profile 的优点是很容易计算其与 Item 之间的相似度，同时比较准确地描述了该用户在 Item 上的偏好，巧妙地避开了用户私人信息这一很难获得的数据，具有保护隐私的能力，进一步，如果加入时间因素，还可以研究用户在 Item 上偏好的变化等等，因此受到广泛应用。

#### 1.3.1.3 评价(Review)

评价是联系一个 User 与一个 Item 的纽带，最简单的 Review 是 User 对某一 Item 的打分(Rate)，表示了该 User 对该 Item 的喜好程度。在常见的推荐算法中，这是一个 1~5 的整数。当然，用户对物品或信息的偏好，根据应用本身的不同，还可能包含很多不同的信息，比如用户对商品的评论文本、用户的查看历史记录、用户的购买记录等，这些信息总体上可以分为两类：一是显式的用户反馈，这是用户对商品或信息给出的显式反馈信息，评分、评论属于该类；另一类是隐式的用户反馈，这类一般是用户在使用网站的过程中产生的数据，它们也反映了用户对物品的喜好，比如用户查看了某物品的信息，用户在某一页面上的停留时间等等。

虽然目前大多数的推荐算法往往都是基于用户评分矩阵(the Rating Matrix)的,但是基于用户评论、用户隐式反馈数据的方法来完成推荐越来越受到人们的关注,这些方面的研究长期以来受到文本挖掘、用户数据收集等方面的难点的制约,没有得到充分的研究,但是它们在解决推荐系统的可解释性、冷启动问题等方面确实具有重要的潜力[5][6][7]。

### 1.3.2 推荐系统的输出数据

对于一个特定的用户,推荐系统给他的输出是一个“推荐列表”,该推荐列表按照优先级的顺序给出了对该用户可能感兴趣的物品。

对于一个实用的推荐系统而言,仅仅给出推荐列表往往是不够的,因为用户不知道为什么系统给出的推荐是合理的,进而也就不太会采纳系统给出的推荐。为了解决这个问题,推荐系统另一个重要的输出是“推荐理由”,它表述了系统为什么认为推荐该物品是合理的,如“购买了某商品的用户有 90%也购买了该商品”等等。

为了解决推荐合理性的问题,推荐理由在产业界被作为一个重要的吸引用户接受推荐物品的方法,在学术届也受到越来越多的关注[9]。

## 1.4. 推荐问题的形式化

这里给出推荐问题一个最典型的形式化,如上所述,该形式化方法来最早自于 GroupLens [2],并在[12]中做了进一步的阐述。

首先我们拥有一个大型稀疏矩阵,该矩阵的每一行表示一个 User,每一列表示一个 Item,每一个数值表示该 User 对该 Item 的打分,这是一个 0~5 的分值,0 表示该 User 未曾对该 Item 打分,1 表示该 User 对该 Item 最不满意,5 表示该 User 对该 Item 最满意。视具体情况,对于每一个 User,可能有其对应的 User Profile,对于每一个 Item,可能有其对应的 Item Profile,如上所述。

我们现在解决这样一个问题:给定该矩阵之后,对于某一个 User,向其推荐哪些他没有打过的 Item 最容易被他接受,这里的“接受”根据具体的应用环境有所不同,有可能是查看该新闻、购买该商品、收藏该网页等等。

对于推荐算法,还需要一系列的评价指标来评价推荐的效果,这些评价方法和评价指标将在第四部分具体说明。

## 1.5. 推荐系统的两大核心问题

有了如上的形式化描述之后,推荐系统所要解决的具体问题总体有两个,分别是预测(Prediction)和推荐(Recommendation)

“预测”所要解决的主要问题是推断每一个 User 对每一个 Item 的喜好程度,其主要手段是根据已有的信息来计算 User 在他没打分的 Item 上可能的打分,计算结果当然是越准确越好,至

于如何来描述和评价预测的准确度，将会在后面的第四部分说明。

“推荐”所要解决的主要问题是根据预测环节所计算的结果向用户推荐他没有打过分的 Item。由于 Item 数量众多，用户不可能全部浏览一遍，因此“推荐”的核心步骤是对推荐结果的排序(Ranking)。当然，按照预测分值的高低直接排序确实是一种比较合理的方法，但是在实际系统中，Ranking 要考虑的因素很多，比如用户的年龄段、用户在最近一段时间内的购买记录等，第一类 User Profile 往往在这个环节派上用场。

虽然人们早就意识到“预测”和“推荐”作为推荐系统的两大核心问题都具有重要的作用，但是目前绝大多数的推荐算法都把精力集中在“预测”环节上，多数论文在给出对预测效果的评估后就结束了。“推荐”作为重要的后续环节需要更多的研究，这与搜索引擎的发展非常类似。目前，推荐多样性[8]、推荐界面等很多方面的研究也在受到越来越多的关注。

## 二、推荐方法的分类

按照不同的分类指标，推荐系统具有很多不同的分类方法，常见的分类方法有依据推荐结果是否因人而异、依据推荐方法的不同、依据推荐模型构建方式的不同等。

### 2.1 依据推荐结果是否因人而异

主要分为大众化推荐和个性化推荐。大众化推荐往往与用户本身及其历史信息无关，在同样的外部条件下，不同用户获得的推荐是一样的。大众化推荐一个典型的例子是查询推荐，它往往只与当前的 query 有关，而很少与该用户直接相关。个性化推荐的特点则是不同的人在这样的外部条件下，也可以获得与其本身兴趣爱好、历史记录等相匹配的推荐，前面已经有所介绍，这里不再详细阐释。

### 2.2 依据推荐方法的不同

也就是考虑如何发现数据的相关性：大部分的推荐系统其工作原理还是基于物品或用户的相似性进行推荐，大致上可以分为如下几种：基于人口统计学的推荐(Demographic-based Recommendation)[10]，基于内容的推荐(Content-Based Recommendation)[11]，以及基于协同过滤的推荐(Collaborative Filtering-Based Recommendation)，以及混合型推荐系统(Hybrid Recommendation)[16]。

其中基于协同过滤的推荐被研究人员研究的最多也最为深入，它又可以被分为多个子类别，分别是基于用户的推荐(User-Based Recommendation)[2]，基于物品的推荐(Item-Based Recommendation)[3]，基于社交网络关系的推荐(Social-Based Recommendation)[13]，基于模型的推荐(Model-based Recommendation)等。

基于模型的推荐是指利用系统已有的数据，学习构建一个模型，进而利用该模型进行推荐，这里的模型可以是 SVD、NMF 等矩阵分解的结果[14]，也可以是利用贝叶斯分类器、决策树、人工神经网络等模型转化为分类问题，或者基于 Co-Clustering 等聚类技术对数据进行预处理的结果[15]等等。

这些方法会在后面的第三部分加以介绍和讨论。

## 2.3 依据推荐模型构建方式的不同

可以分为基于用户或物品本身的启发式推荐(Heuristic-Based,又称为 Memory-Based Recommendation)、基于关联规则的推荐(Association Rule Mining for Recommendation)[17], 基于模型的推荐(Model-based Recommendation), 以及混合型推荐系统(Hybrid Recommendation)。

## 三、典型推荐算法概述及优缺点

### 3.1 基于人口统计学的推荐(Demographic-Based Recommendation)

虽然这种方法已经很少被单独使用, 但是理解这种方法的工作原理对于深入理解推荐系统有很大帮助。

Demographic-Based 方法对应于之前提到的第一种 User Profile, 也就是每一个用户的 User Profile 中记录了该用户的性别、年龄、活跃时间等元数据。该方法所基于的基本假设是“一个用户有可能会喜欢与其相似的用户所喜欢的物品”。当我们需要对一个 User 进行个性化推荐时, 利用 User Profile 计算其它用户与其之间的相似度, 然后挑选出与其最相似的前 K 个用户, 之后利用这些用户的购买和打分信息进行推荐。一种简单且常见的推荐方法就是将这些 User 所覆盖的 Item 作为推荐列表, 并以 Item 在这些 User 上得分的平均值作为依据来进行排序, 并将该排好序的推荐列表提供给用户。

这种 Demographic-Based 推荐方法的优点是计算简单, 由于 User Profile 相对比较固定, 因此用户相似度的计算可以线下完成, 便于实现实时响应。但是它也有诸多的问题, 一个主要的问题是计算可信度比较低, 其根本原因在于即便是性别、年龄等属性都相同的用户, 他们也很可能在物品上有截然不同的偏好, 因此这种计算用户相似度的方法并不能真正与物品之间建立联系。另一个主要的缺点在于推荐结果的可解释性不够, 该方法对于推荐结果的解释方法为“与你具有相似属性的用户也购买了该物品”, 但这显然难以让用户信服——我怎么知道他和我具有相似的属性? Demographic-Based 方法是一种非协同过滤式的方法, 但是它与后面基于内容的推荐(User-Based)这一协同过滤方法很相似, 将在后面介绍。

### 3.2 基于内容的推荐(Content-Based Recommendation)

前面基于人口统计学的推荐对应于第一种 User Profile, 而基于内容的推荐则对应于第二种 User Profile。Content-Based 方法所基于的基本假设是“一个用户可能会喜欢和他曾经喜欢过的物品相似的物品”。这里“曾经喜欢过的物品”就是利用该用户的历史记录计算出来的 Profile, 作为该用户的 User Profile 来使用。

典型的 Content-Based 方法的第一步是构建 User Profile, 一种较为简单的构建方法是: 考虑该 User 曾经打过分的所有 Item, 将这些 Item 的 Item Profile 做一个加权平均, 作为该 User 的 User Profile。当然, 构建 User Profile 的策略可以很复杂, 比如可以考虑时间因素, 计算 User 在不同时间段内的 Profile, 从而了解 User 在历史数据上所表现出来的偏好的变化等等。有了 User Profile 之后, 我们就可以开始推荐了, 最简单的推荐策略就是计算所有该用户未尝试过的 Item 与该用

户的 User Profile 之间的相似度，并按照相似度由大到小的顺序生成推荐列表，作为结果输出。当然，推荐策略也可以很复杂，比如在数据源上考虑本次用户交互过程中所收集到的即时交互数据来决定排序，在模型上使用决策树、人工神经网络等等，但是这些方法最核心的环节都是利用 User Profile 和 Item Profile 之间的相似度计算。

其实在很多 Content-Based 的推荐算法中，并不把 User Profile 显式地计算出来，而是利用 User 打过分的 Item，直接计算推荐列表，一种直观的方法是：对于一个该 User 未尝试过的 Item，计算它与该 User 尝试过的所有 Item 之间的相似度，并将这些相似度根据 User 的打分进行加权平均。这本质上也是 Content-Based 方法，只是绕过了计算 User Profile 的环节。实际上，很多具体的应用表明，绕过 User Profile 的计算，直接利用 Item Profile 计算相似度，往往更灵活，推荐效果更好，因为在计算 User Profile 的过程中，一些有用的信息被丢掉以至于无法在后面的环节中被利用。

Content-Based 方法也是一种非协同过滤式的方法，同样，它与后面 Item-Based 这种协同过滤式的方法很相似，后面将做具体介绍。

Content-Based 方法的优点主要有：对新加入的物品没有冷启动(Cold-Start)的问题，因为只要有该物品的 Item Profile，就可以直接计算它与其它 Item 之间的相似度，不受打分数据稀疏的限制；推荐结果具有较好的可解释性，一种显然的推荐理由是“该物品与你之前曾经喜欢过的某物品相似”。

但是 Content-Based 方法也有一些缺点，首先是系统需要复杂的模块甚至手工来预处理物品信息以得到能够代表它们的特征，然后受信息获取技术的制约、处理对象的复杂性高等因素，这样工作难以达到较好的效果；其次，这种方法无法用户并不熟悉但是具有潜在兴趣的物品，因为该方法总是倾向于向用户推荐与其历史数据相似的物品；另外就是可扩展性不好，需要针对不同的领域构建几乎完全不同的 Profile，针对一个数据集合训练的模型未必适合其它的数据集合，并且随着新物品的加入，模型的可用性也值得商榷。

### 3.3 基于协同过滤的推荐(Collaborative Filtering-Based Recommendation)

是指收集用户过去的行为以获得其对产品的显式或隐式信息，即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、或用户的相关性，然后再基于这些关联性进行推荐。根据前文所述，基于协同过滤的推荐可以分基于用户的推荐 (User-based Recommendation)，基于物品的推荐 (Item-based Recommendation)，基于模型的推荐 (Model-based Recommendation) 等子类。用户对物品的喜好或评分矩阵往往是一个很大的稀疏矩阵，为了减少计算量，可采用对物品或用户进行聚类(Clustering items for Collaborative Filtering)[18]。

#### 3.3.1 memory-based 推荐

User-Based 方法和 Item-Based 方法都属于 memory-based 方法类别，即基于系统中的历史记录（如所有被打过分的物品）进行预测。

##### 3.3.1.1 基于用户的推荐(User-Based Recommendation)

User-Based 方法是最早的一种基于协同过滤的推荐算法[2]，它所基于的基本假设与 Demographic 方法类似，也是“用户可能会喜欢和他具有相似爱好的用户所喜欢的物品”，与之不同的是，这里的“相似用户”，不是用 User Profile 计算出来的，而是用用户的打分历史记录计算出来的。这里的基本想法是，具有相似偏好的用户，他们在所有 Item 上的打分情况(哪些打了份，哪些没打分；打分的分值是多少等)也是相似的。

User-Based 推荐方法的第一步是执行最近邻搜索，我们把每一个 User 看成一个行向量，对于一个 User，我们计算其它所有的 User 行向量与他的相似度。这里的相似度计算有很多不同的指标，比如 Pearson 相关性系数、余弦相似度等，这些指标及其比较将在后面第五部分给出。当我们拥有了每一个用户与该 User 的打分相似度之后，就可以执行预测和推荐了，最常见的方法是选取与该用户相似度最高的前 K 个用户，用这 K 个用户的打分情况对该 User 的打分进行预测，进而进行推荐。

最常见也最简单的推荐方法有两种，分别为 Top-N 推荐和关联推荐。Top-N 推荐是统计在这前 K 个用户中，统计出现频率最高且在目标用户的向量中未出现的物品，将这些物品构建成推荐列表作为输出。关联推荐的基本思想是，利用这前 K 个用户的购买或打分记录进行关联规则挖掘，利用挖掘出的关联规则，结合目标用户的购买记录完成推荐，典型的推荐结果就是我们常见的“购买了某物品的用户还购买了什么物品”。

在 User-Based 方法中，“个性化”的体现方式是对于不同的 User，其最近邻是不同的，进而得到的推荐列表也不一样；“协同过滤”的体现方式是，对一个用户进行推荐时，使用其它用户在物品上的观点来寻找和他相似的用户，这是与 Demographic 方法不同的地方。

User-Based 方法的优点在于，在数据集完善、内容丰富的条件下，能够获得较高的准确率，而且能够避开物品 Item Profile 上的挖掘进行推荐，能够对物品的关联性、用户的偏好进行隐式透明的挖掘。

其缺点在于，随着系统用户数量的增大，计算 Top-K Relevance User 的时间会显著增长，使得该方法难以胜任用户量变化巨大的系统，限制了系统的可扩展性。另外新用户的问题也是 User-Based 方法难以处理的问题：当新用户加入系统时，由于他的打分历史记录很少，难以准确计算真正与之对应的 Top-K Relevance User，这也进一步引出数据稀疏性的问题对系统可扩展性的限制。

### 3.3.1.2 基于物品的推荐(Item-Based Recommendation)

鉴于 User-Based 协同过滤方法随着用户数增加，计算时间显著增长，Sarwar 等人在 WWW 2001 上提出了基于物品的推荐[3]，Item-Based 方法是亚马逊网络商城的专利算法，目前也是亚马逊网络商城推荐系统的底层核心算法。Item-Based 方法所基于的基本假设与 Content-Based 方法类似，也就是“用户可能会喜欢与他之前曾经喜欢的物品相似的物品”。比如喜欢《长尾理论》的人，也很可能去看《世界是平的》。与 Content-Based 方法不同的是，这里的“相似物品”不是通过 Item Profile 来计算的，而是通过物品被打分的历史记录来计算的。

Item-Based 推荐方法的第一步与 User-Based 一样，也是执行最近邻搜索，我们把每一个 Item 看成一个列向量，对于一个 Item，我们计算其它所有的 Item 列向量与它的相似度。有了 Item 之

间的两两相似度之后，就可以进行预测和推荐了。

最简单的线上推荐是，当用户购买了某一商品后，直接向其推荐与该物品相似度最高的前 K 个商品。稍微复杂一点的推荐方法考虑了该用户所有的打分历史记录[3]：对于一个 User 行向量中的 0 值(用户未购买的物品)，我们需要去预测该用户在该物品（记为 I）上可能的打分，预测方法是，考虑所有该用户已经打过的 Item，以它们与 I 的相似度为权重，对它们的分值进行加权平均，作为对 I 的预测打分，最终以预测打分的高低为顺序给出推荐列表。

Item-Based 方法总体来说是一种启发式的方法(Heuristic-Based)，对目标的拟合能力是有限的，但是当把多个启发式方法结果(以及其他的特征)结合起来，也可以有很好的拟合能力[3]。

Item-Based 方法的优点主要有以下几点：一是计算简单，容易实现实时响应，由于在常见的系统中，Item 被打分的变化剧烈程度要比 User 低得多，因此 Item 相似度的计算一般可以采用离线完成、定期更新的方式，从而减少了线上计算，实现实时响应，提高效率，尤其对于用户数远大于商品数的情况下效果更加显著，比如用户新添加了几个感兴趣的物品之后，可以立即给出新的推荐；二是可解释性好，用户可能不了解其他人的购物情况，但是对自己的购物历史总是很清楚的，另外用户总是希望自己有最后的决定权，如果系统推荐的物品不满意，需要有办法让用户改进它，Item-based 方法很容易让用户理解为什么推荐了某个物品，并且当用户在兴趣列表里添加或删除物品时，可以调整系统的推荐结果，这也是其他方法最难做到的一点。

Item-Based 方法的缺点主要有：以物品为基础的系统过滤较少考虑用户之间的差别，因此精度较 User-Based 方法稍微逊色一点；另外还有许多其它的问题有待解决，最典型的就数据稀疏性(Sparsity)和冷启动(Cold-Start)的问题，对于 Cold-Start 问题的解决，目前有了最新的进展，SIGIR2011 的一篇论文[19]对此进行了深入的讨论。

### 3.3.1.3 User-Based vs Item-Based

这里将对 User-Based 和 Item-Based 两种最基本的协同过滤方法进行对比：

计算复杂性：User-Based 是最早的协同过滤方法，如前所述，在 1994 年的论文[2]中被提出，当时还没有 User-Based 和 Item-Based 的概念，人们很自然地首先想到了基于 User 的方法。Item-Based 方法是从 Amazon 的论文和专利发表之后（2001 年）开始流行的，且它被提出的一个主要背景就是解决 User-Based 方法在线计算量大，难以实时响应的问题。对于一个用户数量大大超过物品数量而物品数据相对稳定的应用，往往 Item-Based 方法从性能和复杂度上都比 User-Based 方法更优，这是因为计算物品的相似度不但计算量较小，而且不必频繁更新；而对于诸如新闻，博客或者微内容等物品数量巨大且更新频繁的应用中，User-Based 方法往往更具优势，推荐系统的设计者需要根据自己应用的特点选择更加合适的算法。

适用场景：在非社交网络的网站中，内容内在的联系是很重要的推荐原则，它比基于相似用户的推荐原则更加有效。比如在购书网站上，当用户看一本书的时候，推荐引擎会给用户推荐与其相关的书籍，这个推荐的重要性远远超过了网站首页对该用户的综合推荐。可以看到，在这种情况下，Item-Based 的方法成为了引导用户浏览的重要手段。同时 Item-Based 方法便于为推荐做出解释，在一个非社交网络的网站中，如果给某个用户推荐一本书时给出的解释是某个和你有相似兴趣的人也看了这本书，是很难让用户信服的，因为用户可能根本不认识那个人；但如果解



释为这本书和你以前看的某本书相似，用户就更有可能觉得而采纳该推荐。相反，在社交网站中，User-Based 方法，以及基于 User Network 和 Influence Network 的 User-Based 方法，则是更不错的选择，因为 User-Based 方法加上社会网络信息，可以大大增加用户对推荐解释的信服度。

推荐多样性和精度：研究人员在相同的数据集合上分别用 User-Based 和 Item-Based 方法计算推荐结果，发现在得到的推荐列表中，只有约 50%是一样的，还有约 50%完全不同，但是这两个算法却有相似的精度，可以说这两个算法是很互补的。关于推荐的多样性，有两种度量方法：一是从单个用户的角度来看，就是说给定一个用户，查看系统给出的推荐列表是否多样，也就是比较推荐列表中的物品之间两两的相似度。对于这种度量方法，Item-Based 的多样性显然不如 User-Based 好，因为 Item-Based 推荐本身就是和该用户的历史记录最相似的；第二种度量方法是考虑系统的多样性，也就是覆盖率(Coverage，有的论文也称召回率)，它是指一个推荐系统是否能够提供给所有用户丰富的选择(从整个系统的角度来看)。在这种指标下，Item-Based 方法的多样性要远远好于 User-Based 方法，因为 User-Based 方法总是倾向于推荐热门的物品，而从 Item-Based 方法有较好的新颖性，擅长推荐长尾里的物品(只要有那么一些用户，它同时购买了两个冷门的物品，这两个冷门物品之间就有较高的相似度)。所以，尽管大多数情况下 Item-Based 方法的精度略小于 User-Based，但如果考虑多样性，Item-Based 却比 User-Based 好很多。

用户对推荐算法的适应度：User-Based 方法推荐的原则是假设用户会喜欢那些和他有相同喜好的用户所喜欢的东西，但如果一个用户没有相同喜好的朋友，那么 User-Based 算法的效果就会比较差，所以一个用户对 User-Based 算法的适应度和他有多少共同喜好用户正相关；Item-Based 方法的基本假设是用户会喜欢和他以前喜欢的东西相似的东西，我们可以计算一个用户所喜欢的物品的自相似度，一个用户喜欢物品的自相似度大，就说明他的偏好相对比较单一、稳定，也就是说他比较符合 Item-Based 方法的基本假设，那么他对 Item-Based 方法的适应度就比较好，相反，如果自相似度小，就说明这个用户的偏好并不单一，那么 Item-Based 方法对他做出好的推荐的可能性就比较低，这种情况下，一般用对该用户所喜欢的物品进行聚类进而进行分层推荐的方法来解决。

在一个综合的推荐系统中，一般很少只用一种推荐策略，考虑到 User-Based 和 Item-Based 方法的互补性，很多推荐系统将两者结合起来作为系统的基础推荐算法。

### 3.3.2. 基于模型的推荐(Model-Based Recommendation)

User-Based 或 Item-Based 方法共有的缺点是资料稀疏，难以处理大数据量下的即时结果，因此发展出以模型为基础的协同过滤技术：先用历史数据训练得到一个模型，再用此模型进行预测。

以模型为基础的协同过滤广泛使用的技术包括 Latent Semantic Indexing、Bayesian Networks 等，收集打分数据进行分析和学习并推断出用户行为模型，进而对某个产品进行预测打分(e.g. 以 Item Profile 和 User Profile 中的各个特征的得分为输入，用户打分为输出，作为一个函数，用拟合或回归的方法来学习出一个模型，或者将打分作为类别转化为一个多分类器的问题等等)。这种方式不是基于一些启发规则进行预测计算，而是对于已有数据应用统计和机器学习得到的模型进行预测。

这种方法的优点在于快速相应：只要训练出了模型，就可以对新用户或新物品进行实时快速计算，其问题在于如何将用户实时或者近期新增的喜好信息反馈给训练好的模型，从而提高推荐的准确度，也就是模型的增量训练问题。

### 3.3.3 混合型推荐方法(Hybrid-Based Recommendation)

混合推荐系统是推荐系统的另一个研究热点，它是指将多种推荐技术进行混合相互弥补缺点，从而可以获得更好的推荐效果。最常见的是将协同过滤技术和其他技术相结合，以克服 Cold-Start 的问题。常见的混合方法有如下几种[20]：

**加权融合(Weighted)：**就是将多种推荐技术的计算结果加权混合产生推荐，最简单的方式是基于感知器的线性混合，首先将协同过滤的推荐结果和基于内容的推荐结果赋予相同的权重值，然后比较用户对物品的评价与系统的预测是否相符，进而不断调整权值。

**切换(Switch)：**根据问题背景和实际情况采用不同的推荐技术。比如，使用基于内容推荐和协同过滤混合的方式，系统首先使用基于内容的推荐技术，如果它不能产生高可信度的推荐，然后再尝试使用协同过滤技术。因为需要各种情况比较转换标准，所以这种方法会增加算法的复杂度和参数化，当然这样做的好处是对各种推荐技术的优点和弱点比较敏感。

**混合(Mix)：**将多种不同的推荐算法推荐出来的结果混合在一起，其难点是如何重排序。

**特征组合(Feature Combination)：**将来自不同推荐数据源的特征组合起来，由另一种推荐技术采用。一般会将协同过滤的信息作为增加的特征向量，然后在这增加的数据集上采用基于内容的推荐技术。特征组合的混合方式使得系统不再仅仅考虑协同过滤的数据源，所以它降低了用户对物品评分数量的敏感度，相反，它允许系统拥有物品的内部相似信息，其对协同系统是不透明的。

**级联型(Cascade)：**用后一个推荐方法优化前一个推荐方法。它是一个分阶段的过程，首先用一种推荐技术产生一个较为粗略的候选结果，在此基础上使用第二种推荐技术对其作出进一步精确地推荐。级联型允许系统对某些项避免在后面低优先级的推荐器中被过滤掉，这些项可能是通过第一种推荐技术被较好的予以区分了的，或者是很少被用户评价从来都不会被推荐的物品。因为级联型的第二步，仅仅是集中在需要另外判断的项上。

**特征递增(Feature Augmentation)：**前一个推荐方法的输出作为后一个推荐方法的输入，它与级联型的不同之处在于，这种方法上一级产生的并不是直接的推荐结果，而是为下一级的推荐提供某些特征。一个典型的例子是将聚类分析环节作为关联规则挖掘环节的预处理：聚类所提供的类别特征，被用于关联规则挖掘中，比如对每个聚类分别进行关联规则挖掘。

**元层次混合(Meta-level hybrid)：**将不同的推荐模型在模型层面上进行深度的融合，而不仅仅是把一个输出结果作为另一个的输入。比如，User-Based 方法和 Item-Based 方法的一种组合方式是，先求目标物品的相似物品集，然后删掉所有其它的物品(在矩阵中对应的是列向量)，在目标物品的相似物品集上采用 User-Based 协同过滤算法。这种基于相似物品的邻居用户协同推荐方法，能很好地处理用户多兴趣下的个性化推荐问题，尤其是候选推荐物品的内容属性相差很大

的时候，该方法性能会更好。

## 四、推荐系统的评价方法和指标

### 4.1 评价方法

总体来讲，评估推荐系统的实验方法有三种，分别是离线评测、在线评测和用户调研。

离线评测通过把数据集分成训练集和测试集，在训练集上学习和调整模型参数，在测试集上进行测试，计算精确度、运行效率等指标，达到评测的目的。这种方法实施简单方便，只需要收集数据，不需要与用户的交互。但能说明的问题较少，比如不能评估出推荐系统对用户行为的影响。这种方法一般用以衡量推荐算法在预测环节中的能力，淘汰掉预测不准确的算法。离线评测的具体评价指标在后面介绍。

推荐系统的实际效果依赖于多种因素，如用户意图、用户背景知识、对系统的熟悉度、信任度、用户界面等，这些因素在离线评中不易重现，需要线上评测来完成，一般线上评测可以使用 AB test 的方法。

线上评测和用户调研的方法在人机交互领域对推荐系统的研究中比较常用。

### 4.2 评价指标

所谓评价指标主要包括“技术评价指标”和“业务指标”。技术评价指标包括诸如 RMSE、NDCG、MAP、Recall、Precision 等，业务指标如成交转化率、用户点击率等，这里主要对技术评价指标进行总结概括[21]。

一个比较有意思的事情是，Hulu 在实验报告“Do clicks measure recommendation relevancy”中讨论了点击率是否适用于评测推荐系统，报告认为在搜索领域被广泛认可或验证了的 Position Bias 假设(排在靠前位置的搜索结果得到的点击会比靠后位置的结果多得多)并不适用于推荐系统，他们的实验表明推荐产品的排列位置对点击影响甚微，因此在以 NDCG 为指标的离线测评中性能好的算法，在在线测评中点击率有可能反而比较低。

目前评估推荐系统的指标可分为“准确度（accuracy）”与“可用性（usefulness）”两种。其中准确度衡量的是推荐系统的预测结果与用户行为之间的误差，还可以再细分为“预测准确度（Prediction Accuracy）”和“决策支持准确度（Decision-Support Accuracy）”。而预测准确度又可分为“评分预测准确度”、“使用预测准确度”、“排序准确度”等，以 MAE、NMAE、RMSE、ARMSE 等为常用的统计指标，计算推荐系统对消费者喜好的预测与消费者实际的喜好间的误差平均值；而决策支持准确度则以 Correlation（关联度，包括：Pearson、Spearman、Kendall Tau 等）、Reversal Rate、Precision、Recall、F-measure、ROC curve（Receiver Operating Characteristic）、SwetA 指标（曲线下面积比例）、AVS（Average Ranking Score，平均排序分）等为主要工具。

#### 4.2.1 评分预测准确度（Ratings Prediction Accuracy）

RMSE (Root Mean Squared Error, 根均方差) 是最流行的度量, 预测的是用户对每个商品感兴趣的程度, 优化 RMSE, 实际上就是要预测用户对每个商品的评分。

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}$$

比如 Netflix Prize 竞赛, 就是以 RMSE 为度量, 竞赛者比 Netflix 公司之前使用的推荐系统 CineMatch 低百分之十, 就可获得百万大奖 (团队 BPC, BellKor's Pragmatic Chaos 获奖)。

另一个比较常用的度量是 MAE (Mean Absolute Error, 平均绝对误差)

$$\text{MAE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} |\hat{r}_{ui} - r_{ui}|}$$

两者相比, RMSE 对大误差比较敏感。

与之类似的有 NRMSE (Normalized RMSE) 和 NMAE (Normalized MAE): 两者的归一化版本, 例如可以除以 MaxRate-MinRate 进行归一化。针对物品或用户在测试集中分布不均的情况可以使用 Average RMSE 和 Average MAE, 例如对于物品分布不均时, 单独计算每个物品的 RMSE 然后再平均; 对于用户分布不均时单独计算每个用户的 RMSE 然后再平均。

RMSE、MAE 仅度量误差幅度, 容易理解且计算方式也不复杂, 但其缺点也正是失之于简单, 过度简化事实, 在有些场合可能不能说明问题。例如用户打分 1、2 的差别和打分 4、5 的差别都是 1, 但意义不一样。在这种情况下可以定义适当的扭曲程度度量 distortion measure  $d$  来代替差值以改进度量方法。

与之类似的还有 MAP (平均准确率, Mean Average Precision), 评估用户期望的相关结果是否尽可能排在前面。MAP 是信息检索中解决 PRF 指标的不足而提出的, 单个主题的平均准确率是每篇相关文档检索出后的准确率的平均值, 主集合的平均准确率(MAP)是每个主题的平均准确率的平均值。MAP 是反映系统在全部相关文档上性能的单值指标。系统检索出来的相关文档越靠前(rank 越高), MAP 就可能越高。

$$\text{MAP} = \frac{1}{|U|} \sum_{i=1}^{|U|} \frac{1}{|R_i|} \sum_{j=1}^{|R_i|} \frac{j}{r_{ij}}$$

#### 4.2.2 使用预测准确度 (Usage Prediction)

对评分类的应用可以使用 Reversal Rate: 首先选择 High 和 Low 两个基准数值, 然后比较推荐预测与消费者所给的评分, 如果消费者的评分大过 High 这个参考基准, 但是推荐系统的预测是小于 Low, 我们叫这种情况是 High Reversal; 反之, 则称这种情况为 Low Reversal。计算 Reversal Rate 可以反映这个系统产生的预测的准确程度。

而对于选择类的应用, 可以使用源自信息检索 (Information Retrieval) 中的 Precision-Recall。评分类的应用也可将评分按某阈值分为“选择”与“不选择”后应用此度量方法。

对于推荐数量预定的场合可使用 P@N (Precision at N) 度量, 而推荐数量未指定的使用 PR 曲线或 ROC 曲线, 描述的是喜好的物品被推荐的比例。其中 PR (Precision-Recall) 曲线强调推

荐的物品有多少是喜好的，ROC（Receiver Operating Characteristic）曲线则强调有多少不喜好的物品却被推荐的。与其对应的曲线有 AUC（Area Under the ROC Curve）等（AUC 越大，表示系统能够推荐出越多的好的物品）。

已经有研究人员验证，大部分状况下计算 ROC 和 Precision-Recall 时，会得到相同的 Confusion Matrix，而且从其中一个曲线可以推演出另外一种曲线的状况。不过 PR 曲线比较适合于数据分布高度不平均（highly-skewed）的情况，因此在实际应用中要根据推荐系统选择相应的评估方式。

#### 4.2.3 可用性(Usefulness)

推荐系统的预测准确度当然是评估的重要指标，但是“准确度”不是唯一的标准。推荐系统的可用性一般用覆盖率(召回率)来描述，包括用户覆盖率和物品覆盖率。为了描述推荐效果的多样性(Diversity)，也经常采用 Gini Index 或 Shannon Entropy 等常见的指标，这里不再详述。

除此之外，推荐系统也考虑很多其它的可用性指标，这些指标对于推荐系统的用户体验具有重要的意义。主要有：新颖性 novelty，惊奇性 Serendipity，实用性 Utility，风险 Risk，鲁棒性 Robustness，隐私性 privacy，自适应性 Adaptivity（例如 item 的时效性等），可扩展性 Scalability，推荐效率 Efficiency，可解释性 Explanation 等等。

## 五、典型商业推荐系统浅析

### 5.1 典型的实用推荐系统

电子商务领域：

Amazon、eBay、淘宝、dangdang、京东商城等  
百分点科技为别人做推荐系统

网页标签：

Fab、[del.icio.us](http://del.icio.us)、Foxtrot 等

新闻与阅读：

GroupLens、PHOAKS、Zite、Flipboard、Trap.it 等  
国内：指阅、牛赞网、无觅网（专门为各大博客和论坛提供相似文章推荐的网站）等

电影：

MovieLens、Moviefinder、Netflix、Hulu、豆瓣、猜道等；  
国内的一些影视类网站大都有自己的推荐系统，比如奇艺，优酷，土豆等；

音乐：

Pandora、Ringo、CDNOW、[last.fm](http://last.fm)、豆瓣电台、新浪音乐等；

### 5.2 Amazon 商用推荐系统调研

在 amazon 的商城系统中多处应用了推荐算法，例如：

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物品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物品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。

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: 因为你的购物车中有 XX, 或者因为你购买过 XX, 所以给你推荐类似的 XX; 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; 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; 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。

据 VentureBeat 统计, Amazon 的推荐系统为其提供了 35% 的销售额。另外, 亚马逊早期推荐系统的主要贡献者 Greg Linden 在博文《YouTube uses Amazon's recommendation algorithm》中讨论了 YouTube 在 RecSys2010 上的一篇文章, 该文报告 YouTube 的推荐算法主要采用 Item-based 方法和 Ranking 方法。

## 六、推荐系统目前面临的问题

### 6.1 数据稀疏性 Data Sparsity

表现在: 冷启动问题, 新物品问题、新用户问题等; Reduced Coverage 问题, rating 相对于 item 来说太少了; Neighbour Transitivity Problem, 因为数据量过小, 没有用户对相同的 item 进行过评价, 因此没法计算他们之间的相似性 (例如在电影推荐系统中, 有很多电影只被小部分用户评级, 而且这些电影会很少被推荐, 即使那小部分用户给予很高评级。同样, 对于那些有着不同品味的小众群体, 找不到相同特定口味的用户, 也导致较差的推荐结果了)。

解决方法: 降维技术 (Dimensionality Reduction), 通过奇异值分解 (SVD, Singular Value Decomposition) 来降低稀疏矩阵的维度, 为原始矩阵求的最好的低维近似, 但是存在大数据量运算成本及对效果的影响等问题 (因为经过了 SVD 变化之后, 一些 insignificant 用户或者物品被扔掉了, 对这类用户或者物品的推荐效果就要打折扣, 小众群体的存在体现不出来); 使用 Hybrid

方法，混合使用多种推荐方法能弥补其中某种方法的问题，弥补其造成的冷启动问题；Demographic filtering，解决数据稀疏性问题的另一种方法是通过使用用户资料信息来计算用户相似度，也就是两个用户会被认为相似不只单在相同的电影评级类似，而且也有可能属于同一个人口统计区块（demographic），比如用户的性别、年龄、居住地、教育情况、工作信息等，这样的做法会降低推荐精度，但是在数据非常稀疏的情况下也可以考虑使用。

## 6.2 同义词问题 Synonymy

问题表现为推荐系统中同一类物品有时被归为不同的名字（这也进一步导致了数据稀疏性），解决方法可以有同义词挖掘、利用 SVD 进行语义分析等。

## 6.3 Gray Sheep 问题

问题表现为有些人的偏好与任何人都不同（Black Sheep 是那些偏好与正常人完全相反，根本没有办法向他们推荐的人群，因为在现实中也很难解决这个问题，因此这一般被认为是 acceptable failure）。一般采用 Hybrid（结合 Content-based 和 CF）方法解决该问题。

## 6.4 Shilling Attack

实际上是 AntiSpam 的问题，有些人对于自己的东西或者对自己有利的东西打高分，竞争对手的东西打低分，这会影响协同过滤算法的正常工作。被动的解决办法可以是采用 Item-Based（在 shilling attack 这个问题上，Item-based 的效果要比 User-based 的效果好，因为作弊者总是较少数，在计算 Item 相似度的时候影响较小），Hybrid（hybrid 方法能够在部分的解决 bias injection problem）；主动的解决办法是采用 AntiSpam 的技术识别和去除作弊者的影响。

## 6.5 其它一些问题和挑战

隐私问题、噪声问题、解释性问题、合理性、新颖性、对推荐系统的第二个核心问题 recommendation 的研究等等。

# 七、推荐系统潜在的发展方向

## 7.1 基于 User Review 的推荐

长期以来，推荐系统的各种算法和研究都是基于[2]所提出的形式化模型，均以用户打分为基础，少有基于用户评论语料的研究。基于 User Review 的个性化推荐被很多论文提到，但是研究并不深入，这一方面限于 Text Mining 和 Sentiment Analysis 方面的研究遇到很多难点，另一方面限于之前网络上所积累的 Review 信息还不够多。

2006 年 WI 的一篇论文[22]首次尝试利用用户评论文本语料来进行推荐，但是其系统更像是一个检索系统而非推荐系统(从评论文本中手工挖掘 Item 的属性，构建 Item Profile，当用户查询他所关心的 Feature 时，推荐在这些 Feature 上表现比较好的 Item)。2007 年，同样是该论文的作者在推荐系统的核心期刊 Recommender Systems 上发文，将手工挖掘 Item 属性的步骤实现了自动完成，其它环节未作改变。User Comment 的文本挖掘及情感分析技术在个性化推荐系统中的

应用尚有很多工作要做。

## 7.2 可解释性

推荐系统作为人工智能领域的一个重要产品，要想受到广泛接受和应用，最核心的是要提高推荐结果的合理性，也就要求推荐结果具有较好的可解释性，虽然这一点人们早就意识到，但是针对可解释性的专门研究还很欠缺[9]，当前的研究中，对推荐算法可解释性讨论一般是算法评估之后的选作环节。随着用户要求的提高，“推荐理由”的研究在产业界和学术界都受到越来越高的重视。

## 7.3 用户交互

一些研究表明，即便是同样的打分和评价系统，如果展示给用户的方式不同，也会对用户的使用、评价、效果产生一定的影响。比如 MovieLens 小组的论文[27]第一次研究了用户打分区间、连续打分还是离散(如星标)打分、推荐系统主动欺骗等对用户使用推荐系统造成的影响。向搜索引擎一样，推荐系统的界面设计和交互方式也越来越受到研究人员的关注。

## 7.4 长尾效应

一个推荐系统的好坏不能直接以预测评分的精确度来测量，而应该以用户的满意度来考虑。推荐系统应该以“发现 Discovery”为核心终极目标。而现在存在的一些推荐技术通常会倾向于推荐流行度很高的，用户已经知道的 Item。这样，存在长尾中的 Item 也就不能很好的推荐给相应的用户。但是，这些长尾中的 Item 通常也标注了用户的兴趣偏好。所以，在推荐系统的设计过程中，不仅要考虑预测的精度，而且还要考虑用户真正的兴趣点在哪里。

最近，有很多人也开始考虑长尾在推荐系统设计过程中的应用，开始考虑怎样将长尾中的 Item 推荐给用户。根据[24]，可以将 Item 聚类成几个 small cluster，然后对每个 cluster 中的 item 运用协同过滤。利用这一思想，在文章[25]中，作者将 Item 分成 head 和 tail 两部分，而针对 tail 中的 Item，作者提出了 Clustered Tail 算法来做推荐。另外[26]利用 Diffusion 理论来解决推荐问题。

## 参考文献

- [1] G. Adomavicius, “Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, IEEE Transactions on Knowledge and Data Engineering, 2005.
- [2] P. Resnick, N. Iacovou, etc. “GroupLens: An Open Architecture for Collaborative Filtering of Netnews”, Proceedings of ACM Conference on Computer Supported Cooperative Work, CSCW 1994. pp.175-186.
- [3] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, “Item-Based Collaborative Filtering Recommendation Algorithms”, Proceeding of the 10<sup>th</sup> international conference on World Wide Web, WWW 2001.
- [4] K. Sugiyama, K. Hatano, M. Yoshikawa, Adaptive Web Search based on User Profile Constructed without Any Effort from Users. Proceeding of the 13<sup>th</sup> international conference on World Wide Web, WWW 2004.
- [5] R. Wietsma, F. Ricci, “Product Review in Mobile Decision Aid Systems”, Proceeding of Workshop



Pervasive Mobile Interaction Devices. Pervasive 2005.

- [6] F. Ricci, R. Wietsma, "Product Reviews in Travel Decision Making", Information and Communication Technologies in Tourism: Proc. Int'I Conf.
- [7] S. Aciar, D. Zhang, S. Simoff, J. Debehm. Informed Recommender: Basing Recommendations on Consumer Product Reviews. Recommender Systems. 2007.
- [8] C Ziegler, S. M. McNee, etc. Improving Recommendation List Through Topic Diversification. Proceeding of the 15<sup>th</sup> international conference on World Wide Web. WWW 2005.
- [9] N. Tintarev, J. Masthoff, A Survey of Explanations in Recommender Systems, Proceedings of the 2000 ACM conference on Computer supported cooperative work, CSCW 2000.
- [10] M. J. Pazzani, "A framework for Collaborative, Content-Based and Demographic Filtering", Artificial Intelligence Review, Springer, 1999
- [11] A. Gunawardana, G. Shani, A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. Journal of Machine Learning, Volume 10, pp.2935-2962
- [12] J. L. Herlocker, J. A. Konstan, Al Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, Proceedings of the 22<sup>nd</sup> international SIGIR conference on Research and development in information retrieval, SIGIR 1999.
- [13] H. Kautz, B. Selman, M. Shah, Referral Web: combining social networks and collaborative filtering, Communications of the ACM, Volume 40 Issue 2, March 1997
- [14] B. M. Sarwar, G. Karypis, etc, Application of Dimensionality Reduction in Recommender System – A Case Study, DTIC Document, 2000
- [15] T. George, S. Merugu, A Scalable Collaborative Filtering Framework based on Co-clustering, Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM 2005
- [16] P. Melville, R. J. Mooney, R. Nagarajan, Content-Boosted Collaborative Filtering for Improved Recommendations, Proceedings of American Association for Artificial Intelligence, AAAI 2002
- [17] B. M. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of Recommendation Algorithms for E-Commerce, Proceedings of the 2<sup>nd</sup> ACM conference on Electronic commerce, ACM EC 2000
- [18] M. O’Cornor, Jon Herlocker, Clustering Items for Collaborative Filtering, Proceedings of the ACM SIGIR Workshop, SIGIR 1999
- [19] K. Zhou, S. Yang, H. Zha, Functional Matrix Factorizations for Cold-Start Recommendation, Proceedings of the 34<sup>th</sup> Annual International ACM SIGIR Conference, SIGIR 2011
- [20] B. Burke, Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction, 12: 331-370, 2002
- [21] 刘建国, 周涛等, 个性化推荐系统评价方法综述, 复杂系统与复杂性科学第 6 卷第 3 期, 2009
- [22] S. Aciar, D. Zhang, S. Simoff, J. Debenham, Recommender System Based on Consumer Product Reviews, Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006
- [23] S. Aciar, D. Zhang, S. Simoff, J. Debenham, Informed Recommender: Basing Recommendations on Consumer Product Reviews, Recommender Systems, June 2007
- [24] Truong, K.Q., Ishikawa, F., Honiden, S. Improving Accuracy of Recommender System by Item Clustering, IEICE TRANSACTIONS on Information and Systems, E90-D-I, 2007
- [25] Y. Park, A. Tuzhilin, The long tail of recommender systems and how to leverage it, RecSys, 2008.
- [26] M. Ishikawa, P. Geczy, N. Izumi, T. Yamaguchi, Long Tail Recommender Utilizing Information Diffusion Theory, In WI-IAT, 2008

[27] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, J. Riedl, Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions, Proceedings of the 2003 International Conference on Human Factors In Computing Systems, CHI 2003