大数据与推荐系统

李翠平，蓝梦微，邹本友，王绍卿，赵衍衍
中国人民大学数据工程与知识工程教育部重点实验室 北京 100872

摘要
随着大数据时代的来临，网络中的信息量呈现指数式增长，随之带来了信息过载问题。推荐系统是解决信息过载最有效的方式之一，大数据推荐系统已经逐渐成为信息领域的研究热点。介绍了推荐系统的产生及其在大数据时代的发展现状、推荐系统的领域需求和系统架构、大数据环境下推荐系统的挑战及其关键技术、开源的大数据推荐软件、大数据推荐系统研究面临的问题，最后探讨了大数据推荐系统的未来发展趋势。

关键词
大数据；推荐系统；协同过滤

doi: 10.11959/j.issn.2096-0271.2015026

Big Data and Recommendation System

Li Cuiping, Lan Mengwei, Zou Benyou, Wang Shaoqing, Zhao Kankan
Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, Renmin University of China, Beijing 100872, China

Abstract
In big data era, recommendation system is the key means to tackle the issue of "information overload". Recommendation system has been widely applied to many domains. The most typical and promising domain is the e-commence. Recently, with the rapid development of e-commence, recommendation system becomes more and more important and is promoted as a hot research field. The history and development of recommendation system, its domain requirements and system architecture, its characteristics and challenges under big data environment, its key techniques, open source big data recommendation systems were introduced. And at last, the open research problems and future trends of bid data recommendation system were discussed.

Key words
big data, recommendation system, collaborative filtering

1 推荐系统与网络大数据

随着科技与信息技术的迅猛发展，社会进入了一个全新的高度信息化的时代，互联网无处不在，影响了人类生活的方方面面，并彻底改变了人们的生活方式。尤其是进入Web 2.0时代以来，随着社会化网络媒体的异军突起，互联网用户既是网络信息的消费者，也是网络内容的生产者，互联网中的信息量呈指数级增长。由于用户的辨别能力有限，在面对庞大且复杂的互联网信息时往往感到无从下手，使得在互联网中找寻有用信息的成本巨大，产生了所谓的“信息过载”问题。

搜索引擎和推荐系统的产生为解决“信息过载”问题提供了非常重要的技术手段。对于搜索引擎来说，用户在搜索互联网中的信息时，需要在搜索引擎中输入“查询关键词”，搜索引擎根据用户的输入，在系统后台进行信息匹配，将与用户查询相关的信息展示给用户。但是，如果用户无法想到准确描述自己需求的关键词，此时搜索引擎就无能为力了。和搜索引擎不同，推荐系统不需要用户提供明确的需求，而是通过分析用户的历史行为来对用户的兴趣进行建模，从而主动给用户推荐可能满足他们兴趣和需求的信息。因此，搜索引擎和推荐系统对用户来说是两个互补的工具，前者是主动的，而后者是被动的。

近几年，电子商务蓬勃发展，推荐系统在互联网中的优势地位也越来越明显。在国际方面，比较著名的电子商务网站有Amazon和eBay，其中Amazon平台中采用的推荐算法被认为是非常成功的。在国内，比较大型的电子商务平台网站有淘宝网（包括天猫商城）、京东商城、当当网、苏宁易购等。在这些电子商务平台中，网站提供的商品数量不计其数，网站中的用户规模也非常巨大。据不完全统计，天猫商城中的商品数量已经超过了4 000万。在如此庞大的电商网站中，用户根据自己的购买意图输入关键字查询后，会得到很多相似的结果，用户在这些结果中也很难区分异同，用户也难于选择合适的物品。于是，推荐系统作为能够根据用户兴趣为用户推荐一些用户感兴趣的商品，从而为用户在购物的选择中提供建议的需求非常明显。目前比较成功的电子商务网站中，都不同程度地利用推荐系统在用户购物的同时，为用户推荐一些商品，从而提高网站的销售额。

另一方面，智能手机的发展推动了移动互联网的发展。在用户使用移动互联网的过程中，其所处的地理位置等信息可以非常准确地被获取。基于此，国内外出现了大量的基于用户位置信息的网站。国外比较著名的有Meetup和Flickr。国内著名的有豆瓣网和大众点评网。例如，在大众点评这种基于位置服务的网站中，用户可以根据自己的当前位置搜索餐馆、酒店、影院、旅游景点等信息服务。同时，可以对当前位置下的各类信息进行点评，为自己在现实世界中的体验打分，分享自己的经验与感受。当用户使用这类基于位置的网站服务时，同样会遭遇“信息过载”问题。推荐系统可以根据用户的位置信息为用户推荐当前位置下用户感兴趣的内容，为用户提供符合其真正需要的内容，提升用户对网站的满意度。

随着社交网络的兴起，用户在互联网中的行为不再限于获取信息，更多的是与网络上的其他用户进行互动。国外著名的社交网络有Facebook、LinkedIn、Twitter等，国内的社交网络有新浪微博、人人网、腾讯微博等。在社交网站中，用户不再是

单个的个体，而是与网络中的很多人具有了错综复杂的关系。社交网络中最重要的资源就是用户与用户之间的这种关系数据。在社交网络中，用户间的关系是不同的，建立关系的因素可能是现实世界中的亲人、同学、同事、朋友关系，也可能是网络中的虚拟朋友，比如都是有着共同爱好的社交网络成员。在社交网络中，用户与用户之间的联系反映了用户之间的信任关系，用户不单单是一个个体，用户在社交网络中的行为或多或少地会受到这些用户关系的影响。因此，推荐系统在这类社交网站中的研究与应用，应该考虑用户社交关系的影响。

2 推荐系统的产生与发展

“推荐系统”这个概念是1995年在美国人工智能协会（AAAI）上提出的。当时CMU大学的教授Robert Armstrong提出了这个概念，并推出了推荐系统的原型系统——Web Watcher。在同一个会议上，美国斯坦福大学的Marko Balabanovic等人推出了个性化推荐系统LIRA1。随后推荐系统的研究工作开始慢慢壮大。1996年，Yahoo网站推出了个性化入口My Yahoo，可以看作第一个正式商用的推荐系统。21世纪以来，推荐系统的研究与应用随着电子商务的快速发展而异军突起，各大电子商务网站都部署了推荐系统，其中Amazon网站的推荐系统比较著名。有报告称，Amazon网站中35%的营业额来自于自身的推荐系统。2006年，美国的DVD租赁公司Netflix在网上公开设立了一个推荐算法竞赛——Netflix Prize。Netflix公开了真实网站中的一部分数据，包含用户对电影的评分[2]。Netflix竞赛有效地推动了学术界和产业界对推荐算法的研究，期间提出了很多有效的算法。近几年，随着社会化网络的发展，推荐系统在工业界广泛应用并且取得了显著进步。比较著名的推荐系统应用有：Amazon和淘宝网的电子商务推荐系统、Netflix和MovieLens的电影推荐系统、Youtube的视频推荐系统、豆瓣和Last.fm的音乐推荐系统、Google的新闻推荐系统以及Facebook和Twitter的好友推荐系统。

推荐系统诞生后，学术界对其关注也越来越多。从1999年开始，美国计算机学会每年召开电子商务研讨会（ACM Conference on Electronic Commerce，ACM EC），越来越多的与推荐系统相关的论文发表在ACM EC上。ACM信息检索专业组（ACM Special Interest Group of Information Retrieval，ACM SIGIR）在2001年开始把推荐系统作为该会议的一个独立研究主题。同年召开的人工智能联合大会（The 17th International Joint Conference on Artificial Intelligence）也将推荐系统作为一个单独的主题。最近的10年间，学术界对推荐系统越来越重视。目前为止，数据库、数据挖掘、人工智能、机器学习方面的重要国际会议（如SIGMOD、VLDB、ICDE、KDD、AAAI、SIGIR、ICDM、WWW、ICML等）都有大量与推荐系统相关的研究成果发表。同时，第一个以推荐系统命名的国际会议ACM Recommender Systems Conference（ACM RecSys）于2007年首次举办。在近几年的数据挖掘及知识发现国际会议（KDD）举办的KDD CUP竞赛中，连续两年的竞赛主题都是推荐系统。在KDD CUP 2011年的竞赛中，两个竞赛题目分别为“音乐评分预测”和“识别音乐是否被用户评分”。在KDD CUP 2012年的竞赛中，两个竞赛题目分别为“腾讯微博中的好友推荐”和“计算广告中的点击率预测”。

3 推荐系统的领域需求和系统架构

如上所述，推荐系统在很多领域得到了广泛的应用，如新闻推荐、微博推荐、图书推荐、电影推荐、产品推荐、音乐推荐、餐馆推荐、视频推荐等。不同领域的推荐系统具有不同的数据稀疏性，对推荐系统的可扩展性以及推荐结果的相关性、流行性、新鲜性、多样性和新颖性具有不同的需求。不同领域推荐系统的需求对比见表1。

尽管需求不尽相同，一个完整的推荐系统通常都包括数据建模、用户建模、推荐引擎和用户接口4个部分，如图1所示。

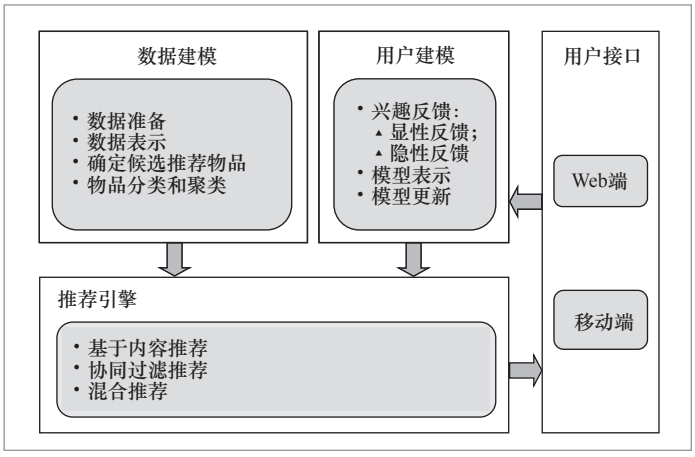


图1 系统架构

数据建模模块负责对拟推荐的物品数据进行准备，将其表示成有利于分析的数据形式，确定要推荐给用户的候选物品，并对物品进行分类、聚类等预处理。用户建模模块负责对用户的行为信息进行分析，从而获得用户的潜在喜好。用户的行为信息包括问答、评分、购买、下载、浏览、收藏、停留时间等。推荐引擎模块利用后台的推荐算法，实时地从候选物品集合中筛选出用户感兴趣的物品，排序后以列表的形式向用户推荐。推荐引擎是推荐系统的核心部分，也是最耗系统资源和时间的部分。用户接口模块承担展示推荐结果、收集用户反馈等功能。用户接口除了应具有布局合理、界面美观、使用方便等基本要求外，还应有助于用户主动提供反馈。主要有两种类型的接口：Web端（Web-based）和移动端（mobile-based）。受篇幅限制，仅对用户建模和推荐引擎这两个重要模块进行详细介绍。

3.1 用户建模

用户模型反映用户的兴趣偏好。用户兴趣的反馈可分为显性反馈和隐性反馈。显性反馈包含两种方式：用户定制和用户评分。用户定制是指用户对系统所列问题的回答，如年龄、性别、职业等。评分又分

表1 不同领域的推荐系统需求对比

应用领域	数据稀疏性	可扩展性	相关性	流行性	新鲜性	多样性	新颖性
新闻推荐	****	****	****	****	***	****	***
微薄推荐	****	****	****	***	****	***	***
图书推荐	***	*	****	**	*	*	*
电影推荐	***	*	***	***	***	*	***
产品推荐	***	**	**	***	**	***	*
音乐推荐	***	*	***	***	***	***	***
餐馆推荐	**	*	*	***	***	***	**
视频推荐	*	*	***	****	**	*	*

注：****强，***较强，**较弱，*弱

为两级评分和多级评分。例如，在Yahoo News中采用两级评分：喜欢（more like this）和不喜欢（less like this）。多级评分可以更详细地描述对某个产品的喜欢程度，如GroupLens中用户对新闻的喜好程度可评价为1~5分。News Dude支持用户的4级反馈：感兴趣、不感兴趣、已知道、想了解更多，然后进行归一化处理。

很多时候用户不能够准确地提供个人偏好或者不愿意显性提供个人偏好，更不愿意经常维护个人的偏好。所以，隐性反馈往往能够正确地体现用户的偏好以及偏好的变化。常用的隐性反馈信息有：是否点击、停留时间、点击时间、点击地点、是否加入收藏、评论内容（可推测用户的心情）、用户的搜索内容、社交网络、流行趋势、点击顺序等。在协同过滤推荐方法中，常常把用户的隐性反馈转化为用户对产品的评分。例如，Google News中用户阅读过的新闻记为喜欢，评分为1；没有阅读过的评分为0。Daily Learner系统中用户点击了新闻标题评分为0.8分，阅读完全文则评分上升到1分；若用户跳过了系统推荐的新闻，则从系统预测评分中减去0.2分作为最终评分。

用户的兴趣可分为长期兴趣和短期兴趣。长期兴趣反映用户的真实兴趣；短期兴趣常与热点话题相关联且经常改变，从最近的历史行为中学习到的短期兴趣模型可快速反映用户兴趣的变化。常用的模型有向量空间模型、语义网络模型、基于分类器的模型等。由于用户的兴趣常受物品本身周期性、热点事件、突发事件的影响，变化性很大。所以，需要经常更新用户模型。

3.2 推荐引擎

推荐引擎的基本推荐方法可分为基于内容的推荐和基于协同过滤的推荐。

基于内容的推荐方法的基本原理是，根据用户以往喜欢的物品，选择其他类似的物品作为推荐结果[2]。例如，现在有一部新电影与用户过去看过的某部电影有相同演员或者题材类似，则用户可能就喜欢这部新电影。通常使用用户模型的向量特征来描述用户的兴趣爱好，同样对于每个物品进行特征提取，作为物品模型的内容特征。然后计算用户模型的向量特征和候选物品模型的向量特征两者之间的匹配度，匹配度较高的候选物品就可作为推荐结果推送给目标用户。

协同过滤技术是由David Goldberg在1992年提出的，是目前个性化推荐系统中应用最为成功和广泛的技术。国外著名的商业网站Amazon，国内比较著名的豆瓣网、虾米网等网站，都采用了协同过滤的方法。其本质是基于关联分析的技术，即利用用户所在群体的共同喜好来向用户进行推荐。协同过滤利用了用户的历史行为（偏好、习惯等）将用户聚类成簇，这种推荐通过计算相似用户，假设被其他相似用户喜好的物品当前用户也感兴趣。协同过滤的推荐方法通常包括两个步骤：根据用户行为数据找到和目标用户兴趣相似的用户集合（用户所在的群体或簇）；找到这个集合中用户喜欢的且目标用户没有购买过的物品推荐给目标用户。

在实际使用中，协同过滤技术面临两大制约：一是数据稀疏问题，二是冷启动问题。协同过滤需要利用用户和用户或者物品与物品之间的关联性进行推荐。最流行的基于内存的协同过滤方法是基于邻居关系的方法。该方法首先找出与指定用户评价历史相近的该用户的邻居，根据这些邻居的行为来预测结果或者找出与查询物品类似的物品。这样做的前提假设是，如果两个用户在一组物品上有相似的评价，那么他们对其他的物品也将会有相似的评价；或者如果两物品在一组用户上有相似

的评价，那么他们对于其他的用户也将会有相似的评价。

协同过滤算法的关键是找寻用户（物品）的最近邻居。当数据稀疏时，用户购买过的物品很难重叠，协同推荐的效果就不好。改进办法之一是，除了直接邻居之外，间接邻居的行为也可以对当前用户的决策行为构成影响。另外一些解决稀疏问题的方法是可以添加一些缺省值，人为地将数据变得稠密一些，或者采用迭代补全的方法，先补充部分数值，在此基础上再进一步补充其他数值。此外，还有利用迁移学习的方法来弥补数据稀疏的问题。但这些方法只能在某种程度上部分解决数据稀疏的问题，并不能完全克服。在真实应用中，由于数据规模很大，数据稀疏的问题更加突出。数据稀疏性使协同过滤方法的有效性受到制约。甄别出与数据稀疏程度相匹配的算法，以便能根据具体应用情况做出正确选择，是非常有价值的研究课题。

常用的协同过滤方法有两类：基于内存的方法和基于模型的方法。前者主要是内存算法，通过用户与物品之间的关系来导出结果；后者需要找到一个合适的参数化的模型，然后通过这个模型来导出结果。

基于用户的协同过滤[4]鉴别出与查询用户相似的用户，然后将这些用户对物品评分的均值作为该用户评分结果的估计值。与此类似，基于物品的协同过滤鉴别出与查询物品类似的物品，然后将这些物品的评分均值作为该物品预测结果的估计值。基于邻居的方法随着计算加权平均值方法的不同而不同。常用的计算加权平均值的算法有皮尔逊系数、矢量余弦、MSD。

基于模型的方法通过适合训练集的参数化模型来预测结果。它包括基于聚类的CF[5~7]、贝叶斯分类器[8,9]、基于回归的方法[10]。基于聚类方法的基本思想是将相似的用户（或物品）组成聚类，这种技术有助于解决数据稀疏性和计算复杂性问题。贝叶斯的基本思想是给定用户A其他的评分和其他用户评分情况下，计算每个可能评分值（比如电影推荐中的1~5分）的条件概率，然后选择一个最大概率值的评分作为预测值。基于回归方法的基本思想是先利用线性回归模型学习物品之间评分的关系，然后根据这些关系预测用户对物品的评分。Slop-one算法[13]在评价矩阵上使用了线性模型，使之能够快速计算出具有相对较好精确度的结果。

最近一类成功的基于模型的方法是基于低秩矩阵分解的方法。例如，SVD[11]和SVD++[12]将评价矩阵分解为3个低秩的矩阵，这3个矩阵的乘积能对原始矩阵进行某种程度的复原，从而可以评估出缺失值。另一种方法是非负矩阵分解[13]，其不同之处在于，矩阵分解的结果不得出现负值。基于低秩矩阵分解的方法从评分矩阵中抽取一组潜在的（隐藏的）因子，并通过这些因子向量描述用户和物品。在电影领域，这些自动识别的因子可能对应一部电影的常见标签，比如风格或者类型（戏剧片或者动作片），也可能是无法解释的。

矩阵分解能够对两类变量进行交互关系的预测。Tensor分解模型则能够将这种不同类变量的交互预测扩展到更高的维度。然而，如果将因子分解模型应用到一个新的任务，针对新问题往往需要在原有因子分解基础上推导演化，实现新的模型和学习算法。例如SVD++、STE、FPMC、timeSVD++、BPTF等模型，都是针对特定问题在原有因子分解模型基础上做的改进。因此，普通的因子分解模型具有较差的泛化能力。在模型优化学习算法方面，虽然对基本矩阵分解模型的学习已经有很多算法，如（随机）梯度下降、交替最小二乘法、变分贝叶斯和MCMC（Markov chain Monto Carlo），但是对于更多的复杂分解

模型而言，最多且最常用的方法是梯度下降算法。

因子分解机（factorization machine）是Steffen Rendle于2010年提出的一个通用的模型[3]。凭借该模型，Rendle在KDD Cup 2012中分别取得Track1第2名和Track2第3名的成绩。与原有的因子分解模型相比，该模型将特征工程的一般性与分解模型的优越性融合。它能够通过特征工程来模拟绝大多数的因子分解模型。LibFM是因子分解机的开源实现，简单易用，不需要太多专业知识，其中包括3类优化学习算法：随机梯度下降、交替最小二乘法和MCMC。

这里提到的Tensor分解模型和因子分解机都属于上下文感知推荐算法的范畴。上下文感知的推荐算法将二维协同扩展到多维协同。从学科渊源来看，上下文感知推荐系统既是一种推荐系统，也是一种上下文感知应用系统。Adomavicius和Tuzhilin等人较早指出，把上下文信息融入推荐系统将有利于提高推荐精确度，并提出被广泛引用的“上下文感知推荐系统（context-aware recommender systems，CARS）”的概念。他们将传统的“用户—项目”二维评分效用模型扩展为包含多种上下文信息的多维评分效用模型。Sun等人首先将HOSVD的方法用于网页搜索，提出了CubeSVD算法[14]，算法将用户的位置信息作为上下文信息，用于搜索引擎的结果排序，取得了比较好的结果。Renle等人提出RTF算法[15]，与HOSVD不同，RTF算法根据用户的排序进行优化，可以获得比较好的准确度。

基于内容的推荐方法和基于协同过滤的推荐方法各有其优缺点。现有的系统大部分是一种混合系统，它结合不同算法和模型的优点，克服它们的缺点，从而得到了较好的推荐准确度。

4 大数据环境下的推荐系统

4.1 特点与挑战

虽然推荐系统已经被成功运用于很多大型系统及网站，但是在当前大数据的时代背景下，推荐系统的应用场景越来越多样，推荐系统不仅面临数据稀疏、冷启动、兴趣偏见等传统难题，还面临由大数据引发的更多、更复杂的实际问题。例如，用户数目越来越多，海量用户同时访问推荐系统所造成的性能压力，使传统的基于单节点 LVS 架构的推荐系统不再适用。同时 Web 服务器处理系统请求在大数据集下变得越来越多，Web服务器响应速度缓慢制约了当前推荐系统为大数据集提供推荐。另外，基于实时模式的推荐在大数据集下面临着严峻考验，用户难以忍受超过秒级的推荐结果返回时间。传统推荐系统的单一数据库存储技术在大数据集下变得不再适用，急需一种对外提供统一接口、对内采用多种混合模式存储的存储架构来满足大数据集下各种数据文件的存储。并且，传统推荐系统在推荐算法上采取的是单机节点的计算方式，不能满足大数据集下海量用户产生的大数据集上的计算需求[16]。大数据本身具有的复杂性、不确定性和涌现性也给推荐系统带来诸多新的挑战，传统推荐系统的时间效率、空间效率和推荐准确度都遇到严重的瓶颈。

4.2 关键技术

4.2.1 采用分布式文件系统管理数据

传统的推荐系统技术主要处理小文件存储和少量数据计算，大多是面向服务器

的架构，中心服务器需要收集用户的浏览记录、购买记录、评分记录等大量的交互信息来为单个用户定制个性化推荐。当数据规模过大，数据无法全部载入服务器内存时，就算采用外存置换算法和多线程技术，依然会出现I/O上的性能瓶颈，致使任务执行效率过低，产生推荐结果的时间过长。对于面向海量用户和海量数据的推荐系统，基于集中式的中心服务器的推荐系统在时间和空间复杂性上无法满足大数据背景下推荐系统快速变化的需求[16]。

大数据推荐系统采用基于集群技术的分布式文件系统管理数据。建立一种高并发、可扩展、能处理海量数据的大数据推荐系统架构是非常关键的，它能为大数据集的处理提供强有力的支持。 Hadoop 的分布式文件系统（Hadoop distributed file system，HDFS）架构是其中的典型。与传统的文件系统不同，数据文件并非存储在本地单一节点上，而是通过网络存储在多台节点上。并且文件的位置索引管理一般都由一台或几台中心节点负责[16]。客户端从集群中读写数据时，首先通过中心节点获取文件的位置，然后与集群中的节点通信，客户端通过网络从节点读取数据到本地或把数据从本地写入节点。在这个过程中由 HDFS 来管理数据冗余存储、大文件的切分、中间网络通信、数据出错恢复等，客户端根据 HDFS 提供的接口进行调用即可，非常方便。

4.2.2 采用基于集群技术的分布式计算框架

集群上实现分布式计算的框架很多，Hadoop中的MapReduce 作为推荐算法并行化的依托平台，既是一种分布式的计算框架，也是一种新型的分布式并行计算编程模型，应用于大规模数据的并行处理，是一种常见的开源计算框架。MapReduce算法的核心思想是“分而治之”，把对大规模数据集的操作，分发给一个主节点管理下的各个分节点共同完成，然后通过整合各个节点的中间结果，得到最终结果。MapReduce 框架负责处理并行编程中分布式存储、工作调度、负载均衡、容错均衡、容错处理以及网络通信等复杂问题，把处理过程高度抽象为两个函数：map和reduce。map负责把任务分解成多个任务，reduce负责把分解后多任务处理的结果汇总起来[16]。例如，2010 年，Zhao等人针对协同过滤算法的计算复杂性在大规模推荐系统下的局限性，在 Hadoop平台上实现了基于物品的协同过滤算法。2011年，针对推荐系统无法在每秒内给大量用户进行推荐的问题，Jiang等人将基于物品的协同过滤推荐算法的3个主要计算阶段切分成4个MapReduce阶段，切分后各阶段可以并行运行在集群的各个节点上。同时他们还提出了一种 Hadoop平台下的数据分区策略，减少了节点间的通信开销，提高了推荐系统的推荐效率。

4.2.3 推荐算法并行化

很多大型企业所需的推荐算法要处理的数据量非常庞大，从TB级别到PB级甚至更高，例如腾讯Peacock主题模型分析系统需要进行高达十亿文档、百万词汇、百万主题的主题模型训练，仅一个百万词汇乘以百万主题的矩阵，其数据存储量已达3 TB，如果再考虑十亿文档乘以百万主题的矩阵，其数据量则高达3 PB[17]。面对如此庞大的数据，若采用传统串行推荐算法，时间开销太大。当数据量较小时，时间复杂度高的串行算法能有效运作，但数据量极速增加后，这些串行推荐算法的计算性能过低，无法应用于实际的推荐系统中。因此，面向大数据集的推荐系统从设计上就应考虑到算法的分布式并行化技术，使得推荐算法能够在海量的、分布式、

异构数据环境下得以高效实现。

5 开源大数据典型推荐软件

5.1 Mahout

Mahout[1]是Apache Software Foundation（ASF）旗下的一个全新的开源项目，其主要目标是提供一些可伸缩的机器学习领域经典算法的实现，供开发人员在 Apache 许可下免费使用，旨在帮助开发人员更加方便、快捷地开发大规模数据上的应用程序。除了常见的分类、聚类等数据挖掘算法外，还包括协同过滤（CF）、维缩减（dimensionality reduction）、主题模型（topic models）等。Mahout集成了基于Java的推荐系统引擎“Taste”，用于生成个性化推荐“Taste”支持基于用户的、基于物品的以及基于slope-one的推荐系统。在Mahout的推荐类算法中，主要有基于用户的协同过滤（user-based CF）、基于物品的协同过滤（item-based CF）、交替最小二乘法（ALS）、具有隐含反馈的ALS（ALS on implicit feedback）、加权矩阵分解（weighted MF）、SVD++、并行的随机梯度下降（parallel SGD）等。

5.2 Spark MLlib

Spark MLlib[2]对常用的机器学习算法进行了实现，包括逻辑回归、支持向量机、朴素贝叶斯等分类预测算法，*K*-means聚类算法，各种梯度下降优化算法以及协同过滤推荐算法。MLlib当前支持的是基于矩阵分解的协同过滤方法，其函数优化过程可采用其提供的交替最小二乘法或者梯度下降法来实现，同时支持显性反馈和隐性反馈信息。

5.3 EasyRec

EasyRec[3]是SourceForge的一个开源项目。它针对个人用户，提供低门槛的易集成、易扩展、好管理的推荐系统。该开源产品包括了数据录入、数据管理、推荐挖掘、离线分析等功能。它可以同时给多个不同的网站提供推荐服务。需要推荐服务的网站用户只需配合着发送一些用户行为数据到EasyRec，EasyRec则会进行后台的推荐分析，并将推荐结果以XML或JSON的格式发送回网站。用户行为数据包括用户看了哪些商品、买了哪些商品、对哪些商品进行了评分等。EasyRec为网站用户提供了访问EasyRec全部功能的接口，可通过调用这些接口来实现推荐业务。

5.4 Graphlab

Graphlab[4]始于2009年，是由美国卡内基梅隆大学开发的一个项目。它基于C++语言，主要功能是提供一个基于图的高性能分布式计算框架。GraphLab能够高效地执行与机器学习相关的数据依赖性强的迭代型算法，为Boosted决策树、深度学习、文本分析等提供了可扩展的机器学习算法模块，能对分类和推荐模型中的参数进行自动调优，和SPARK、Hadoop、Apache Avro、OBDC connectors等进行了集成。由于功能独特，GraphLab在业界很有名气。针对大规模的数据集，采用GraphLab来进行随机游走（random walk）或基于图的推荐算法非常有效。另外，GraphLab还实现了交替最小二乘法ALS、随机梯度下降法 SGD、SVD++、Weighted-ALS、Sparse-ALS、非负矩阵分解（non-negative matrix factorization）等算法。

1 http://mahout.apache.org/

2 http://spark.apache.org/docs/0.9.0/api/mllib/index.html#org.apache.spark.mllib.recommendation.package

3 http://easyrec.org/

4 https://github.com/dato-code/PowerGraph

5.5 Duine

Duine框架是一套以Java语言编写的软件库，可以帮助开发者建立预测引擎。Duine提供混合算法配置，即算法可根据数据情况，在基于内容的推荐和协同过滤中动态转换。例如在冷启动（比如尚无任何评价的时候）条件下，它侧重基于内容的分析法，推荐模块主要通过算法，从用户资料和商品信息中提取信息、计算预测值，主要包括以下几种方法：协同过滤法、基于实例的推理（用户给出相似评分的商品）和GenreLMS（对分类的推理）。Duine具有一个反馈处理器模块，它以增强预测为目标，利用程序学习和获取用户的显性和隐性反馈，用算法进行处理后用以更新用户的资料[18]。

6 大数据推荐系统研究面临的问题

6.1 特征提取问题

推荐系统的推荐对象种类丰富，例如新闻、博客等文本类对象，视频、图片、音乐等多媒体对象以及可以用文本描述的一些实体对象等。如何对这些推荐对象进行特征提取一直是学术界和工业界的热门研究课题。对于文本类对象，可以借助信息检索领域已经成熟的文本特征提取技术来提取特征。对于多媒体对象，由于需要结合多媒体内容分析领域的相关技术来提取特征，而多媒体内容分析技术目前在学术界和工业界还有待完善，因此多媒体对象的特征提取是推荐系统目前面临的一大难题[19]。此外，推荐对象特征的区分度对推荐系统的性能有非常重要的影响。目前还缺乏特别有效的提高特征区分度的方法。

6.2 数据稀疏问题

现有的大多数推荐算法都是基于用户—物品评分矩阵数据，数据的稀疏性问题主要是指用户—物品评分矩阵的稀疏性，即用户与物品的交互行为太少。一个大型网站可能拥有上亿数量级的用户和物品，飙升的用户评分数据总量在面对增长更快的“用户—物品评价矩阵”时，仍然只占极少的一部分，推荐系统研究中的经典数据集MovieLens的稀疏度仅4.5%，Netflix百万大赛中提供的音乐数据集的稀疏度是1.2%。这些都是已经处理过的数据集，实际上真实数据集的稀疏度都远远低于1%。例如，Bibsonomy的稀疏度是0.35%，Delicious的稀疏度是0.046%，淘宝网数据的稀疏度甚至仅在0.01%左右[19]。根据经验，数据集中用户行为数据越多，推荐算法的精准度越高，性能也越好。若数据集非常稀疏，只包含极少量的用户行为数据，推荐算法的准确度会大打折扣，极容易导致推荐算法的过拟合，影响算法的性能。

6.3 冷启动问题

冷启动问题是推荐系统所面临的最大问题之一。冷启动问题总的来说可以分为3类：系统冷启动问题、新用户问题和新物品问题。系统冷启动问题指的是由于数据过于稀疏，“用户—物品评分矩阵”的密度太低，导致推荐系统得到的推荐结果准确性极低。新物品问题是由于新的物品缺少用户对该物品的评分，这类物品很难通过推荐系统被推荐给用户，用户难以对这些物品评分，从而形成恶性循环，导致一些新物品始终无法有效推荐。新物品问题对不同的推荐系统影响程度不同：对于用户可以通过多种方式查找物品的网站，新物

品问题并没有太大影响，如电影推荐系统等，因为用户可以有多种途径找到电影观看并评分；而对于一些推荐是主要获取物品途径的网站，新物品问题会对推荐系统造成严重影响。通常解决这个问题的途径是激励或者雇佣少量用户对每一个新物品进行评分。新用户问题是目前对现实推荐系统挑战最大的冷启动问题：当一个新的用户使用推荐系统时，他没有对任何项目进行评分，因此系统无法对其进行个性化推荐；即使当新用户开始对少量项目进行评分时，由于评分太少，系统依然无法给出精确的推荐，这甚至会导致用户因为推荐体验不佳而停止使用推荐系统[20]。当前解决新用户问题主要是通过结合基于内容和基于用户特征的方法，掌握用户的统计特征和兴趣特征，在用户只有少量评分甚至没有评分时做出比较准确的推荐。

6.4 可扩展性问题

扩展性问题是推荐系统面临的又一难题，特别是随着大数据时代的到来，用户数与物品数飞涨，传统推荐系统会随着问题规模的扩大而效率大大降低。花费大量时间才能得到推荐结果是难以接受的，特别是对于一些实时性要求较高的在线推荐系统。使用基于内存的推荐系统，用户或者物品间的相似度计算会耗费大量时间；使用基于模型的推荐系统，利用机器学习算法学习模型参数同样会耗费大量时间，这里学习时间主要用在求解全局最优问题上。解决扩展性问题，工业界一般采取的方法是线下学习、线上使用：先通过离线数据事先算好用户/物品间相似度或者模型参数，然后线上只需要利用这些算好的数值进行推荐[20]。但是这并没有从根本上提高推荐算法的效率，Sarwar等人2002年提出了一种增量SVD协同过滤算法，当评分矩阵中增加若干新分值时，系统不用对整个矩阵重新计算，而只需要进行少量计算对原模型进行调整，因此大大加快了模型的更新速度。同时，若干文献提出使用聚类的方式解决扩展性问题，通过聚类能有效减少用户和物品规模，但是这样会一定程度地降低推荐精度。在求解模型全局优化问题上，学者也做了大量工作，希望能加快收敛速度，例如人们提出了并行的随机梯度下降法和交替最小二乘法等。

7 总结与展望

随着互联网的飞速发展，人们对于个性化的信息需求已经非常急切，推荐系统的出现可以很好地解决用户在使用互联网和电子商务网站时的“信息爆炸”问题。本文主要针对互联网大数据时代推荐系统的产生和发展现状、领域需求和系统架构、用户建模和推荐引擎、大数据时代推荐系统的特点挑战和关键技术、开源的大数据推荐软件、大数据推荐系统研究面临的问题等进行了介绍。

大数据推荐系统的未来研究方向主要在以下几个方面。

- 从系统推荐到社会推荐，即在推荐的过程中，除了考虑用户的历史行为信息，还需要利用用户的社会网络信息来增强推荐的效果；同时，在进行社会网络上的人与人之间的推荐时，也要综合利用用户的历史行为信息，做到社会网络和历史行为信息的互相利用和推荐效果的相互增强。
- 从以精确性为中心到综合考虑精确性、多样性和新颖性的评估体系。
- 从单一数据源到交叉融合数据平台，比如依据用户的跨网站行为数据，解决某一网站上的冷启动推荐问题。
- 从高速服务器到并行处理到云计算。

● 从静态算法到动态增量算法、自适应算法，从脆弱算法到顽健算法。

参考文献

[1] 曾春, 邢春晓, 周立柱. 个性化服务技术综述. 软件学报, 2002(10): 1952~1961
Zeng C, Xing C X, Zhou L Z. A survey of personalization technology. Journal of Software, 2002(10): 1952~1961

[2] Bell R M, Koren Y. Lessons from the Netflix prize challenge. ACM SIGKDD Explorations Newsletter, 2007, 9(2): 75~79

[3] Rendle S. Factorization machines with libFM. ACM Transactions on Intelligent Systems & Technology, 2012, 3(3): 451~458

[4] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009: 421425

[5] Chee S H S, Han J, Wang K. Rectree: an efficient collaborative filtering method. Proceedings of Data Warehousing and Knowledge Discovery: Third International Conference, Munich, Germany, 2001

[6] Connor M, Herlocker J. Clustering items for collaborative filtering. Proceedings of ACM SIGIR Workshop on Recommender Systems, New Orleans, Louisiana, USA , 2001

[7] Ungar L H, Foster D P. Clustering methods for collaborative filtering. Proceedings of AAAI Workshop on Recommendation Systems, Madison, Wisconsin, USA, 1998

[8] Miyahara K, Pazzani M J. Collaborative filtering with the simple bayesian classifier. Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, Melbourne, Australia, 2000: 679~689

[9] Miyahara K, Pazzani M J. Improvement of collaborative filtering with the simple bayesian classifier. IPSJ Journal, 2002,43(11): 3429~3437

[10] Vucetic S, Obradovic Z. Collaborative filtering using a regression-based approach. Knowledge and Information Systems, 2005, 7(1): 1~22

[11] Paterek A. Improving regularized singular value decomposition for collaborative filtering. Statistics, 2007: 2~5

[12] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, 2008: 426~434

[13] Lee D, Seung H. Algorithms for non-negative matrix factorization. Proceedings of Neural Information Processing Systems, Denver, Colorado, USA, 2000

[14] Sun J T, Zeng H J, Liu H, *et al*. CubeSVD: a novel approach to personalized Web search. Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, 2005: 382~390

[15] Steffen R, Leandro B M, Alexandros N, *et al*. Learning optimal ranking with tensor factorization for tag recommendation. Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Paris, France, 2009: 727~736

[16] 王俞翔. 面向大数据集的推荐系统研究(硕士学位论文). 秦皇岛: 燕山大学, 2014
Wang Y X. Research on recommender system for big dataset (master dissertation). Qinhuangdao: Yanshan University, 2014

[17] 黄宜华. 大数据机器学习系统研究进展. 大数据, 2015004
Huang Y H. Research progress on big data machine learning system. Big Data Research, 2015004

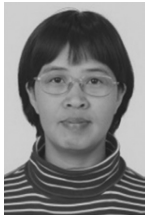
[18] 米可菲, 张勇, 邢春晓等. 面向大数据的开源推荐系统分析. 计算机与数字工程, 2013, 41(10): 1563~1566
Feben T, Zhang Y, Xing C X, *et al*. An analysis of open source recommender systems in the big data era. Computer and Digital Engineering. 2013, 41(10): 1563~1566

[19] 孙远帅. 基于大数据的推荐算法研究(硕士学位论文). 厦门: 厦门大学, 2014
Sun Y S. Recommendation algorithms in the big data era (master dissertation). Xiamen: Xiamen University, 2014

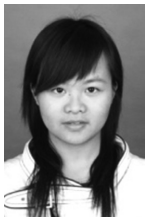
[20] 刘士琛. 面向推荐系统的关键问题研究及应用(博士学位论文). 合肥: 中国科学技术大学, 2014

Liu S C. Research on the key issues for the recommender systems (doctor dissertation). Hefei: University of Science and Technology of China, 2014

作者简介



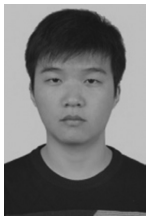
李翠平，女，中国人民大学信息学院教授、博士生导师，中国计算机学会杰出会员，中国计算机学会大数据专家委员会、数据库专家委员会委员。目前研究方向为数据仓库、数据挖掘、社会网络分析和社会媒体推荐等。主持和参与国家自然科学基金、“973”计划、“863”计划等10多项国家级和省部级项目，在国内外重要期刊和国际会议上发表论文50多篇。



蓝梦微，女，中国人民大学信息学院博士生，CCF学生会员，主要研究领域为推荐系统、数据挖掘、大数据分析。

邹本友，男，中国人民大学信息学院博士生，CCF学生会员，主要研究领域为推荐系统、数据挖掘、大数据分析。

王绍卿，男，中国人民大学信息学院博士生，CCF学生会员，主要研究领域为推荐系统、数据挖掘、大数据分析。



赵衍衍，男，中国人民大学信息学院博士生，CCF学生会员，主要研究领域为推荐系统、数据挖掘、大数据分析。

收稿日期：2015-08-11

基金项目：国家基础研究发展计划（“973”计划）基金资助项目（No.2014CB340402），国家高技术研究发展计划（“863”计划）基金资助项目（No.2014AA015204），国家自然科学基金资助项目（No.61272137, No. 61033010, No.61202114），国家社会科学基金资助项目（No.12&ZD220），国家高等学校学科创新引智计划（“111”计划）基金资助项目

Foundation Items: National Basic Research Program of China (973 Program) (No.2014CB340402), National High Technology Research and Development Program of China (863 Program) (No.2014AA015204), The National Natural Science Foundation of China(No.61272137, No.61033010, No.61202114), The National Social Science of Foundation of China (No.12&ZD220), The Project of Attracting Talents of Discipline to National Universities (111 Project)

论文引用格式：李翠平, 蓝梦微, 邹本友等. 大数据与推荐系统. 大数据, 2015026

Li C P, Lan M W, Zou B Y, *et al*. Big data and recommendation system. Big Data Research, 2015026