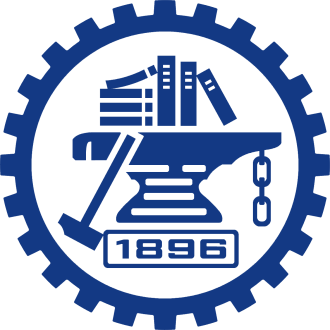
申请上海交通大学硕士学位[论文](#_OFDM技术简介_1)

机票个性化推荐方法

****

**学校：** 上海交通大学

**院系：** 电子信息与电气工程学院

**班级：** B1303392

**学号：** 1120339034

**硕士生：** 杨芳洲

**专业：** 计算机科学与技术

**导师：** 曹健

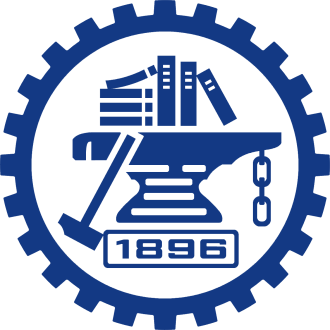
**上海交通大学电子信息与电气工程学院**

**2016年1月**

**A Dissertation Submitted to Shanghai Jiao Tong University for the**

**Degree of Master**

Personalized Flight Recommender

****

**Author:** Yang, Fangzhou

**Specialty:** Computer Science

**Advisor:** Cao, Jian

School of Electronic Information and Electrical Engineering

Shanghai Jiao Tong University

Shanghai, P.R.China

January, 2016

**上海交通大学**

**学位论文原创性声明**

本人郑重声明：所呈交的学位论文《机票个性化推荐系统的研究与探索》，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

**上海交通大学**

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

**保密**□，在年解密后适用本授权书。

本学位论文属于

**不保密**□**√**。

（请在以上方框内打“**√**”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

**机票个性化推荐方法**

摘要

随着互联网的发展以及大数据时代的到来，信息爆炸所带来的信息过载的问题也越发明显。推荐系统作为解决信息过载问题的一个有效解决方案，能够有效地为用户个性化地推荐其感兴趣的产品与信息，其在过去数十年中也逐渐成为一个重要的研究热点并被广泛地应用到工业领域。

论文主要研究推荐系统在机票个性化推荐问题中的应用。与传统的推荐系统的推荐对象，如电影、书籍等具有相对固定属性的静态商品不同，机票是属于易受时间影响的，且价格敏感的动态商品。同一张机票在距离起飞的不同时间有着较大的价格波动，而不同的机票价格波动将直接影响用户的购买行为。

文中通过研究和分析用户的历史机票订单数据特征，提出了一种基于用户偏好模型的机票个性化推荐算法，该方法类似于基于内容推荐的KNN的方法，并且引入了信息熵的概念，用来计算用户在不同航线中各个机票特征上的个性化的偏好权重。此外，针对用户在非活跃航线上数据稀疏的问题，还提出了基于航线的协同过滤的算法，用来帮助用户进行跨航线的偏好学习。

此外，文中还提出了一种基于选择模型的机票个性化推荐算法。通过对用户在机票历史订单的成对选择分析，来建立用户选择机票时的效用目标函数及其优化问题。同时，还针对机票动态商品属性，提出了一种结合回归隐语义模型的效用函数模型，其能够很好地结合航班的固定信息和机票本身的动态属性特征，通过隐式空间特征更精准地刻画用户对机票的偏好。

最后，论文还结合了大数据技术，提出了一个面向大数据的机票个性化推荐系统设计框架，包含了数据层，应用层，以及包含了一个离线计算单元和一个在线计算单元的推荐逻辑层。同时，还对文中提出的两种机票个性化推荐算法进行了基于Spark并行计算框架的并行化设计与实现。

**关键词**：机票个性化推荐，推荐算法，隐式反馈，协同过滤，选择模型

Personalized Flight Recommender

ABSTRACT

As the internet develops and the time of big data comes, more and more data is produced. Information explosion has brought the problem of information overload. As an effective solution for information overload problem, recommendation system can recommend users’ preferred and interested products and information from massive data. It has become a very popular research topic and widely applied in industry in the last decades.

In this thesis, we research the personalized flight recommendation problem. Different as the recommended items in traditional recommendation system, such as movie and book having static characteristics, flight tickets are time-variant and price-sensitive dynamic products. The price flight tickets will change from time to time before the take-off date, and different price will effect directly users’ purchase behaviors.

In this thesis, we propose a personalized flight recommendation algorithm based on user preference model. This algorithm is similar to a KNN content-based recommendation algorithm, but it introduces entropy to calculate personalized weights of users’ feature preferences. Besides, we have proposed an airline-based collaborative filtering method for solving data sparsity problem in user’ inactive airlines.

In addition, we propose another personalized flight recommendation algorithm based on paired-choice model. We build the target function and optimizing problem for the user’ choice utility of flight tickets by paired-choice analysis on users’ historical orders. At the same time, we design a regression latent factor model for the utility function for flight tickets. This model can combine the static information of flights and the dynamic features of tickets, and has a stronger expression for user preferences through latent factors.

At last, we design a system framework for personalized flight recommender based on big data techniques. Moreover, parallelized computing processes for our personalized flight recommendation algorithms are designed and implemented on Spark parallel computing framework.

**Keywords:** Personalized Flight Recommendation, Recommendation Algorithm, Implicit feedback, Collaborative filtering, Paired-choice Model.

目录

[摘要 I](#_Toc439109880)

[ABSTRACT II](#_Toc439109881)

[第一章 绪论 5](#_Toc439109882)

[1.1 研究背景及意义 5](#_Toc439109883)

[1.2 国内外研究现状 7](#_Toc439109884)

[1.2.1 推荐系统的研究现状 7](#_Toc439109885)

[1.2.2 机票个性化搜索的研究现状 8](#_Toc439109886)

[1.3 本文研究内容和结构安排 9](#_Toc439109887)

[1.3.1 研究内容 9](#_Toc439109888)

[1.3.2 文章结构安排 10](#_Toc439109889)

[第二章 相关技术 11](#_Toc439109890)

[2.1 隐式反馈推荐系统 11](#_Toc439109891)

[2.1.1 隐式反馈推荐的关键问题 11](#_Toc439109892)

[2.1.2 面向隐式反馈的推荐技术 13](#_Toc439109893)

[2.1.3 隐式反馈推荐方法的评价指标 15](#_Toc439109894)

[2.2 大数据技术原理与应用 16](#_Toc439109895)

[2.2.1 大数据的简介 16](#_Toc439109896)

[2.2.2 大数据的处理流程 17](#_Toc439109897)

[2.2.3 大数据的关键技术 18](#_Toc439109898)

[2.2.4 大数据技术在在线旅游系统中的应用 20](#_Toc439109899)

[2.3 本章小结 20](#_Toc439109900)

[第三章 基于用户偏好模型的机票个性化推荐算法 21](#_Toc439109901)

[3.1 机票的数据特征 21](#_Toc439109902)

[3.2 用户偏好模型及推荐算法 22](#_Toc439109903)

[3.2.1 基于信息熵的用户偏好分析 22](#_Toc439109904)

[3.2.2 基于信息熵的用户偏好模型 25](#_Toc439109905)

[3.3 结合协同过滤的跨航线偏好学习 29](#_Toc439109906)

[3.4 相关实验与结果分析 33](#_Toc439109907)

[3.4.1 实验数据集 33](#_Toc439109908)

[3.4.2 评价指标 35](#_Toc439109909)

[3.4.3 相关的比较算法 35](#_Toc439109910)

[3.4.4 实验结果与分析 36](#_Toc439109911)

[3.5 本章小结 41](#_Toc439109912)

[第四章 基于选择模型的机票个性化推荐算法 43](#_Toc439109913)

[4.1 选择模型 43](#_Toc439109914)

[4.1.1 选择分析（Paired Choice Analysis） 43](#_Toc439109915)

[4.1.2 参数的先验 45](#_Toc439109916)

[4.1.3 优化问题 45](#_Toc439109917)

[4.2 机票选择的效用函数建模 46](#_Toc439109918)

[4.2.1 基于显性偏好的效用函数模型 46](#_Toc439109919)

[4.2.2 基于RLFM的效用函数模型 47](#_Toc439109920)

[4.3 基于选择模型的学习和推论 49](#_Toc439109921)

[4.3.1 参数估计 49](#_Toc439109922)

[4.3.2 机票推荐和预测 54](#_Toc439109923)

[4.4 相关实验与结果分析 54](#_Toc439109924)

[4.4.1 实验数据集 54](#_Toc439109925)

[4.4.2 评价指标 54](#_Toc439109926)

[4.4.3 相关的比较算法 55](#_Toc439109927)

[4.4.4 实验结果与分析 56](#_Toc439109928)

[4.5 本章小结 59](#_Toc439109929)

[第五章 面向大数据的机票个性化推荐系统设计 61](#_Toc439109930)

[5.1 面向大数据的机票个性化推荐系统框架设计 61](#_Toc439109931)

[5.1.1 面向在线旅游服务的大数据平台 61](#_Toc439109932)

[5.1.2 面向大数据的机票个性化推荐系统设计 63](#_Toc439109933)

[5.2 结合Spark的推荐算法并行化实现 64](#_Toc439109934)

[5.2.1 基于用户偏好模型的机票个性化推荐算法的并行化实现 64](#_Toc439109935)

[5.2.2 基于选择模型的机票个性化推荐算法的并行化实现 66](#_Toc439109936)

[5.2.3 推荐算法并行化的性能测试 68](#_Toc439109937)

[5.3 本章小结 70](#_Toc439109938)

[第六章 全文总结与展望 71](#_Toc439109939)

[6.1 本文工作总结 71](#_Toc439109940)

[6.2 未来工作展望 71](#_Toc439109941)

[参考文献 73](#_Toc439109942)

[致谢 77](#_Toc439109943)

[附录一符号与标记 79](#_Toc439109944)

[附录二英文缩略语表 81](#_Toc439109945)

[攻读硕士学位期间已发表的论文 83](#_Toc439109946)

# 绪论

* 1. 研究背景及意义

随着国内在线旅行社（Online Travel Agent，OTA）市场的快速发展，越来越多的人开始习惯于在网上购买旅行产品。在线机票搜索引擎作为OTA市场中重要的一块核心业务，为乘坐飞机出行的人们提供了极大的便利。乘客在网页上输入出行时间以及起飞到达城市，即可搜索所有出满足条件的待售的机票，之后只需在搜索结果页中筛选出想要购买的机票便可在线完成机票的预订和交易。相较于传统的通过线下的旅行社以及航空公司购买机票的途径，在线机票搜索引擎为旅客提供了更加全面的航班信息的渠道，使用户在选择机票的过程中操作更加简单，信息更加及时，渠道更加透明，从而得到最能够满足自己出行需求的机票。

然而，丰富的选择也可能带来了一个信息过载的问题。一条热门航线上直达航班的机票搜索结果可达几十上百条，包含了不同的航班，航司（航空公司），舱位，价格，退改签等特征属性。若是考虑往返以及中转城市，则其机票结果的组合可达成千上万条。如果此时搜索引擎能够在众多的搜索结果中快速有效地帮助用户选择出最优的机票，一方面可以减少用户购买机票时的费力度，增强用户体验，另一方面可以潜在地提高转化率，增加销量。在众多的提供在线预订机票的OTA公司中，已经有许多公司开始在搜索结果采用一些方法，从而让用户能够更快捷高效地选择到心仪的机票。例如，Google Flight[1]在搜索结果中为订票的用户提供了最佳机票的推荐，被推荐的通常是所有搜索结果中价格实惠，同时起飞时间比较适合出行的机票；携程旅行网Ctrip[2]在机票的搜索结果中会为一些当前销售较多的一些热门机票打上标签来帮助用户进行快速选择。

上述所说的这些方法主要是经验式的为所有用户提供的一种普适性的推荐方法。然而不同的用户对机票有着不同的偏好和需求， 例如有的用户偏好低价，总是优先选择最低的价格，有的用户是某个航空公司的会员，出行偏向这个航司的航班从而获得更多的积分。因此，如何为不同的用户个性化地推荐出最能够满足其偏好和出行需求的机票，成为一个具有重要实践意义的问题。

推荐系统[3-6]作为解决“信息超载”问题的一个非常有潜力的办法，能够分析用户的兴趣、需求等，将用户最有可能感兴趣的产品个性化地推荐给用户。随着信息时代的发展，推荐系统已广泛地应用于工业中的各个领域，如Amazon根据用户的购买行为对用户进行书籍的推荐[7]，NetFlix根据用户对电影的评分进行电影的推荐[8]等等。然而，相对于传统的推荐系统推荐的电影，书籍等具有固定属性的静态商品不同，机票是属于受时间影响的价格敏感的动态商品。这里被我们称为的动态商品，其区别于静态商品的主要不同体现在以下的两个方面:

1. 特征的波动性

相对于静态商品，动态商品的某一特征在不同的时间内具有较为明显的波动性。 对于机票来说，同一航班同一舱位的机票在距离起飞时刻的不同时间有着较大的价格波动，这种价格与航空公司的实时定价策略有关，较难预测。

1. 特征的敏感性

商品的波动性特征会直接影响用户的购买行为决策。同样对于机票来说，当用户选择搜索机票时，价格将是用户非常重要的决策因素，用户总是会倾向于选择满足其需求的价格低的机票。

针对以上机票的价格敏感的动态特性，传统的基于协同过滤的推荐方法，例如矩阵分解技术以及Item-based的协同过滤方法并不能直接地应用到机票推荐中。因为这里并没有一个固定的物品集合，并且机票的价格将会随着时间波动并影响着用户对该机票的选择和购买决策。

尽管目前推荐系统作为当前研究的一大热门，已经发展出了各个分支并广泛的应用到工业界的各个领域。然而，目前推荐系统在机票的个性化搜索领域的应用和研究却相对较少。在2004年，Lorcan Coyle提出了一种基于案例推理（Case-Based Reasoning，CBR）的机票个性化推荐算法[9]。随着近十年来推荐系统领域的研究快速的发展，很多新的推荐技术和机器学习的方法可以被应用到机票个性化搜索的领域中。

另一方面，随着近年来大数据技术的迅猛发展，大数据已经成为IT领域的另一大热点，并开始渗透到金融，电子商务，移动互联网，医疗，交通等各个领域。而大数据技术对海量数据极强的处理，存储，分析的能力也将为旅游服务信息化提供强有力的技术支持。同时大数据技术为推荐系统也带来了更快更有效的数据处理能力，基于Map-Reduce计算框架[10]的分布式的推荐算法使得推荐系统具有更好的并行性和扩展性。

综上，研究推荐系统在个性化机票搜索领域的应用，并结合大数据的相关技术，无论在工业上OTA行业的应用领域还是在学术上推荐系统的研究领域都具有重要的意义。

* 1. 国内外研究现状

### 推荐系统的研究现状

推荐系统[3-6]，又称为推荐引擎，指的是一种能够分析和预测用户对物品的偏好的信息过滤的系统。在1997年，Resnik和Varian[4]首次将推荐系统作为一个独立的概念提出。随后随着互联网的普及和Web技术的快速发展，推荐系统在电子商务领域得到的广泛地应用和研究并且带来了巨大的商业利益。

推荐系统作为当前的一个学术领域的研究热点，在过去的数十年得到了突飞猛进的发展并逐渐发展成一个独立的领域。根据不同的推荐方法，我们可以将推荐系统分为如下几类[11]：基于内容（content-based）的推荐系统，协同过滤（collaborative filtering）推荐系统，基于网络结构（network-based）的推荐系统以及混合（Hybrid）推荐系统。

基于内容的推荐方法源于信息获取[12]和信息过滤[13]领域，该方法根据用户的历史选择的对象，从推荐对象中选出与历史选择对象特征相似的对象作为该用户的推荐结果。在传统的基于信息获取的推荐方法中，最常用的方法为TF-IDF[14]，通过文本表征的方法来抽取特征计算相似性。此外，一些结合基于统计学习和机器学习模型，如Bayes分类器[15]、神经网络[16]等技术也可以应用到基于内容的推荐方法。

协同过滤的推荐方法源于20世纪90年代，并逐渐发展成为推荐系统领域最为成功的一种方法策略。该方法的核心思想是利用用户和对象间的历史信息计算用户或者对象的相似性，通过找出与目标用户相似的其他用户的偏好，或者目标用户在与推荐对象相似的其他对象上的偏好，来为用户进行推荐。协同过滤的方法可以分为基于记忆（memory-based）[17, 18]和基于模型（model-based）[19-21]的两类。

基于网络的推荐方法忽略用户的属性和物品的内容特征，将用户和物品看成抽象的节点，根据用户对物品的选择关系，利用用户-物品二部分网络（bipartite network）来挖掘用户和物品的关联关系[22-24]。

混合推荐方法是针对基于内容，协同过滤，以及基于网络结构的推荐算法在实际应用过程中的缺陷提出的一种混合的推荐方法。通过对不同的推荐技术的合理的混合使用[25-27]，可以获得比独立的推荐系统更好的准确率。

通常，推荐系统按照其获取数据的方式还可以将其分为显式反馈推荐系统和隐式反馈推荐系统：

显式反馈是用户主动输入，将相关信息提交给系统，例如在NetFlix[8]中用户可以给进行电影评分，在MyYahoo和WebWatcher[28]中用户可以输入感兴趣的话题。通过显性反馈来获取用户的信息，是一种非常简单直接又高效的方法，因为用户主动输入的信息通常比较全面、具体，能够比较准确地反应出用户的偏好和需求。

隐式反馈是指通过追踪用户的在网站中行为操作来分析用户的偏好和需求。用户在浏览网站时的很多行为动作通常都能暗示出用户的一些潜在偏好，例如查询，点击，浏览，收藏某些特定种类的商品，可以说明用户可能对这类商品比较感兴趣。另一方面，我们还可以分析利用用户的历史购订单数据，来预测用户未来感兴趣的商品和潜在的需求。通过隐式反馈的方式，我们可以在用户正常使用网站服务的同时收集用户的数据。然而，通过隐式反馈收集到的用户信息常常是模糊的，不够直观的，我们常常需要对隐式反馈收集的数据进行一定的挖掘和处理。

在机票个性化推荐的问题中，我们无法收集用户对所浏览和购买的机票的类似于显性评分的显式反馈， 推荐系统所能收集使用的主要数据为用户的订单数据和行为数据，因此我们可以将机票个性化推荐问题看作是隐式反馈的推荐问题。

### 机票个性化搜索的研究现状

目前，对机票个性化搜索的研究和工业界上的应用还相对较少，而机票个性化搜索在工业界上的在线应用更是近乎为零。虽然已经有很多OTA公司在提供机票在线预订服务时开始关注用户搜索机票时的体验，并在搜索结果和搜索页面中做了许多的优化，例如通过在搜索页面中置顶最佳的航班，或者为热门的机票打上标签来推荐给用户，从而帮助用户快速的找到满足出行需求的最佳机票，但是这些方法更多的是一种基于业务经验上提供的对于所有用户的普适的推荐。然而不同的用户有着不同的偏好和出行需求，因此我们需要一种针对于每个用户的个性化的机票搜索、推荐的方法。

在2004年，Lorcan Coyle[9]提出了一种基于案例推理（CBR）的机票个性化推荐方法。案例推理[29]是一种通过分析，调整，使用历史案例来解决新的案例的人工智能方法。其过程主要包括四个步骤：案例的获取，案例的复用，案例的修正和案例的保存。

基于CBR的机票个性化推荐算法[9]的主要思想是将用户过去购买过的机票作为历史案例，当用户进行新的一次机票搜索时，将搜索结果中与历史案例最相似的机票推荐给用户。不同机票的相似度函数被定义为，其中Q，C表示两张机票，和**F**分别是抽取出的机票的一个特征和对应的特征集合， 是用户对于这个特征的权重， 是机票Q和C在特征上的相似度。整个机票个性化推荐的CBR的过程如下：

1. 案例的获取：将用户过去购买过的机票作为历史案例
2. 案例的复用：当用户进行新的一次机票搜索时，利用相似度函数计算搜索结果中所有机票与该用户的历史案例相似度，并将最相似的机票推荐给用户。
3. 案例的修正：如果用户选择的机票与系统推荐的机票不相符，则调整相似度函数中的特征权重，使其选择的机票与通过调整后的相似度函数计算出的推荐结果想匹配。
4. 案例的保存：将用户该次的选择保存为用户新的一个历史案例。

这种基于CBR的机票个性化推荐算法更多的像是一种基于内容的推荐，其强调使用结构化的表示和相似度的计算来进行推荐。而过去的十年中，随着推荐系统的研究领域的迅猛发展，像一些新的协同过滤方法以及基于贝叶斯排序等一些新的推荐技术可以被应用到机票个性化搜索领域，同时结合最新的大数据技术，从而获得一个性能更强效率更高精度更好的一个应用于机票个性化搜索的推荐系统。

* 1. 本文研究内容和结构安排
     1. 研究内容

本文主要研究推荐系统在机票个性化搜索中的应用，其主要研究的内容有：

1. 提出了一种基于用户偏好模型的机票个性化推荐算法

通过分析用户的历史机票订单数据特征，我们提出了一种基于用户偏好模型的机票个性化推荐算法，对用户在某条航线上的搜索结果进行推荐。同时，针对用户在非活跃航线上数据稀疏的问题，我们还提出了一种基于航线的协同过滤的算法，进行用户在不同航线上偏好的跨航线的学习。

1. 提出了一种基于选择模型的机票个性化推荐算法

根据用户在购买预订机票时的决策过程，我们还提出了一种基于选择模型的机票个性化推荐算法，该方法通过对用户机票历史订单的成对选择分析，来建立用户的偏好目标函数和优化问题。此外，针对机票价格波动性和价格敏感性的动态商品属性，我们还结合了回归隐语义模型（RLFM），来为选择模型中用户选择机票时的效用函数进行函数建模。

1. 基于大数据技术的机票个性化推荐系统架构的设计

设计了一个面向大数据的机票个性化推荐系统的系统框架，同时针对本文提出的两种机票个性化推荐算法，我们基于Spark并行计算框架对其进行了并行化的设计与实现，并且在Spark集群上进行了相关并行化性能的评估和测试。

* + 1. 文章结构安排

本文共有六个章节，各个章节的组织结构如下：

第一章绪论，首先阐述了文章的研究背景和意义。同时分析了国内外目前对于推荐系统以及机票个性化推荐的研究现状。最后介绍本文的主要的研究内容以及结构的安排。

第二章相关技术理论基础，首先对隐式反馈的推荐系统做了详细的介绍，研究并总结了隐式反馈推荐系统的关键问题，面向隐式反馈推荐系统的推荐技术以及隐式反馈推荐算法的相关评价指标。其次，本章还介绍了大数据技术以及其在在线旅游系统中的应用。

第三章基于用户偏好模型的机票个性化推荐算法，首先根据用户的机票订单数据统计和分析机票的数据特征，并抽取了相关的机票显性特征属性。其次，结合机票的数据特征，提出了一种基于用户偏好模型的机票个性化推荐算法，对用户在某条航线上的搜索结果进行推荐。接着，针对用户在非活跃航线上数据稀疏的问题，提出了了基于航线的协同过滤算法，来帮助用户在不同的航线上进行跨航线的偏好学习。本章最后通过真实的用户数据对机票个性化推荐算法进行了一系列的实验评估和比较分析。

第四章基于选择模型的机票个性化推荐算法，首先根据选择模型，通过对用户在机票历史订单的成对选择分析，建立了用户选择机票时效用的目标函数及优化问题。接着，引入了简单回归模型和回归隐语义模型两种模型对效用函数进行函数建模，并针对建立好的目标函数和优化问题，提出了随机梯度下降的方法来对模型中的待估参数进行学习和训练。最后，通过相关的实验和测试，对不同的机票个性化推荐算法进行了进一步的评估和分析。

第五章面向大数据的机票个性化推荐系统设计，首先结合面向旅游服务的大数据平台，设计了一个面向大数据技术的机票个性化推荐系统的系统框架。接着，详细介绍了基于Spark的分布式个性化推荐算法的设计与实现，并通过相关实验对其扩展性和并行性能进行了测试。

第六章全文总结与展望，总结本文所完成的主要工作，并提出对未来的研究工作的展望。

# 相关技术

## 隐式反馈推荐系统

互联网时代的到来给我们带来了丰富和高效的资讯和信息，然而，这些海量的信息也带来了“信息超载”的问题。推荐系统作为解决“信息超载”问题的一个非常有效的办法，可以收集用户数据，分析用户偏好和需求，并且精确高效地为用户从海量信息中过滤筛选出用户最感兴趣的信息或者商品推荐给用户。目前，推荐系统已经成为一个非常热门的研究领域，并且广泛地应用到各个工业界领域中。

推荐系统通过对用户偏好和推荐对象进行建模，并结合推荐算法找出与用户偏好最相符的对象作为呈现给用户的推荐结果。其中，用户的偏好可以通过用户的商品的评分、评论等一些用户主动提供的反馈信息来获得，这类反馈信息通常可以直观、准确地表现出用户的偏好，因而被称为显式反馈。相对于显式反馈，那些不能直观表现用户偏好的信息，如用户的购买历史记录、行为记录等，被称作为隐式反馈。目前，许多推荐系统仅仅针对于显式反馈的信息，这主要是因为隐式反馈的信息无法完全表达用户的偏好，因此我们无法直接根据隐式反馈信息对用户偏好建立模型。尽管如此，在很多实际应用的场景中，隐式反馈的信息比显示反馈的要丰富。近年来，许多专家学者针对基于隐式反馈的个性化推荐系统也进行了大量的研究。

本节主要分析和总结了当前研究中隐式反馈的推荐系统中的关键问题以及面向隐式反馈的推荐技术。

### 隐式反馈推荐的关键问题

在推荐系统中，显式反馈能够直观地反映用户的偏好和需求，通常被作为反映用户偏好的真实值（Ground Truth）。显式反馈中最常见的类型是用户对商品的显性评分，例如Amazon收集用户对购买的商品的评分[7]，Netflix收集用户对看过的电影的评分[8]；另一种常见的类型是收集用户的标签数据，从而获得更加精细的用户偏好信息。然而显式反馈在很多实际的应用场景中常常是不可用的，一方面是数据来源的问题，在很多应用场景中用户并不愿意花费额外的时间主动提供反馈信息；另一方面用户在某些情况下可能并不确定自己的偏好，缺乏对事物的客观评价的能力，同时其偏好和需求也会随时间发生变化，而新的偏好和需求常常不能够及时地通过主动提供的反馈信息进行更新。

相比较之下，隐式反馈能够在一定程度上缓解上述显式反馈中出现的问题。隐式反馈收集和追踪用户的浏览数据，行为数据，能够在不干扰用户正常使用应用功能的情况下收集用户的信息。同时，隐式反馈的信息是用户访问系统时连续产生的，无需用户主动来提供，并且这类信息广泛地存在于各类系统应用之中。

尽管如此，相对于显式反馈，针对基于隐式反馈的推荐系统依然存在着许多的问题和挑战，Hu等人[30]将其归纳为：

1. **缺少负反馈**

通过用户的浏览记录和历史购买记录等隐式反馈的信息，我们可以推断用户喜欢那些购买过的物品，或者对那些浏览过的商品保有较大的兴趣。然而，我们无法推断出用户不喜欢的物品。例如，当一个用户没有观看某部电影，可能是因为他不喜欢这部电影，也可能是因为他不知道这部电影。这种信息的不对称性不会出现在显式反馈中。在显式反馈中，用户可以明确告诉喜欢或者不喜欢某件物品的信息，例如对某件商品的高低的评分。然而在隐式反馈中收集的信息通常只有正反馈信息，负反馈的信息缺失会导致用户偏好的不平衡，从而导致推荐结果的不准确。因此，对于缺失的负反馈信息的处理是隐式反馈推荐方法需要解决的一个重点问题。

1. **数据噪音**

隐式反馈的信息天生具有噪音性质。当我们被动地追踪用户的行为时，我们只能猜测其用户的偏好和真实的目的。例如，我们可以看到一个用户的购买行为，但是从这个购买行为我们并不能完全地推断出用户对这个商品具有积极的偏好。因为这个商品可能是为别人购买的，或者这个用户在购买了以后对该商品表示很失望。由于隐式反馈不是用户的主观反馈，因此通过隐式反馈获得的信息具有更大的噪音性质。

1. **偏好表现与可信度**

显式反馈中，用户可以直接表达其对某件物品的喜好程度，例如在评分系统中通常为1分（非常不喜欢）至5分（非常满意）。然而在隐式反馈中，我们需要首先对用户的行为数据进行分析，例如分析其在某件商品上的点击数，购买数量等等。更高的数值不一定能绝对地表达用户对商品的偏好更强烈，但是更高的数值从一定程度上表现出了用户对于该商品偏好更强的可信度。

1. **评估标准的有效性**

在传统的显式反馈的推荐系统中，我们通过计算预测的评分和用户真实评分的均方误差（Mean Square Error, MSE）来评估一个推荐方法的好坏。然而在隐式反馈的推荐方法中，我们需要考虑物品的可用性，物品之间的竞争关系，以及重复的反馈。因此，如何针对隐式反馈推荐方法的合理地设计一个评估标准，也是隐式反馈的推荐系统中的一个重要问题。

Jawaheer等人[31]比较了隐式反馈和显式反馈的相关特点，基于此，我们可以将显式反馈和隐式反馈的特点归纳为如下表所示：

表2-1 显式反馈与隐式反的特征

Table 2-1 Characteristics of Explicit Feedback and Implicit Feedback

|  |  |  |
| --- | --- | --- |
|  | 显式反馈 | 隐式反馈 |
| 表征用户偏好的准确性 | 高 | 低 |
| 信息的丰富度 | 低 | 高 |
| 上下文敏感 | 是 | 是 |
| 用户偏好的表达 | 正反馈和负反馈 | 只有正反馈 |
| 推荐效果的评估标准 | 绝对比较 | 相对比较 |
| 反馈信息的获取难度 | 困难 | 容易 |
| 数据噪音 | 较小 | 较大 |

### 面向隐式反馈的推荐技术

由于隐式反馈缺少负反馈以及数据噪声等缺陷，我们无法直接将传统的推荐方法直接应用到隐式反馈的信息中。针对上述的问题，近年来提出了许多基于隐式反馈的推荐方法。通过对这些方法的研究和总结，可以将这些方法分为三类：单类协同过滤的推荐，基于内容的推荐以及基于排序的推荐。

1. **单类协同过滤的推荐方法**

由于隐式反馈难以区分负样本和潜在的正样本，针对于只存在正样本的协同过滤，Pan等人[32]提出了单类协同过滤问题（One-class Collaborative Filtering，OCCF）。单类协同过滤问题的关键在于如何处理缺失的负样本数据，一种方法是对于每个样本实例设置不同的权重，把缺失的数据一起放入基于权重的模型中进行训练；另一种方法是对缺失的负样本数据进行分布假设，通过概率概率模型来避免隐式反馈中正样本和负样本失衡的问题。

对于基于权重的方法，Hu[30]提出了样本实例置信度的概念，并定义了置信度公式：，其中表示用户行为的频次，表示控制系数。从公式中我们可以看出，当用户对一个物品的行为次数越高，用户对该物品喜好的置信程度越高。将置信度作为权重放入协同过滤的目标函数，可得到单类协同过滤的目标函数：

其中为二元值，0表示未知的用户-物品对的样本，1表示正样本。该模型通过将用户行为的频次转换为置信度作为协同过滤中目标函数的权重，从而将协同过滤的方法应用到隐式反馈的推荐问题中。类似地，Pan[32]在解决单类协同过滤的问题中提出另外三种权重构造的策略。

另一种处理单类协同过滤问题的方法，即基于概率或采样的方法，其主要思想是通过对未知的用户-物品对的采样，或者概率分布的假设来产生负样本。对于负样本的采样，Pan[32]提出了三种基本的策略：以均匀概率来抽样；根据用户的活跃程度来抽样；根据物品的流行程度来抽样。对于负样本的概率模型，Sindhwani等人[33]根据缺失数据中正样本和负样本的分布概率，将缺失样本替换为正样本和负样本的概率值。

无论是从加权还是概率或者采样的角度，单类协同过滤的问题都是根据缺失的数据对负样本构建合理的假设，从而将隐式反馈转化为显式反馈，从而使得问题可以应用协同过滤的方法来进行解决。

1. **基于内容的推荐方法**

与显式反馈中基于内容的推荐方法相似，隐式反馈的基于内容的推荐通过分析物品的相关属性特征，从用户的正样本反馈中提取出用户的兴趣偏好，并构建用户画像。在为用户进行物品的推荐时，从待推荐对象中找出与用户偏好最匹配的作为推荐结果。基于内容推荐的隐式反馈中的关键是对物品特征属性的分析和定义，特征的好坏将直接印象用户偏好和推荐结果的准确性。同时，不同的物品在不同的应用场景中通常也会有很大的变化。

Pazzani等人[34]针对基于内容的推荐，提出了一种公共的方法来创建描述用户对物品喜好的用户画像，通过比较物品和用户画像来决定给用户的推荐对象。用户的画像可以表现出用户的偏好和需求，并可以通过用户的对物品操作的隐式反馈自动创建和更新。

Chu和Park[35]提出了一种对于互联网服务中的动态内容（例如新闻文章）的推荐方法，该方法基于用户画像和内容画像的所有特征，建立预测双线性回归模型来为已经存在的用户以及新用户、新物品进行推荐，很好的解决了推荐系统中动态内容的冷启动的问题。

1. **学习排序的推荐方法**

推荐系统的目标在于根据不同用户的偏好和需求，提供一个个性化的物品排序列表。因此，学习个性化推荐的列表可以看作是一个学习排序（Learning to Rank, LtR）的问题。在信息获取领域和许多机器学习社区中，学习排序问题已经吸引了广泛的研究和关注。近年来，有许多学者开始将排序模型应用到隐式反馈的推荐算法中，从而提升推荐系统中排序的效果。

Karatzoglou[36]等人将基于排序学习的推荐算法分为了：逐点排序（point-wise rank）、逐对排序（pair-wise rank）以及逐列排序（list-wise rank）的三种方法。

逐点排序方法主要是基于用户对于单个物品的选择或者偏好的分数来学习一个排序模型。Koren等人[37]提出了一种基于逐点排序的协同过滤的推荐方法，该方法将用户对于物品的反馈作为排序的序数而不是通常的数值，并对用户的反馈信息进行排序模型的建立。

逐对排序方法主要考虑用户在一对物品上的偏好的强弱，其中典型的方法是贝叶斯个性化排序[38]（Bayesian Personalized Ranking，BPR）。该方法通过对用户对物品的偏序选择对上的偏好建立贝叶斯模型，并通过sigmoid函数转化为优化目标函数的最优化问题进行求解。

逐列排序方法则是基于用户对一个列表上物品的偏好进行建模。Yue等人[39]提出了TFMAP模型，该模型通过最大化平均准确率均值（Mean Average Precision，MAP）来为每个用户在一个给定的上下文创建一个最优的排序序列。TFMAP使用张量分解（tensor factorization）技术来为含有上下文信息的隐式反馈数据进行建模。

### 隐式反馈推荐方法的评价指标

在显式反馈推荐系统中，我们常用的平均平方误差（Mean Squared Error，MSE）、均方根误差（Root Mean Squared Error，RMSE）通过计算预测的分值和用户真实评分的误差来评价推荐算法的好坏。然而，在隐式反馈推荐系统中并没有这样的显式的信息，因此，我们需要其他的方法来对隐式反馈的推荐算法的准确性进行有效的评估。这里，我们研究并总结了隐式反馈推荐系统中几种常用的评价指标：

1. AUC

AUC指标是基于分类准确度的一种评价指标，是指ROC（Receiver Operator Curve）曲线下的面积，英文全称为Area Under Curve，它能够有效地衡量一个推荐系统区分用户喜欢和不喜欢的物品的准确性。ROC曲线是由假正命中率（false positive rate）为横轴，真正命中率（true positive rate）为纵轴绘制出的一条曲线。而在推荐系统中，我们通常并不直接通过绘制曲线，而是通过近似的方法来求AUC的值。Rendle等人[38]在贝叶斯个性化排序中，对在推荐列表上的推荐对象与未被推荐的对象进行逐对构造，记为，其中，为推荐对象的集合，为未被推荐对象的结合，设所有构造的对的数量为N，在N对中，用户对推荐列表中的推荐对象的偏好大于未被推荐的对象的对的数量为，则我们有：

从该式我们可以看出，AUC为0-1取值范围内的值，值越大则表示推荐算法的准确性越高，当值大于0.5时，表示推荐算法的效果优于随机推荐的效果。

1. MAP

在隐式反馈推荐系统中，由于没有显式的信息，我们通常还可以采用基于排序的方式来对推荐算法的准确度进行评估。平均准确率均值（MAP）通过计算对于用户的推荐列表中各个对象的平均准确率（Average Precision, AP）来评估推荐效果的好坏，用户*u*的平均准确率的定义如下式所示：

其中表示物品*i*对于用户*u*的百分位排序，该值越小，则表示用户偏好的物品排序越靠前，推荐效果则越好。当MAP小于50%时，则推荐算法优于随机的推荐。

除了上述的两种常用的评价指标，还有一类是基于排序的带权重的评价指标，例如半衰期效用指标[17]（Half-Life Utility，HLU）以及折扣累计利润指标[40]（Discounted Cumulative Gain，DCG），其主要思想是当用户偏好的对象被排在前边比被排在后边具有更大的实用性，且能够能大程度地增强用户体验。

## 大数据技术原理与应用

随着移动互联网，云计算以及物联网的发展，越来越多的数据被生产出来，呈现出了一种信息爆炸的形式。针对这些海量的数据问题，大数据技术应运而生，并获得了世界范围内空前的关注。本节将介绍大数据的概念和大数据的处理流程，并介绍当前主流的一些开源的大数据技术，以及大数据技术在在线旅游系统中的应用的情况。

### 大数据的简介

大数据是一个抽象概念，目前并没有一个明确的定义。著名IT研究与顾问咨询公司高德纳（Gartner）[41]将大数据定义为：大数据是能够满足以高效，创新的方式处理信息，从而能够增强洞察力，决策力，以及自动化处理的能力的高容量，快速度，多种类的信息资产。

通常，我们认为大数据有四个V的特征，分别是容量大（Volumn），种类多（Variety），速度快（Velocity），以及价值密度低（Value）。

容量大（Volumn）指大数据中巨大的数据量和数据的完整性。在过去，由于技术手段和成本的限制，我们无法保存生产中所有的数据，而是通过采样，转换的方法保存和分析相对重要的数据，通常原始状态的数据难以被保存下来。随着大数据技术的到来，保存数据的成本也越来越低廉，这使得我们可以保存过去难以想象的巨量的数据，并可以对原始的数据进行全面高效的分析。

种类多（Variety）指多种多样的数据类型，并且在海量的，多种类的数据源和内容之间发现内在的相互关联。传统的数据类型为结构化的数据，数据以一条条记录的结构化的形式存在关系型的数据库中。而大数据使得我们可以存储和处理非结构化的数据，可以更加灵活地应对海量的、不同种类的数据源。

速度快（Velocity）指大数据可以满足实时性的需求，一方面指信息的获取的速度，另一方面指信息的处理速度。大数据技术能够应对处理海量的数据交换和数据处理，从而实现快速低延时，高时效的应用需求。

价值密度低（Value）指在海量的信息中，信息需要通过分析和挖掘来获得其巨大的内在价值。相比传统数据中通过筛选采样的到重要的信息，大数据中更注重对原始的完整数据直接的内在关联的分析和挖掘。

### 大数据的处理流程

在对大数据技术的研究中，刘智慧和张泉灵[42]总结并提出了大数据的基本处理流程，其处理流程可分为四个阶段，分别为：数据采集，数据处理与集成，数据分析和数据解释。

数据的采集是大数据处理流程中最基础的一部分，包括对传统的结构化的数据，例如关系型数据库的数据；半结构型数据，例如HTML；以及非结构化的数据，例如文档内容等数据的采集和整合。

数据处理与集成是对已经收集到的数据进行数据清洗，去噪，从而得到可信赖的数据并进行进一步的集成和存储。常用数据清洗和去噪的方法是通过聚类分析，关联分析设置一系列的数据过滤器，将不相关的无用的数据过滤掉，以防他们对整体的数据产生负面的影响。

数据分析是指通过统计分析，机器学习，智能算法等方法，对预处理和集成过的信息进行进一步的加工和分析，挖掘数据中内在的关联性和潜在的价值，从而使得数据可以提供决策支持，信息预测，商业智能等应用，是大数据处理流程中最为核心的部分。

数据解释是将数据分析得到的结果进行进一步的可视化的展示。随着数据量的增大和数据分析的复杂性增强，传统的数据显示方法已经无法满足分析结果的显式的需求。通过大数据可视化技术[43]以及人机交互技术，可以将数据分析得到的结果简介、形象、生动地展示给用户。

### 大数据的关键技术

大数据的核心是对海量数据的分析和处理能力，刘智慧和张泉灵[42]在对大数据研究的综述中总结了大数据的关键技术，包括：云计算，MapReduce计算框架，分布式文件系统，分布式并行数据库，Hadoop开源大数据平台以及数据的可视化。在这里，我们将主要介绍应用在机票个性化推荐系统中的大数据技术，包括Hadoop以及近年来非常盛行的Spark分布式计算系统。

1. **Hadoop**

Hadoop[10]是有Apache基金会所开发的一个开源的分布式系统平台。用户在该平台上，可以在不需要了解底层分布式系统的实现细节的情况下开发和使用分布式程序进行大型的并行计算。

在Hadoop框架中，最为核心的两个部分为：Hadoop分布式文件管理系统（Hadoop Distributed File System, HDFS）和MapReduce分布式计算框架。HDFS作为一个分布式的文件管理系统，具有很高的容错性，可以用来部署在廉价的硬件上，同时它具有很高的吞吐量，可以为大规模的服务器集群提供快速的数据读写访问。MapReduce作为计算框架可以很好地和HDFS结合，能够针对存储在HDFS上的一个很大的数据集的任务分解成并行地运行在集群多个节点中的运算。

随着Hadoop的不断完善，在Hadoop周围也发展出了许多子项目，并逐渐发展成了一个完善的大数据处理平台的生态系统。根据Hadoop生态系统中的组件，Intel给出了Hadoop商业版大数据平台的实现，图2-1为Intel Hadoop发行版大数据平台[44]的组件的结构。

在该系统中，底层为HDFS作为大数据分布式文件系统，提供稳定、高容错的分布式的文件存储；HBase[45]作为分布式的数据库，提供兼容非结构化数据的高效并行的海量数据的读写；MapReduce作为分布式计算框架发，提供业务作业的并行计算能力；上层的Hive [46]作为分布式数据仓库引擎，提供了基于分布式文件系统和分布式数据库的兼容SQL语言的增、删、改、查的统计分析的操作；Pig Latin[47]作为数据流处理语言，提供了过程式语言数据流的特点；Mahout[48]作为基于分布式数据源的数据挖掘的工具，提供了丰富的分布式机器学习和数据挖掘的算法；右侧的Zookeeper[49]作为分布式系统的可靠协调系统，提供分布式系统中配置维护、名字服务、分布式同步、组服务等可靠服务；左侧Sqoop[50]是一个用来将数据在Hadoop和传统的关系型数据库之间相互转移的工具；Flume[51]作为日志收集系统，提供高效、稳定的日志数据的收集服务。

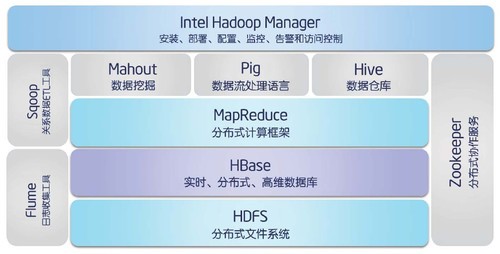


图2-1 Intel Hadoop发行版大数据平台组件架构

Figure 2-1 Architecture of Intel Hadoop Big Data Platform

1. **Spark**

Spark [52]是UC Berkeley AMP实验室所开发的类似于Hadoop MapReduce的开源通用的并行计算框架。与Hadoop MapReduce不同的是，Spark中使用启用了内存分布数据集，可以将计算作业输出的中间结果保存在内存中，而不再需要每次都重新写入HDFS，因此能在某些工作负载方面，特别对于在一些机器学习的迭代算法中表现更加优秀。此外，Spark中提出了弹性分布式数据集 （RDD）的抽象概念，并且提供了除Map和Reduce外更多、更为丰富的算子，用户可以通过对抽象的RDD直接进行算子的操作来对数据集进行高效的并行运算。

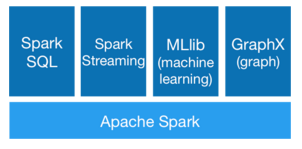


图2-2 Apache Spark组件结构

Figure 2-2 Architecture of Spark components

图2-2为Spark生态系统中各个组件的结构图，可以看到，基于Spark的基础上又发展出了支持SQL语句数据库查询引擎Spark SQL；面向流数据的Spark Streaming；基于Spark的分布式机器学习算法的工具开发库MLlib；以及基于Spark的分布式图形计算的工具开发库GraphX。同时，Spark还可以很好地和新版Hadoop中的资源管理器YARN以及分布式文件系统HDFS结合，分别作为Spark大数据平台中的资源管理器和分布式文件系统。

### 大数据技术在在线旅游系统中的应用

随着大数据技术的迅猛发展，大数据开始逐渐应用到包括商业，金融，交通，能源等各个行业。旅游业作为第三产业，在经济的发展中起到了巨大的支撑作用。随着旅游行业的信息化，智能化的发展，大数据技术的应用将能够帮助我们更好地挖掘旅游信息中的数据价值。

罗成奎[53]研究分析了大数据技术在智慧旅游中的应用，并将智慧旅游总结归纳为旅游服务，旅游管理以及旅游营销三个主要的应用方面。

旅游管理的应用是从企业内部管理系统入手，通过大数据平台增强对内部数据的存储和管理能力，进而通过信息智能化的手段优化旅游管理企业内部的管理流程。

旅游营销的应用是指通过大数据平台对在线旅游系统中的用户数据以及互联网上的社区信息，社交网络等信息对用户的出行需求进行深度的分析和挖掘，从而实现旅游行业的精准营销。

旅游服务的应用是指通过依托大数据平台，将海量分散的旅游信息进行整合，并通过分析和利用旅行者的数据，建立合适的分析模型，为用户提供更加高效，更加优质的个性化智能旅游服务。例如，携程旅行网[2]在官网上提供了“猜你喜欢”的板块，该功能是基于大数据平台通过对上亿用户的用户数据进行深度的挖掘分析来实现个性化的旅游产品的推荐。本文研究的机票个性化推荐系统也是属于大数据技术在旅游服务智能化上的一个应用。

## 本章小结

本章主要介绍了本文研究中的相关技术理论基础，首先分析和总结了当前研究中隐式反馈推荐系统的关键问题以及面向隐式反馈的推荐技术。接着还介绍了大数据的基本概念以及当前主流的大数据技术的基本情况，同时还介绍了大数据技术在在线旅游系统中的应用情况。

# 基于用户偏好模型的机票个性化推荐算法

本章通过分析机票的数据特征，提出了一种基于用户偏好模型的机票个性化推荐算法，对用户在某条航线上的搜索结果进行推荐。同时提出了一种基于航线的协同过滤的算法进行用户在不同航线上偏好的跨航线的学习。

## 机票的数据特征

相对于电影，书籍等具有较为固定属性的静态商品，机票作为一种受时间影响的价格敏感的动态商品，其价格具有波动性和敏感性的特征，具体表现在：

1. 价格的波动性：同一航班同一舱位的机票在距离起飞时刻的不同时间有着较大的价格波动，这种价格与航空公司的实时定价策略有关，较难预测。

2. 价格的敏感性：当用户选择搜索机票时，价格将是用户非常重要的决策因素，用户总是会倾向于选择满足其需求的价格低的机票。

由于机票价格的波动性和敏感性，我们很难将某一个航班舱位下的机票当做是一个固定的商品。与此同时，机票还具有一些其他显性的特征属性，每一个属性都有可能影响用户的购买决策。



图3-1上海-北京航线的部分搜索结果页

Figure 3-1 Sample of search result page in airline Shanghai to Beijing

图3-1显示了于2015年12月02号在携程旅行网[2]上查询的2016年01月02号的上海到北京这条航线上的部分搜索结果。可以看到在该次的搜索结果中一共有97条航班信息，航班信息中包含了航空公司，起飞到达时间，起飞到达机场等信息。其中，每个航班中又包含了若干张不同的机票，对应着不同的舱位，价格，以及退改签政策信息。

在本文中，我们使用了由国内某大型在线旅行网提供的用户国内航线直飞的历史订单数据。由于国内航线大多数机票订单为直飞，因此本文主要研究直飞航线上的机票个性化推荐问题。通过对用户历史订单数据的分析，对于一条给定的航线，我们抽取了若干显性的机票特征属性，如下表所示：

表 3-1 机票的特征属性

Table 3-1 Features of Flight Tickets

|  |  |
| --- | --- |
| **特征** | **描述** |
| 出发城市（dcity） | 航线的出发城市 |
| 到达城市（acity） | 航线的到达城市 |
| 价格（price） | 机票的票面价格 |
| 起飞时间（take-off time） | 航班的起飞时间 |
| 航空公司（airline） | 航班的航空公司 |
| 舱位等级（class） | 机票的舱位等级（分为F,C,Y三等） |
| 起飞机场（dport） | 航班的起飞机场（有些城市可能有多个机场） |
| 到达机场（aport） | 航班的到达机场 |
| 退改签政策（policy） | 机票的退改签政策 |
| 机型大小（size） | 航班的机型大小 |

## 用户偏好模型及推荐算法

### 基于信息熵的用户偏好分析

不同的用户在选择机票时有着不同的偏好。有的用户可能是价格敏感的用户，总是选择最低价；有的用户可能是某个航空公司的会员，选择机票时在合理的价格内可能更倾向某个航空公司从而获得更多的里程积分；有的用户出行时偏好选择离家近的机场。

在进行用户偏好建模前，我们选取了上海到北京这条航线上的历史订单数据，并对其中的活跃用户不同的偏好进行了分析。这里的活跃用户定义为用户过去两年上海到北京这条航线的历史订单数大于等于四单的用户，因为只有当用户有了一定的历史订单后，我们才能分析用户对某个特征的偏好情况。

在进行偏好分析中，我们引入了信息熵的概念。信息熵通常是用来描述信息源的不确定度，其定义为：

其中*x*表示某一个信源，而表示这个信源的某一种可能的取值，表示*x*取值为的概率。信息熵H(*x*)越大，则表示*x*的信息量越大，不确定性越高，相反H(*x*)越小，则*x*的信息量越小，确定性越高。在决策树ID-3[54]中，信息熵也常用来表示对于某个特征的决策的不确定度。

和决策树类似，在这里我们用信息熵来表示用户在选择机票时，对某一个特征选择的不确定度，其中*x*表示机票的某一个特征，则表示机票这个特征的某一种选择。我们通过分析历史订单来计算用户在选择机票时各个特征的信息熵，从而来确定用户某些特征是否表现出强的偏好。

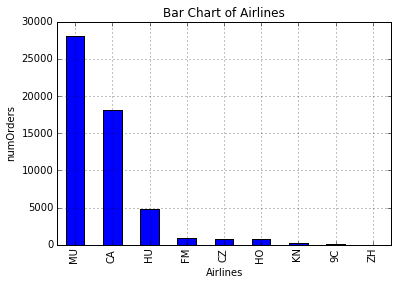


图3-2 【SHA-BJS】航线的航司直方图

Figure 3-2 Bar Chart of Airline Company

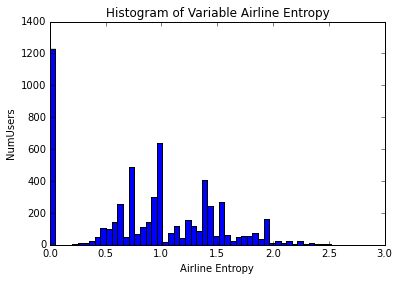


图3-3【SHA-BJS】航司信息熵的分布图

Figure 3-3 Bar Chart of Airline Entropy

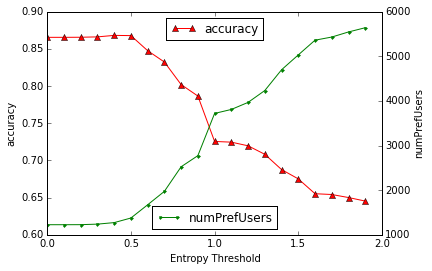


图3-4【SHA-BJS】航司信息熵的偏好阈值与准确率以及偏好用户数量的变化图

Figure 3-4 Prediction Accuracy and # of Targeted Users @ Entropy Threshold

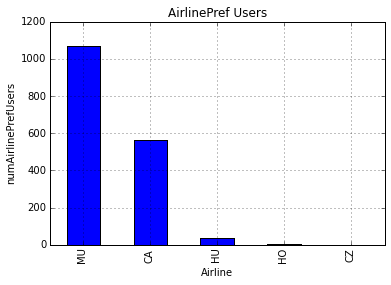


图 3-5【SHA-BJS】航司偏好的分布图

Figure 3-5 Bar Chart of Airline Preference

图3-2为上海到北京航线的航司直方图，横轴为不同的航空公司，纵轴为用户历史订单的数量。可以看出，上海到北京航线上的订单主要来自东航（MU）,国航（CA）以及海航（HU）。

根据用户的历史订单，我们可以根据式（3-1）计算出活跃用户对航司这个特征选择的信息熵。图3-3为上海到北京航线的航司分布图，其中横轴为航司信息熵的值，纵轴为航司信息熵值对应的活跃用户的数量。从该图中可看出，有约1200个活跃用户的航司信息熵值为0，意味着这些用户在历史订单中总是选择某一个特定的航空公司。根据航司信息熵的分布图，我们可以设定一个阈值，当用户的航司信息熵的值小于这个阈值时，我们标记这类用户为对航司这个特征具有强偏好的用户，对于这类用户我们可以为其进行机票搜索预订时推荐具有该特征属性的机票。

图3-4 为上海到北京这条航线上，航司信息熵的偏好阈值与准确率以及偏好用户数量的变化图，其中横轴为划分用户在航司属性上是否有强偏好的信息熵的阈值，右纵轴是该阈值对应的被标记为航司偏好的用户的数量，而左纵轴对应的准确率指的是，对于这类被标记为有偏好的用户中，新预订的一个订单中航司为用户所偏好的航司的用户数量的百分比。从图中可以看出，航司信息熵阈值小于等于0.5时，可以保持一个较高的准确率。当阈值不断增大时，被标记为航司偏好的用户增加，而准确率则逐渐减小。当阈值等于0.5时，可以在保持较高准确率的条件下，获得较多的被标记用户数。

图3-5 展示的是航司信息熵阈值为0.5时，被标记为航司偏好用户所偏好的航司的分布图。可以看到，大多数用户的偏好来自东航（MU）,国航（CA）以及海航（HU）。

图3-2—3-5展示了对上海到北京这条航线上活跃用户的航司偏好的分析，可以看出信息熵能够较好的表示出用户对航司特征的偏好情况，并做出准确的预测。用同样的方法，我们对舱位等级，起飞时间段，价格区间三个特征做了偏好分析，并根据各个特征的信息熵的偏好阈值与准确率以及偏好用户数量的变化图为相应的用户打上偏好的标记标签。

图3-6是上海到北京这条航线上活跃用户的偏好标签数量的分布图，其中横轴表示每个用户被标记了特征偏好的数量，这些特征偏好包括，航司偏好，舱位偏好，起飞时间段的偏好以及价格区间的偏好，而纵轴表示对应的活跃用户的数量。可以看出，大部分活跃用户在航线上有不少于一个的特征偏好，还有少部分用户在所有四个特征属性中都有偏好。

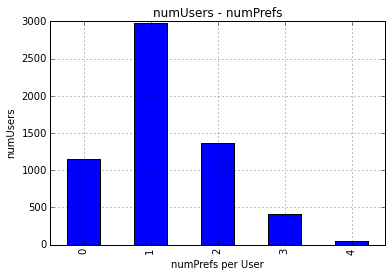


图 3-6 【SHA-BJS】用户偏好标签数量的分布图

Figure 3-6 【SHA-BJS】Bar chart of the number of preferences per user

### 基于信息熵的用户偏好模型

在用户的偏好分析中，我们通过分析特征的信息熵给一部分用户打上偏好的标签，对于这部分用户上的偏好特征可以获得比较好的预测的准确率。因此我们可以看出，信息熵能够较好的反应用户对机票各个特征选择的一个偏好程度。

在此，我们提出了一种基于信息熵的用户偏好模型。该偏好模型类似于Content-Based KNN的推荐模型，通过计算待选机票与用户购买的历史机票的相似度作为用户对这张待选机票的偏好得分，相似度越高则得分越高。在Content-Based KNN中，我们需要定义一个相似度的函数用来计算物品之间的相似度，该相似度函数通常如式（3-2）所示

其中*i*和*j*为两个不同的物品，*F*为物品的特征属性的集合，是物品*i*和*j*对应的特征*f*的值，为相似度在特征*f*上的权重。

在确定相似度函数时，我们需要首先量化物品的各个特征的值，并根据经验或者实验确定出各个特征在相似度函数上的权重，这些权重通常是对于所有用户的一个普适性的权重。然而在用户选择机票的过程中，不同的用户对机票不同的特征属性有着不同权重的偏好，因此这种普适性的相似度函数并不能很好地表现出不同用户的不同特征偏好。

在之前的用户偏好分析中，我们发现信息熵可以很好的表示出不同用户对于不同特征的偏好情况，因此，我们可以使用不同用户对于机票不同特征的信息熵来量化相似度函数中不同特征的权重。这里，我们定义相似度函数中用户在不同的机票特征上的偏好权重为：

其中为的不同取值，为对应的概率，表示特征*f*经过标准化后的信息熵的值。从该式可以看出，对于用户在历史订单中对于特征*f*选择的确定性越强，其特征的信息熵值越小，用户对特征*f*的偏好权重越大。

对于机票订单的特征属性，除了价格和起飞时间为连续的变量外，其余的特征变量都为离散变量。为此，我们为价格和起飞时间做了变量离散化的处理。在处理价格时，我们还引入了一个价格敏感指数（Pi）的概念。因为当用户在不同的时刻搜索查询机票时，同样的一个价格代表着不同的价格敏感程度。例如当票量充裕时，航空公司会采取低价的折扣，1000元在搜索的机票结果中可能是一个比较高的价格，而当票量紧俏时，航空公司会重新调整价格策略，这时1000元在搜索结果中可能是一个低价。为了消除机票票面的绝对价格在不同时间的订单对用户价格敏感特征的影响，我们定义了如下的价格敏感指数：

其中是这个航班对应的经济舱的标准价格（全价），*P*是机票的当前价格，是当前搜索结果页中的最低价格。从该式可以看出，当用户选择的票价越接近当前的最低价，Pi越接近1，表示用户对价格越敏感；反之，当用户选择的票价越高，Pi越小，表示用户对价格越不敏感。

在对价格敏感指数以及起飞时间进行离散化处理后，我们可以为这些特征定义其离散变量的相似函数。 对于式（3­-2）中的两张机票*i*, *j*在特征*f*上的相似度，我们有如下定义：

其中为机票*i*和*j*在特征*f*上的离散值。

在定义了机票的相似度函数（3-2）后，我们可以用KNN的方法，根据用户航线上的历史订单为用户搜索机票时进行个性化的推荐。算法3-1为基于信息熵的KNN的机票个性化推荐算法。

从算法3-1可知，用KNN的方法进行个性化推荐的计算时，在计算搜索结果中每一机票s的得分时都要计算*s*与历史订单中的相似度，因此其时间的复杂度为，其中n为搜索结果页中机票的数量，m为用户历史订单的数量。mlogK 和 nlogN为计算Top K和Top N时排序的复杂度，如果我们使用两个大小为K和N的最小堆来计算Top K 和 Top N，可以把复杂度优化为， 因为K和N为常数，复杂度可化简为。

|  |
| --- |
| **Algorithm 3-1:**  **Entropy-based KNN Top N Recommendation Algorithm** |
| **Input： –** Ticket Set of Historical Flight Order of User u  **S** – Tickets Set in search result page  **F** – Set of Flight Ticket Features  K – number of neighbours  N – number of recommendation |
| **Output：R** – Set of Top N Recommended Flight Tickets in **S** |
| 1. **Calculate the entropy for each feature in F by :**   For in **F**:    End For   1. **Calculate the weight for each feature preference:**   For in **F**:    End For   1. **Calculate the score of each ticket i in S for recommendation**   Define the score array as arr\_score  For s in **S**:  Initialize the similarity array: arr\_sim  For t in :  End For  arr\_sim.rank(ascending = False)    End For   1. **Return tickets with Top N scores in arr\_score as recommendation** |

在机票的历史订单数据集中，用户的在一条航线上的历史订单的数量m通常是一个比较小的值。 因此，机票s与K最邻近的订单的平均相似度可以近似为s与历史订单集中的所有订单的平均相似度。

式（3-7）中我们将相似度函数中具有n个可能取值的离散的特征变量转换为模为1的n维的向量。例如离散变量*x*可能的取值有为0，1，2，3，且*x*=1，则对应的离散变量的向量为(0,1,0,0)。在离散变量转换成对应的向量后，我们可以得到等式。

根据式（3-7）我们可以推导得到式（3-8）。从式中我们可以看出，在计算一个用户对机票s的得分时，我们可以提前根据式（3-9）为每个用户计算出用户在各个特征上的偏好向量，而在实时地进行推荐时，我们的计算的复杂度可以优化为O(n)。

因此我们可以将算法分拆成离线的用户偏好模型的计算以及在线的推荐计算两个部分。算法3-2离线计算用户偏好模型，而算法3-3为在线计算搜索结果页中机票的得分，并为用户进行Top N的推荐。

|  |
| --- |
| **Algorithm 3-2:**  **Offline User Preference Calculation (UserPref)** |
| **Input： –** Ticket Set of Historical Flight Order of User u  **F** – Set of Flight Ticket Features |
| **Output：p** – Set of User Preference Vector for each feature  **w** – Set of weight for each feature |
| 1. **Calculate the entropy for each feature in F by :**   For in **F**:    End For   1. **Calculate the weight for each feature preference:**   For in **F**:    End For   1. **Calculate the preference vector for each feature :**   For in **F**:      End For   1. **Return preference vector and weight for each feature** |

|  |
| --- |
| **Algorithm 3-3:**  **Online Top N Recommendation** |
| **Input： p –** Set of User Preference Vector for each feature f  **w** – Set of weight for each feature f  **F** – Set of Flight Ticket Features  **S** – Tickets Set in search result page  N – number of recommendation |
| **Output：R** – Set of Top N Recommended Flight Tickets in **S** |
| 1. **Calculate the score of each ticket i in S:**   Define the score array as arr\_score  For s in **S**:    End For   1. **Return tickets with Top N scores in arr\_score as recommendation** |

## 结合协同过滤的跨航线偏好学习

在上一节中，我们提出了基于信息熵的用户偏好模型，该模型基于用户的历史订单数据来计算出用户对机票各个特征的偏好向量及偏好权重，以此来为用户进行个性化的搜索和推荐。然而，不同航线有着不同的数据分布，例如，图3-7显示了上海到北京和上海到广东两条航线上航空公司的分布图，其中横轴为不同的航空公司，纵轴为2013及2014年两年的订单数量。从图中可以看出，在上海到北京和上海到广东这两条航线上，航空公司的分布具有较大的差异。另一方面，用户在不同的航线上也会有着不同的偏好，例如有的用户经常去某些城市出差，在某些航线上出行偏商务，会预订比较舒适，甚至是头等舱，商务舱的机票，而当用户自己出行去其他的城市时，则会选择相对经济实惠的机票。因此，我们无法将用户在不同航线上的历史订单记录简单的放在一起来为用户计算一个单一的偏好模型，并应用到不同的航线，用户不同航线上的订单数据将对应用户在不同航线上的偏好模型。

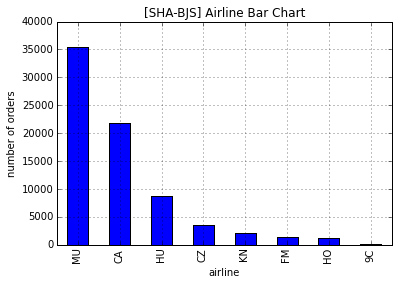
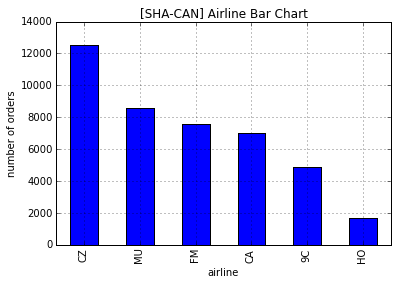
 

图 3-7 航线【上海到北京】以及【上海到广东】的航司分布图

Figure 3-7 Bar Chart of airline companies in airline SHA-BJS and SHA-CAN

当用户在某些航线上订单过少时（我们称这样的航线为用户的非活跃航线），其对应的偏好模型可能并不能很好的反应出该用户在该航线上的一个整体的偏好。为了解决用户在一些非活跃航线中历史订单稀疏的推荐问题，我们使用了协同过滤的技术，提出了一种基于航线的协同过滤（Airline-based Collaborative Filtering）的方法，通过学习用户在其他相似航线上的历史订单，来加强用户在该航线上的偏好的学习。

和Item-based的协同过滤算法相似，我们把航线当做Item，通过定义一个航线的相似度的函数来计算航线间的相似度，并找出目标航线的相似航线。然而不同的是，航线并不是我们要直接推荐的对象，我们通过找出与用户目标航线相似的航线，并将这些航线上的历史订单迁移应用到我们的目标航线来计算用户在目标航线的偏好，从而来加强偏好模型对用户在该目标航线上的推荐效果。

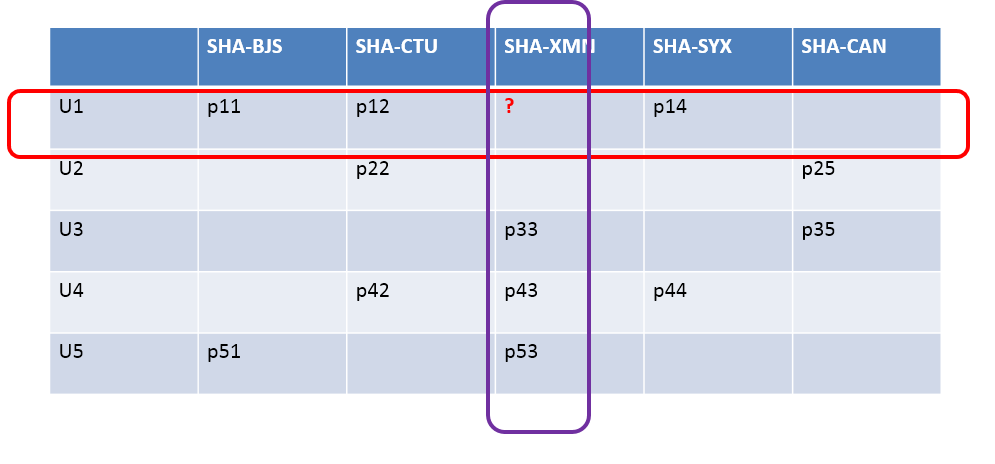


图 3-8 基于航线的用户偏好协同过滤的解释图

Figure 3-8 Illustration of Airline-based Collaborative filtering Algorithm

图3-8作为一个例子阐释了我们基于航线的协同过滤方法的基本思路。图中包含了不同的五个用户和五条航线，其中标有p的指的是用户在该航线上根据历史订单得到的偏好，为空的则表示该航线对于该用户是陌生的（无历史订单）或者非活跃的（历史订单较少），。在图中，上海到厦门的航线（SHA-XMN）对于用户U1来说是一条陌生的航线，或者说是非活跃的航线我们可以通过分析所有用户在这条航线上的历史订单情况，以及用户在其他航线上的历史订单情况，找出用户U1与该条目标航线（SHA-XMN）相似的其他活跃航线（历史订单较多），以此来估计用户在目标的非活跃航线上的偏好。

在这里，我们通过分析不同航线上用户的订单在各个机票特征的分布情况，定义了一个航线的相似度，如式（3-10）所示。

其中，*a*，*b*为两条不同的航线，为这两条航线所有用户的订单在特征f上的一个离散分布，我们利用之前的方法将离散的分布转换成标准化后的模为1的向量**，**并用cos余弦函数来计算这两个向量的相似度，如式（3-11）所示。

定义了相似度函数之后，我们就可以将用户在与目标相似的其他航线上的偏好通过协同过滤的方法加强用户在目标航线上的偏好，由于不同航线上特征分布的不同，我们在利用其它航线上的偏好时，应该对其进行去分布的处理，并将其转化成服从目标航线的特征分布的偏好。这里，如果我们将不同的航线当做是不同的域（Domain），我们会发现Airline-based 的协同过滤与Instance-based Transfer[55-57]的迁移学习的思想非常类似，该迁移学习方法的思想主要是通过迁移具有不同数据分布的其他领域上的数据，来加强目标领域上的学习效果。

式3-12中，A为用户历史订单中所有的航线的集合，为我们要进行推荐的目标航线，为用户在航线*a*上的历史订单数量，为用户在航线*a*，特征*f*上去分布处理后的偏好向量。可以看出，这里我们通过加权航线相似度以及用户在航线上历史订单数量（）来得到用户在相似航线上的一个总体偏好。

在对偏好向量进行标准化之后，我们可以通过式（3-13）来将相似航线上的偏好整合到目标航线。

其中，为用户在目标航线上特征*f*的偏好向量（若为完全陌生的航线则该偏好向量为零向量），而参数为一个hyper-parameter，我们将其称为迁移比率（transfer rate），用来调整相似航线偏好占整体偏好的比例。

算法3-4展示了上述我们提出的基于航线的协同过滤的用户偏好模型算法，相比较上一节的计算用户偏好的算法（算法3-2），通过Airline-based协同过滤的方法，使用其他相似航线上的用户偏好来加强目标航线偏好的学习，从而解决了用户在非活跃航线上的数据稀疏的问题。和算法3-2类似，该算法也可作为离线的计算单元提前计算出用户的偏好，并结合算法3-3为用户在线搜索机票时进行个性化的推荐。

|  |
| --- |
| **Algorithm 3-4:**  **Airline-based Collaborative Filtering for User Preference** |
| **Input： –** Ticket Set of Historical Flight Order of User u  – target airline  **F** – Set of Flight Ticket Features  **DIST\_AIRLINES** – Set of feature distribution in airlines  – Hyper-parameter |
| **Output：p** – Set of User Preference Vector for each feature in target airline  **w** – Set of weight for each feature in target airline |
| 1. **Calculate the preference vector in different airlines:**   Define **A** as the airline set for user historical flight orders  For in **A**:  define as the flight orders of airline in  define  For in F:    End For  End For   1. **Adjust user preferences of other airlines by DIST\_AIRLINES:**   define := **DIST\_AIRLINES**[]  For in **A**\:  Set := DIST\_AIRLINES[]  For in **F**:    End For  End For   1. **Transfer the preferences of other airlines in target airline:**   For in **F**:        End For   1. **Calculate the entropy for each feature in F:**   For in **F**:    End For   1. **Calculate the weight for each feature preference:**   For in **F**:    End For   1. **Return preference vector and weight for each feature** |

## 相关实验与结果分析

### 实验数据集

本文使用了国内某大型在线旅行网在2013年至2014年的国内机票订单数据作为我们实验的数据集。该数据集包含了1600万个用户在不同航线上的9400万条订单数据。由于机票个性化推荐的主要应用在于热门的航线上，即当用户在搜索航线上的机票时，搜索引擎会返回数量较多的结果，这时推荐系统可以根据学习得到的用户偏好从众多的搜索结果中为用户做出较为精准的推荐。而对于一些冷门的航线，用户可能并没有太多的选择，因此也不存在信息过载的问题。

这里我们抽取了4条热门航线作为测试航线，包括上海到北京（SHA-BJS），上海到广州（SHA-CAN），上海到厦门（SHA-XMN），上海到成都（SHA-CTU）。图3-9为这四条航线的订单数量的直方图，其中纵轴表示在2013到2014年内用户在对应航线上订单的总数量。从图中我们可以看出，上海到北京航线上拥有最多的订单量，超过了7万张，上海到广州航线和上海到成都航线分别约为4万张和3万张，上海到厦门航线最少，不到2万张。另外，我们选取的这四条热门航线还具有一定的出行目的航线属性的代表性。其中，上海到北京的航线属于偏商旅的航线，而上海到厦门则属于偏旅游的航线，而上海到广州，上海到成都的航线则兼有了商旅和旅游的用户出行属性。另外，为了防止一些代理商的用户订单干扰个性化的推荐效果，我们将两年内航线上订单数量大于10的用户从实验数据集中剔除。

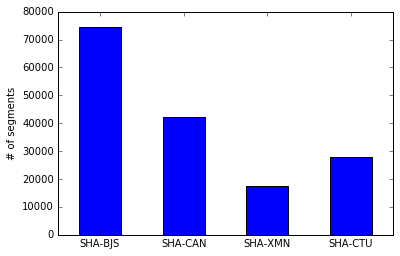


图 3-9 测试航线的订单数量直方图

Figure 3-9 Bar Chart of the number of order segments in test airlines

为了能够以更加真实的情况来测试推荐算法的性能，我们将用户订单的数据集按照时间的先后划分为训练集和测试集两个部分。其中，2013年1月至2014年11月的用户订单数据作为训练集，2014年12月的用户订单数据作为测试集。

在对于测试集进行测试和评估时，我们不仅需要用户的订单数据，还需要知道用户在选择预订该机票订单时所看到的其他未选择的机票，即用户在当时搜索机票时所看到的机票搜索列表页上的机票。然而，我们并没有这一部分的数据。这里，我们通过找出用户下单时刻前后一段时间区间内的其他用户的订单，并去掉不同用户预订的相同的机票订单，来模拟用户在预订机票时所看到的可供选择的机票，即机票搜索结果。通过这样的模拟方式会带来不可避免的系统误差，一种情况是当某个时段用户的预订量较少时，我们只能模拟还原出一小部分的搜索结果；另一种情况是，机票的价格在一定的时间区间内也会发生变动。

对于热门航线来说，用户的预订量通常都比较大，同时，机票价格在短时间内变动的机率相对较小。因此对于热门航线，我们可以通过上述的方式，较好地还原出用户预订机票时所看到的搜索结果。

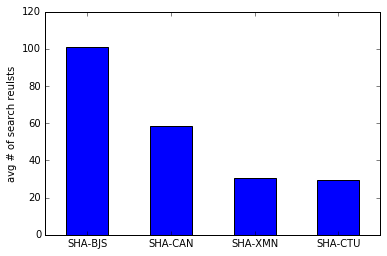


图 3-10 测试航线的平均模拟搜索结果数量直方图

Figure 3-10 Average number of search results in different airlines

图3-10为测试集中不同测试航线上平均的模拟搜索结果数量的直方图。从图中可知，在上海到北京这条最为热门的航线上，用户在测试订单预订时，通过其他用户的订单数据模拟得到的搜索结果页的平均机票数量约为100条，在上海到广州航线上约为60条，上海到厦门和上海到成都航线约为30条。

### 评价指标

为了测试和评估不同个性化推荐算法的准确性，我们需要针对机票个性化推荐问题制定一个有效的评价指标。在第二章中，我们已经总结介绍了一些常用的隐式推荐算法的评价指标。这里，我们将采用其中的平均准确率均值的方法（MAP）来对基于用户偏好模型的机票个性化推荐算法进行评估。

MAP是基于排序的评价指标，它通过计算用户在推荐列表中偏好对象的平均准确率（见式（2-1）），来作为评估推荐结果的准确率，能够很好的反应出用户偏好物品之间的排序的关系。

与传统的推荐问题不同，机票个性化推荐问题很强的实时性，用户通常每次搜索只会在搜索结果页面中选择一张机票进行机票的预订和购买。在基于用户偏好的机票个性化推荐算法中，我们通过用户偏好来计算用户对实时的搜索结果中各个机票的个性化的偏好得分，得分越高表示偏好越强。因此，根据计算出的偏好得分，我们可以为搜索结果中的各个机票进行偏好得分从高到低的降序排序。我们定义为机票i在排序中的序号，则我们可以定义推荐算法对于用户在这次选择中推荐效果的准确率为：

进一步，我们可以得到对于测试集中所有用户订单的平均准确率均值：

其中，表示测试订单集合，表示测试订单集合中订单的数量。

同样，MAP值越大，则表示用户在预订计票时偏好的机票排序越靠前，推荐效果则越好。当MAP小于50%时，则推荐算法优于随机的推荐。

另外，我们根据机票个性化推荐的问题，还设计了Top N准确率的评价指标。Top N准确率指在所有测试订单中，根据用户的偏好在用户预订机票时的搜索结果页中为用户推荐偏好最强Top N个机票是否命中的百分比，如式（3-16）所示。

### 相关的比较算法

在下述实验中，我们根据当前隐式推荐系统的一些主要方法，设计并实现了一组可以应用在机票个性化推荐的问题中的推荐算法，用来作为我们基于用户偏好模型的机票个性化推荐算法（UserPref）的评估和比较：

1. 基于CBR的机票个性化推荐算法（CBR）

在第一章中，我们简单介绍了Coyle[9]提出的基于CBR的个性化推荐算法。这里，我们根据我们上文抽取的机票的数据特征，设计实现了其中效果较好的Session-based的推荐方法来作为我们模型的实验参照的一种推荐方法。

1. 单类协同过滤的方法（OCCF）

在第二章中，我们介绍了基于隐式反馈信息的单类协同过滤的方法[32]，然而上文中我们曾介绍到，机票是受时间变化的具有价格波动性和价格敏感性的一种动态商品，因此我们无法给机票建立一个固定的物品或者对象的集合，因此无法直接使用协同过滤的方法来根据用户的历史订单为用户进行机票的推荐。为了能够使用协同过滤的技术，我们枚举了所有机票的离散属性的组合，并将每一种组合作为一个对象从而组成一个数量巨大的对象的集合，并基于这些组合对象应用单类协同过滤的方法为用户进行个性化的机票推荐。

1. 低价排序（Price-Rank）

在用户预订机票时，低价排序作为最常用的一种排序方法能够帮助用户很好的根据机票价格进行机票的筛选。然而，并不是所有的用户都倾向于最低的价格。这里，我们将机票搜索结果页中低价排序得到的推荐结果作为机票个性化推荐的一个基准（Baseline）。

### 实验结果与分析

利用上边所述的测试数据集，即2014年12月份的国内机票在测试航线上的订单数据，我们对基于用户偏好模型的机票个性化算法以及上述的其他用来比较的推荐算进行了测验和评估。

图3-11及对应的表3-2展示了在不同测试航线上个性化推荐算法的平均准确率均值（MAP）的比较。从图表中我们可以看出，我们的基于用户偏好模型的机票推荐算法在四条航线中都具有最低的MAP，即最好的推荐效果。基于CBR的机票推荐算法也具有较好的推荐效果，其在各个测试航线的MAP值略高于基于用户偏好模型的机票推荐算法。单类协同过滤的方法（OCCF）的推荐效果相对较差，其原因可能在于将推荐对象按照机票的动态特征属性的组合来进行建模不能够很好地还原机票以及航班之间本身的内在特征和联系，而且组合后的对象集合数量较大，存在数据稀疏的问题。最后，按照基于低价排序的策略的推荐效果仅仅略优于随机的推荐策略（MAP = 50%），这也说明了有很大一部分用户并不倾向于的价格最低的机票。

图 3-11 不同测试航线上个性化推荐算法的MAP比较

Figure 3-11 MAP of Personalized Flight Recommendation Algorithm in test airlines

表3-2不同航线上个性化推荐算法的MAP比较

Table 3-2 MAP of Personalized Flight Recommendation Algorithm in test airlines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **MAP (%)** | | | |
| **UserPref** | **CBR** | **OCCF** | **Price-Rank** |
| **SHA-BJS** | 26.61 | 29.83 | 41.05 | 48.51 |
| **SHA-CAN** | 29.69 | 32.56 | 42.36 | 46.49 |
| **SHA-XMN** | 28.05 | 31.94 | 41.78 | 50.07 |
| **SHA-CTU** | 30.56 | 33.28 | 43.25 | 45.52 |

为了能够进一步地测试和评估我们基于用户偏好模型的机票推荐算法，我们选取了最为热门的上海到北京航线进行了更加深入的实验和分析。图3-12展示了上海到北京航线测试订单的排序百分位的分布直方图，这里的排序百分位实际上就是式（3-14）中用来计算MAP的准确率（Precision），该值越小，则表示用户实际预订的机票在个性化的排序中越靠前，则推荐效果越好。从图3-12中我们可以看出，其测试订单上的排序百分位的分布呈现一个长尾分布形态，大多集中在较低位置，且在靠近1%处有一个较为集中的峰值。因此我们可以得知，推荐算法对于大多数的测试订单具有较好的推荐效果，但对于小部分的订单也存在推荐效果较差的情况。

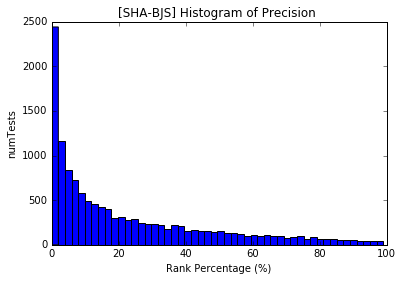


图 3-12 上海到北京航线测试订单的排序百分位的分布直方图

Figure 3-12 Number of test instances in different testing rank percentage

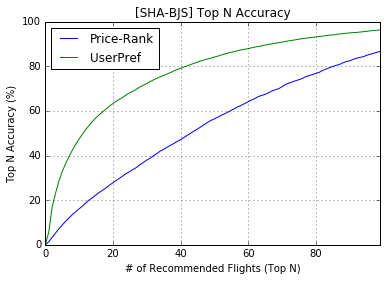


图 3-13 上海到北京航线Top N准确率

Figure 3-13 Top N Accuracy in Airline SHA-BJS

在图3-13中，我们测试了上海到北京航线Top N的准确率，其中横轴是Top N推荐机票的个数，即N的值；纵轴为式（3-16）中定义的Top N的准确率（命中率），该值越大则表示推荐效果越好。在该图中，绿色曲线表示基于用户偏好的推荐方法（UserPref），蓝色曲线表示基于低价排序的推荐方法（Price-Rank），随着推荐的Top N对象的数量N的增加，准确率也会随之增加。但是，UserPref的准确率明显优于Price-Rank的准确率，例如当Top N=20时，UserPref的准确率可达约62%，然而Price-Rank的准确率只有约30%。

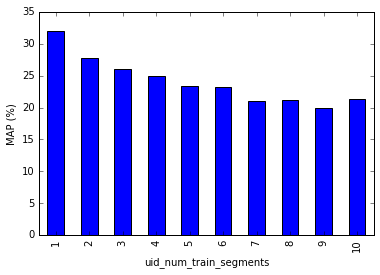


图 3-14 上海到北京航线上MAP在不同历史订单数量的用户上的分布直方图

Figure 3-14 MAP for users with different number of train segments

在本章的3.3节中我们提到过，当用户在某些非活跃航线上时，即在该航线上的历史订单过少时，其对应的偏好模型可能并不能很好的反应出该用户在该航线上的一个整体的偏好。图3-14中我们测试了上海到北京航线上MAP在不同历史订单数量的用户上的分布，其中，横轴表示测试订单中用户对应的用来计算偏好的历史订单的数量。从该图中我们可以看出，当用户在该航线上的历史订单较少时，MAP的值较大，即推荐效果较差，例如当对于在该航线只有一次历史订单的用户群体，其MAP值为32.03%，远远大于表3-2中该航线所有用户的总体的平均值26.61%。随着用户历史订单数量的增加，基于用户偏好的推荐效果越好。

为了进一步强化用户在非活跃航线，甚至是陌生航线上的偏好，在本章3.3节中我们提出了基于航线的协同过滤的方法，该方法通过整合用户在其他活跃的相似航线上的偏好来加强目标航线的偏好。在这里，我们从测试订单中筛选出了一部分具有其他活跃航线偏好的用户进行对基于航线的协同过滤的方法实验测试。

图3-15为我们通过Grid Search测试得到的，上海到北京航线上基于航线的协同过滤方法中transfer rate与Airline-based CF的MAP值的曲线图，从图中可以看出，随着transfer rate值的提高，用户在目标航线上的推荐效果得到加强，在达到约0.7时开始减弱，当transfer rate为1时（完全使用用户在其他航线的偏好），其推荐效果大大差于transfer rate为0时（只使用目标航线的偏好）的推荐效果。

图3-16比较了在上海到北京航线上，使用了基于航线的协同过滤方法来强化用户偏好的方法（Airline-based CF，蓝色）与原始的基于用户偏好模型的方法（Origin UserPref，绿色）在不同历史订单的用户上的MAP值，其中，横轴表示测试订单中用户对应的用来计算偏好的历史订单的数量。从图中我们可以看出，对于该航线上历史订单较少的用户，经过基于航线的协同过滤方法来强化用户偏好的推荐效果有较显著的提升，例如对于只有一次历史订单的用户群体，Origin UserPref方法的MAP值为32%，而Airline-based CF方法的MAP值为28.37%。随着用户历史订单数量的增加，通过其他航线来强化偏好的优化效果减弱。因此，通过基于航线的协同过滤的算法我们能够进一步地提升推荐算法对用户在非活跃航线时的推荐效果。

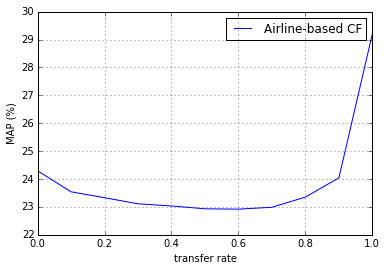


图 3-15 上海到北京航线上基于航线的协同过滤方法中transfer rate与MAP值的曲线图

Figure 3-15 MAP @ transfer rate for Airline-based CF algorithm.

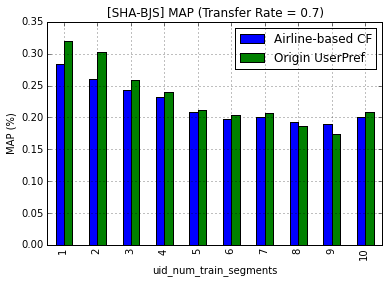


图 3-16 上海到北京航线Airline-based CF与UserPref两种方法的MAP在不同历史订单数量的用户上的分布直方图比较

Figure 3-16 MAP of Airline-based CF and UserPref for users with different # of train segments

## 本章小结

本章提出了一种基于用户偏好模型的机票个性化推荐算法，该算法根据用户的历史机票订单数据，通过分析用户在不同机票特征数据上的信息熵来为用户建立在不同特征上的个性化的权重以及用户偏好，从而构造出个性化的相似度函数来对用户在某条航线上的搜索结果进行实时的推荐。

同时，针对用户在非活跃航线以及陌生航线的数据问题，本章还提出了一种基于航线的协同过滤的算法进行用户在不同航线上偏好的跨航线的学习。

在本章的最后，我们使用国内某大型在线旅行网站提供的真实用户机票订单数据，对我们的基于用户偏好模型的机票个性化推荐算法以及基于航线的协同过滤算法进行了一系列的测试、评估和分析。

# 基于选择模型的机票个性化推荐算法

在上一章中，我们通过研究机票的显性特征属性，根据用户历史订单来为用户建立偏好模型，其思想类似于基于内容的KNN的推荐方法。本章将根据选择理论，来为用户购买机票决策过程进行建模，通过对用户机票历史订单的成对选择分析，来建立用户偏好的目标函数和优化问题。此外，我们还结合RLFM（回归隐语义模型），提出了一种基于隐性空间特征的选择模型的机票个性化推荐算法。

## 选择模型

### 选择分析（Paired Choice Analysis）

通常来说，离散的选择模型(choice model)常常是由效用理论(utility theory)[58]导出。效用(Utility)通常用来表示在一系列可选择的物品集合的一种偏好。对于一个特定的用户*u*，我们使用符号来表示用户*u*在选择空间中对两个不同对象之间的偏好关系。在本文中，其选择空间通常为用户在线搜索机票时的搜索结果的集合，定义为**S**。例如当用户*u*在搜索机票时，对机票*i*的购买意愿强于对机票*j*的（），则我们可以将这种偏好关系表示为。

此外，我们还定义了一个效用函数(Utility Function) ，用来量化用户*u*在选择购买机票时的一个效用值。根据上述定义，我们可以根据效用理论得到一个如式（4-1）所示的充分必要的条件。

成对选择模型(Pairwise Choice Modeling)作为一种成对比较(pairwise comparison)的方法，已经被广泛地应用到偏好的科学研究中。这种方法最早可以追溯到1927年Thurstone[59]建立的比较判定定律(Law of Comparative Judgment)。其模型背后的基本思想是，通过每次在一对选择对象的简单比较（）来刻画用户对每个选择对象的偏好，刻画偏好的过程通常是基于一系列这样的成对的比较。

通过式（4-2）利用随机效用模型（RUM）[60]对选择模型的推导，我们将效用函数分成了两个部分：

其中，函数通常被称为代表性效用(representative utility)，则是表示错误的项，它能够抓取那些能够影响效用但又不能被表示的特征。表示效用函数中用户*u*选择对象*i*时的输入特征。这里，对用户选择模型进行建模的一个关键是根据机票的数据特征，对建立有效的效用函数模型，其函数模型的建立以及训练将分别在4.2和4.3节进行详细的介绍。

根据Thurstone的比较判定定律中的Case 5 [59]，偏好的概率能被定义为选择的效用，如式（4-3）所示：

如果我们将式（4-2）带入式（4-3），我们可以得到：

其中，CDF表示累计密度函数（cumulative density function）。

如果我们假设，将可以得到一个logit模型[60]。最后，在一选择对上的偏好概率模型能够被表示为一个logistic函数，如式(4-5)所示，其中表示函数模型中的待估参数的集合。

在用户搜索选择并购买机票的过程中，我们可以认为用户选择购买的那张机票的偏好大于搜索结果中其他未选择购买的机票。因此，进一步我们可以推导出用户在历史购买的订单中，基于所有的选择对，即选择购买的机票和搜索结果中其他机票组成的选择对的偏好，。我们定义**T**为用户购买的历史订单的集合，为用户选择历史订单*i*时对应的搜索结果的集合。因此，我们可以得到上的联合似然函数：

相似的，我们可以进一步推导对与在历史购买订单中，所有的用户的购买偏好产生的选择对，，其联合似然函数为：

其中，**U**是所有用户的集合，而则为用户的历史订单。

### 参数的先验

上一部分里我们推导出了似然函数，因此我们可以使用最大似然估计（MLE）的方法来估计式（4-7）中模型的参数。然而，的参数估计缺少正则化（Regularization），可能会出现过度拟合的问题从而导致一个差的预测性能。我们可以通过在模型参数上添加一个先验来完善我们的模型成为贝叶斯模型，其中指的是超参数（hyper-parameters）。这里我们可以采用下述两种先验：

1. **高斯先验（Gaussian Prior）**

其中，K是待估参数的数量，表示方差，**I**表示单位矩阵，表示标准化常数。

1. **拉普拉斯先验（Laplace Prior）**

同样，K是待估参数的数量，表示方差，表示标准化常数。

相比较于高斯先验，拉普拉斯先验有更稀疏的尾，因此对于给定的相同的方差能使后验更集中在零均值（zero mean）。通常使用带有拉普拉斯先验的参数导出稀疏的结果。

### 优化问题

通过似然（式4-7）以及先验（式4-8和式4-9），我们可以通过贝叶斯理论得到后验概率：

因此，我们可以通过最大化后验来学习待估参数。显然，最大化式(4-9)可以等效于最小化它的负的log形式。通过这样的变换，我们可以得到目标函数：

在目标函数中，我们定义式作为正则化项，用来防止过度拟合问题。

对于高斯先验的情况，正则化项对应了L2-norm正则化（L2-norm Regularization）：

其中，作为正则化参数。L2正则化的作用主要在于其能够缩小待估参数的值，防止值过大导致过拟合。

对于拉普拉斯先验的情况，正则化项对应了L1-norm的正则化（L1-norm Regularization）：

同样，在式中为正则化参数。L1正则化会诱发稀疏，也就是说，待估参数中大部分的值将会是0。

## 机票选择的效用函数建模

在上一节中，我们通过对用户机票历史订单的成对选择分析，为用户在历史订单上的选择的偏好建立了似然函数，并根据最大似然估计，建立用户的偏好目标函数和优化问题。效用函数（式4-2），作为用户在选择过程中对物品的偏好的表现，其函数模型的好坏将直接影响到用户对机票选择预测的结果。

本节将主要讨论我们如何结合机票的数据特征，来为表现用户在选择机票过程中偏好的效用函数进行函数模型的建立。

### 基于显性偏好的效用函数模型

在上一章中，我们根据用户机票的历史订单，通过分析机票的显性特征属性来为用户建立偏好模型，其用户的偏好直接作用于机票的各个显示特征上。但是，该用户偏好模型中并没有一个目标函数和优化问题，其本质是一种基于内容的KNN的推荐算法。

这里，我们可以参照上一章基于用户偏好模型的方法来定义效用函数，如式（4-14）所示，其中为机票的显性的特征属性的向量，为效用函数中的待估参数，表示用户对机票显性特征的偏好。

这里的为用户对机票的输入特征，包括用户*u*的信息，机票*i*的显性特征属性，以及当前用户搜索的航线*a*。不同于上一章基于用户偏好模型的方法中将一张机票中的每一个特征属性的离散变量转换成一个离散的向量，这里的指的是一张机票i中所有特征属性的向量，其将原本的不同特征的向量拼接成一个更长的向量，即，其维度我们定义为Nt。

如3.3节中所述，由于不同航线上的机票特征具有不同的数据分布，同时用户在不同的航线上也会有着不同的偏好，因此我们无法通过一个统一的偏好向量来抓取用户在不同航线上的不同偏好。式（4-15）中我们添加了偏好向量，用来描述用户在特定航线*a*上的偏好，而偏好向量是用来抓取用户在所有航线上订票时的偏好。

进一步，我们还可以在效用函数中添加一项公共的偏好来抓取所有用户在这条航线上的偏好，从而来处理用户在一些非活跃航线以及新用户的冷启动问题，如式（4-16）所示。也就是说当这个用户的历史订单很少时，即我们很难通过训练得到这个用户的偏好时，我们可以用这条航线上所有用户的一个公共的偏好来表示这个用户可能的偏好。

因此，用户在搜索航线*a*上的机票时，作用于机票显性特征属性的偏好可以分为三部分：用户在航线*a*上的特定偏好，用户在所有航线上对机票的总体偏好，以及所有用户在航线*a*上的公共偏好。这里的三个待估的向量参数的维度等于经离散化处理后的机票特征的向量的维度Nt**。**

### 基于RLFM的效用函数模型

隐语义模型(LFM)是一种统计学的模型，可以将显性变量转化为一系列的隐性变量，从而发觉模型中的一些隐性特征。隐语义模型在推荐算法中的应用通常又被称为矩阵分解(Matrix Factorization)技术[8]，其通过发现用户以及物品的隐性空间特征，来为用户在物品集上其他物品的偏好做出准确的推荐。Rendle[38]在贝叶斯个性化排序（BPR）中，也结合了矩阵分解技术，其对应的效用函数可被看做：

其中表示用户*u*的隐性空间特征，表示物品*i*的隐性空间特征。

对于矩阵分解技术来说，通常我们需要一个固定的物品集合，来学习集合中每个物品的隐性空间特征。然而，正如前文所说的，机票作为一种受时间影响的价格敏感的动态商品，具有价格波动性和价格的敏感性的特征。此外，机票的其他一些特征属性，例如退改签的政策，在不同的时刻也会发生变化。对于这些波动性较强的属性，我们无法很好地为机票建立一个固定的物品集合。

通过分析机票的数据特征我们发现，对于机票所属的航班来说，其航班的信息，例如航班的起飞到达时间，航空公司，机型的大小等等，通常具有较好的稳定性。也就是说，对于一个特定的航班，其航班信息即在相当长的一段时间内不会随意发生变化。因此，我们可以将机票的特征属性分为两个部分，一个是机票对应的航班的静态属性，一个是机票其他非航班信息的动态属性。对于前者，我们可以为机票在航班的维度上建立一个航班的集合**M**。

Agarwal[61]对带有外部特征的二元数据提出了一种基于回归的隐语义模型(Regression-based Latent Factor Model, RLFM)，其通过学习和训练一个公共的矩阵将外部特征映射到隐性空间。类似地，如果我们把这种思路放到我们的选择模型中，将用户*u*和机票*i*对应的航班看成二元数据，而机票的其他动态的显性特征属性可以看做是该航班的外部特征，对此，我们可以为效用函数建立如下的函数模型：

其中，为机票的显性特征。为将机票的显性特征映射到隐性空间的公共的矩阵。因此，对于机票i上的隐性空间的特征可以表示为两个部分的叠加，一部分是通过映射到隐性空间特征的机票i显性特征属性，另一部分是其对应的航班上的隐性空间特征。

考虑到不同航线上的机票特征具有不同的数据分布，式(4-19)中的应该是区分航线的，也就是说，不同航线上应该有一个对应当前航线*a*的公共的矩阵，用来将该航线上机票的显性特征转化为隐性空间特征。同时,考虑到用户在不同航线上会有不同的偏好，我们可以在式(4-18)中加入一个属于用户航线偏好的隐性空间特征项。此外，我们还可以加入一个表示机票显性特征属性的回归项，例如对于一些折扣大价格低的机票常常对用户具有额外的吸引力。综上，我们可以得到下列公式所表示的效用函数的模型：

其中函数模型的待估参数包括向量参数，，，，参数以及矩阵参数，其中*u*，*a*，分别指用户集合中的用户*u*，航线集合中的航线*a*以及机票*i*对应的航班集合中的航班。假设我们给定的隐性空间特征的数量为Nk，显性特征的数量即的维度为Nt，则向量参数，和的维度为Nk，矩阵参数的维度为，向量参数的维度为Nt。

## 基于选择模型的学习和推论

上两节中我们介绍了如何使用选择模型来对用户购买机票的偏好进行建模，并提出了目标函数和优化问题。本节将主要介绍我们如何通过训练用户的历史订单数据来估计参数，解决我们的优化问题，并对用户在搜索机票时进行个性化的机票推荐和预测。

### 参数估计

最小化目标函数（式 4-11）能被看做是解决正则化的逻辑回归问题。通常，我们可以使用基于梯度(gradient-based)的方法来找出待估参数。这里，目标函数的梯度是一系列参数的偏导数的集合，如式（4-25）所示。

首先，似然函数的偏导数可以很容易地通过式(4-5—4-7)导出：

其中，

进一步，我们能给出L2和L1-norm的正则项的偏导数：

* **L2正则化（参见式4-12）：**
* **L1正则化（参见式4-13）：**

其中，是一个分类函数，定义如下：

对于所有用户的历史订单来说，其可能的选择对（）的数量非常大，该数量可由式（4-30）估计。其中表示历史订单的数量，表示历史订单对应的平均的搜索结果的数量。所有用户的历史订单的数量通常是一个很大的值，例如在我们的数据集中，过去两年的国内机票的历史订单数量有近9000万条。

对于如此大数量的，用标准的梯度下降方法来解决优化问题显然并不高效。为了解决这个问题，我们采用了随机梯度下降算法（SGD）来更新参数，也就是说我们在每个数据点，在这里指的是每个选择对上来更新参数。

此外，我们在随机梯度算法中还采用了bootstrap抽样的策略，来随机地抽取选择对进行参数的更新。这样的策略在贝叶斯排序模型[38]中被证明是非常高效的，特别是对于大量的选择对来训练参数的情况。

1. **L2正则化的随机梯度算法**

对于一个数据点，也就是一个选择对，其对于各个参数的随机梯度的更新由式（4-31）给出：

其中为学习比率(learning rate)。然而，我们并不需要对每一个选择对都循环遍历地进行随机梯度的更新。因此，我们可以使用bootstrap抽样的策略有放回地随机地抽取一部分的选择对的集合。例如，对于用户*u*的某个历史订单*t*，假设*t*对应的用户当时的搜索结果**S**数量有100条（包括*t*），则我们可以产生99个选择对, 其中，通常我们并不需要所有的选择对，这里我们可以使用bootstrap抽样，有放回地抽取一部分比例（Sample Rate）的选择对进行参数的训练。相对应地，我们设计了L2正则化的随机梯度算法，参见算法（4-1）。

在算法（4-1）更新参数的12—13步骤中，表示效用函数中待估参数的集合，而表示待估参数集合中的某个待估参数，其可能对应用户集合中不同用户和航班集合中的不同航班，同时不同的效用函数模型也对应不同的参数，对于我们在4.2节中定义的两个函数模型：

1. **基于显性偏好的效用函数模型**

根据基于显性偏好的效用函数模型，式（4-16），其待估参数包括用户集合中不同用户*u*和航线集合中不同航线*a*的对应的向量参数,,，这里的三个参数的维度等于经离散化处理后的机票特征的向量的维度Nt**。** 因此，对于基于显性偏好的效用函数模型，算法(4-1)中用来更新参数的12—13步骤可以被进一步细化为：

为了方便区分，我们将基于显性偏好的效用函数模型的L2正则化随机梯度训练算法命名为PCM-R-L2，其中PCM表示成对选择模型，R表示回归，L2表示正则化项。

1. **基于RLFM的效用函数模型**

根据基于RLFM的效用函数模型，式（4-21—4-24），其待估参数包括用户集合中不同用户*u*，航线集合中不同航线*a*和航班集合中不同航班的对应的向量参数，，，，参数，以及矩阵参数。同样，对于基于RLFM的效用函数模型，算法(4-1)中用来更新参数的12—13步骤可以被进一步细化为：

相应地，我们将基于RLFM的效用函数模型的L2正则化随机梯度训练算法命名为PCM-RLFM-L2

1. **L1正则化的随机梯度算法**

L1正则化通常不能够很有效地应用到基于随机梯度的训练算法中，一方面是由于高维度的特征向量，另一方面是由于估计的梯度的具有扰动性，这种不稳定性会破坏从单个数据点得到的梯度的稀疏性[62, 63]。

为了能够通过随机梯度的方法得到足够稀疏的待估参数，我们采用了一种简单但有效的策略，通过累计的惩罚来惩罚每个参数[63]。对应的L1正则化的随机梯度算法可参见算法4-2。

同之前L2正则化的随机梯度算法类似，我们也可以相应地在算法（4-2）更新参数的12—13步骤中，对于4.2节定义的两个不同的函数模型，细化其参数更新，并得到相应的两个基于L1正则化随机梯度的PCM-R-L1和PCM-RLFM-L1算法。

1. **算法的复杂度**

在算法（4-1）和算法（4-2）中，每一次迭代的复杂度都为，其中，N为用来训练的所有用户的历史订单的数量，B为历史订单对应的bootstrap抽样搜索结果的数量，K为所有待估参数的数量。由于B和K可以被当做为常数，因此每一次迭代的复杂度为线性时间的O(n)。

|  |
| --- |
| **Algorithm 4-1:**  **Paired Choice Model – L2** |
| **Input： T –** Ticket Set of Historical Flight Order of all Users  **S** – Ticket Set of search results for each historical flight order  **F** – Set of Flight Ticket Features  **U** – Set of all users  **M** – Set of all airlines  – Regularization Parameter  r – bootstrap sampling rate  – initial learning rate  Nk – number of Latent Factors  MAXN – maximum number of iterations |
| **Output： –** set of to be estimated parameters in ,  which includes parameters for different users and different airlines |
| 1. , 2. for it = 1 to MAXN: 3. for in **T**: 4. Define as the user of 5. Define as the airline of 6. Define as the flight of 7. Define as the search results for in **S** 8. Set as the bootstrap sampling samples from 9. for in : 10. if == : 11. continue 12. for in according to , , : 14. End for 15. End for 16. Return ,if it convergences or stop conditions are reached. 18. End for 19. End for |

|  |
| --- |
| **Algorithm 4-2:**  **Paired Choice Model – L1** |
| **Input： T –** Ticket Set of Historical Flight Order of all Users  **S** – Ticket Set of search results for each historical flight order  **F** – Set of Flight Ticket Features  **U** – Set of all users  **M** – Set of all airlines  – Regularization Parameter  r – bootstrap sampling rate  – initial learning rate  Nk – number of Latent Factors  MAXN – maximum number of iterations |
| **Output： –** set of to be estimated parameters in ,  which includes parameters for different users and different airlines |
| 1. , 2. for it = 1 to MAXN: 4. for in **T**: 5. Define as the user of 6. Define as the airline of 7. Define as the flight of 8. Define as the search results for in **S** 9. Set as the bootstrap sampling samples from 10. for in : 11. if == : 12. continue 13. for in according to , , : 15. CumPenalty() 16. End for 17. End for 18. End for 19. Return ,if it convergences or stop conditions are reached. 21. End for   **----------------------------------------**   1. Sub Procedure CumPenalty(): 3. if then 4. if then |

### 机票推荐和预测

根据效用理论，用户对机票的选择可以用效用来测定（式4-1）。效用越高则意味着用户对这张机票的偏好越强。因此，我们可以通过计算用户在搜索机票时当前搜索结果里机票的效用来对其进行排序，并为用户推荐出偏好最强，也就是效用最高的一些机票。

在上一章中，我们提出了一个离线计算和在线计算单元分离的机票个性化推荐的计算框架。同样，对于本章的方法，我们也可以先离线地使用算法（4-1，4-2）通过机票的历史订单数据训练和更新所定义的效用函数的参数，然后通过该效用函数来在线地实时计算用户搜索结果中的效用，并为其购买做出推荐和预测。

## 相关实验与结果分析

### 实验数据集

在本章中，我们使用与上一章实验相同的订单数据集，并选择了相同的4条热门航线作为测试航线，包括上海到北京（SHA-BJS），上海到广州（SHA-CAN），上海到厦门（SHA-XMN），上海到成都（SHA-CTU）。

此外，本章使用了选择模型对机票个性化问题进行建模，并通过贝叶斯模型建立了目标函数从而将机票个性化问题转化为选择模型中用户效用函数的参数优化问题。为了解决该优化问题，我们需要训练用户在预订历史机票时选择机票成对比较的数据，也就是用户的订单（选择的机票 i），以及用户在当时预订的搜索结果页看到的其他的机票（未选择的机票 j）组成的逐对排序 。因此，我们在训练模型时，除了上一章所述的机票的用户订单数据外，还需要用户在预订机票时搜索结果页面上所看到其他机票的数据。但是，我们的数据集中并不包含这部分数量巨大的用户行为数据。

同样，我们这里使用了上一章在测试集中通过找出用户下单时刻前后一段时间区间内其他用户的订单的方法，来模拟用户在预订机票的时刻所看到搜索结果。如上一章所述，这样的方法会给实验带来系统误差，因此在本章中我们也使用热门航线上的数据来尽可能地降低由于用户行为数据缺失所带来的这一部分的系统误差。

### 评价指标

为了能够很好地和上一章中的实验进行横向的比较，本章中也使用了平均准确率均值（MAP）的评价指标来比较和评估个性化推荐算法在不同测试航线上的推荐效果。MAP的值越低，则表示用户实际选择的机票在推荐列表的排序中越靠前，即推荐效果越好。

此外，由于本章使用了选择模型来对用户在选择过程中逐对（pair-wise）的偏好进行优化，因此我们还引入了第二章中介绍的AUC的评价指标，结合基于选择模型机票个性化推荐算法，AUC可以被定义为：

其中，为用户的集合，为用户*u*对应的测试订单集合，为测试订单*i*对应的用户当前搜索结果的集合，和分别表示根据训练模型预测得出的用户对机票*i*和机票*j*的效用值。

### 相关的比较算法

本章使用了选择模型对机票个性化推荐问题进行建模，根据模型的先验，我们提出了两种L2和L1的正则化方式。此外，对效用函数我们也建立了显式特征的线性回归(Regression)以及回归隐语义模型（RLFM）的两种不同函数模型。因此，我们将首先测试和评估上述模型两两组合的四种方法，分别为：PCM-R-L2，PCM-R-L1，PCM-RLFM-L2和PCM-RLFM-L1。其中R表示作用在显式特征上的回归模型的效用函数模型，RLFM表示回归隐语义模型的效用函数模型，L2和L1分别表示两种正则化方式。

除了本章提出的方法，我们还将测试和评估其他几种比较算法，包括贝叶斯排序的算法中的矩阵分解的方法（BPR-FM）以及上一章中提出的基于用户偏好模型的机票个性化推荐算法（UserPref），还有作为基准线（Baseline）的低价排序（Price-Rank）的方法。

在上一章中我们有提到，对于协同过滤的方法我们无法直接对机票这种具有价格波动性和价格敏感性的动态商品进行一个固定的物品集合的建立，我们的一个解决方法是枚举动态属性中所有的组合，来构建这样的一个推荐对象的集合。在BPR-FM中，我们也使用同样的方法来对机票的不同动态属性的组合进行物品集合的构建。

### 实验结果与分析

与上一章的实验部分类似，我们使用2014年12月份的测试航线上的订单数据对机票个性化推荐算法进行测试。

图4-1和表4-1展示了不同测试航线上基于选择模型的个性化推荐算法的MAP评价指标的比较。在图中，我们首先测试比较由不同正则化项L1和L2，以及不同效用函数模型R（回归）和RLFM两两组合得到的共四种选择模型的算法，即PCM-R-L1，PCM-R-L2，PCM-RLFM-L1，以及PCM-RLFM-L2。

图4-1不同测试航线上基于选择模型的个性化推荐算法的MAP比较

Figure 4-1 MAP for personalized flight recommendation algorithm in test airlines

表 4-1不同测试航线上基于选择模型的个性化推荐算法的MAP比较

Table 4-1 MAP for personalized flight recommendation algorithm in test airlines

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **MAP (%)** | | | | | | |
| **PCM-R-L1** | **PCM-R-L2** | **PCM-RLFM-L1** | **PCM-RLFM-L2** | **BPR-MF** | **UserPref** | **Price-Rank** |
| **SHA-BJS** | 27.03 | 26.16 | 25.2 | 24.76 | 41.58 | 26.61 | 48.51 |
| **SHA-CAN** | 30.23 | 29.31 | 28.51 | 27.81 | 42.31 | 29.69 | 46.49 |
| **SHA-XMN** | 29.1 | 28.23 | 27.26 | 26.53 | 41.95 | 28.05 | 50.07 |
| **SHA-CTU** | 31.03 | 29.78 | 29.5 | 28.23 | 43.65 | 30.56 | 45.52 |

在这四种选择模型的算法中，如果我们比较两种效用函数模型：回归隐语义模型（RLFM）和简单回归模型（R），我们可以发现RLFM效用函数的选择模型比简单回归效用函数的选择模型具有更低MAP值，即更好的推荐效果。与对机票显性特征的简单的回归模型相比，RLFM能够从隐式空间特征中更精准地刻画出用户在选择机票过程中的效用偏好以及各机票显性特征之间的内在联系。

类似地，如果我们比较两种正则化L1和L2方法，我们可以发现L2正则化方法比L1正则化方法具有更低的MAP值，即更好的推荐效果。产生这一结果的原因可能在于：L1正则化会诱发稀疏，导致待估参数中大多数参数为0，而L2正则化能够使待估参数，即隐式空间特征中的值具有较好的连续性，同时防止其过大产生过拟合。这也从侧面反映出了，用户选择机票大多数情况是综合了多方面特征因素后做的一个决策，而不是只是由某些特定的特征因素导致的。

在图4-1中，我们还测试评估了贝叶斯个性化排序的矩阵分解算法（BPR-MF）。该方法也属于LFM模型的一种方法，然而相对于PCM-RLFM-L2和PCM-RLFM-L1，该方法的MAP值却很大，推荐效果较差。正如前文所述，造成这种方法推荐效果差的原因可能在于：由于机票的动态属性特征，我们通过枚举机票所有属性组合的方法来构建的推荐对象集合不能很好地反映出机票各个特征属性之间的内在联系，而且由于组合数量巨大，会给协同过滤方法带来严重的数据稀疏问题。

此外，图4-1中还测试比较了我们上一章中提出的基于用户偏好模型的机票个性化推荐算法（UserPref）。从图中可以看出，基于用户偏好模型的机票个性化推荐算法的MAP略大于基于选择模型的方法，但是差距并不大。特别是当我们将UserPref与PCM-R-L2/L1比较，我们会发现这两种推荐的方法都是基于相同的显式特征，并通过用户的历史订单数据计算用户对这些特征的显性偏好。不同的是，UserPref的方法更类似于基于内容推荐的一种KNN的方法，其偏好可以简单直接地从用户历史订单中计算获得，而不需要花费大量的时间进行迭代式地参数训练和学习；而PCM-R-L2/L1则是通过回归模型效用函数来解决用户对机票选择模型中的最优问题，其解决的方式是通过随机梯度下降迭代式地进行回归模型中参数的估计。假设我们可以通过这样的迭代训练和学习得到一个最优解，那么我们就可以将PCM-R-L2/L1的方法近似地看做是UserPref方法的一个最优的边界。从图4-1中的实验结果上看，UserPref的MAP值已经非常接近PCM-R-L2/L1，因此，我们可以得出，基于用户偏好模型的机票个性化推荐算法具有较好的推荐效果，且其推荐效果接近于显式特征上线性回归模型的最优边界。

最后，图4-1中还测试比较了基于低价排序的推荐方法（Price-Rank），与上一章的结果相同，基于低价排序的策略并不能满足多数用户的需求，其推荐效果较差，仅仅略优于随机的推荐方法（MAP=50%）。

对于LFM模型，不同的隐式因子的数量K也会影响模型的推荐效果。通常来说，K越大，推荐效果相对越好，但是待估的隐式因子越多，模型训练需要计算量越大，收敛越慢，训练时间越长。

图4-2展示了上海到北京航线上，PCM-RLFM-L2算法在不同的隐式因子数量K上的迭代次数与AUC值的关系图，其中横轴表示算法训练时迭代的次数，纵轴表示当前学习得到的模型在测试集上的推荐效果。这里，我们使用了AUC的评价指标来表示推荐算法模型在用户逐对选择过程中的推荐效果，AUC的值越大，表示推荐效果越好。从图中可以看出，随着迭代次数的增加，AUC的值逐渐变大，到达一定的数值后开始收敛。对于不同的隐式因子的数量K，当K值从10开始逐渐增大到30时，我们可以看出收敛后的AUC的值也逐渐变大，而当AUC的值从30增加到50再增加到100时，收敛后的AUC值不再有明显的增大。因此，从该图中我们还可以看出，迭代训练收敛后的MAP的值随着隐式因子的数量K的增大而增大，后逐渐收敛并趋于饱和。

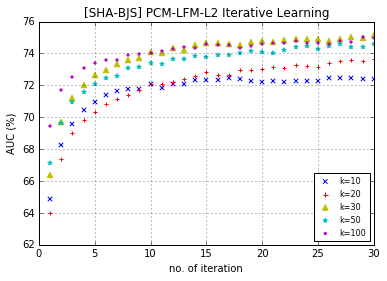


图 4-2 上海到北京航线PCM-RLFM-L2算法在不同隐式因子数量K上的迭代次数与AUC值的关系图

Figure 4-2 AUC @ iteration no. for PCM-RLFM-L2 algorithm with different number of latent factors k

## 本章小结

本章根据选择模型，通过对用户在机票历史订单的成对选择分析，建立了用户在选择机票时效用函数的目标函数及其优化问题。同时，本章还使用了简单回归模型和回归隐语义模型这两种模型对选择模型中的效用函数进行函数建模，并提出了随机梯度下降的方法对模型中的待估参数进行迭代式的训练和学习。在本章最后，通过相关的实验，进一步地对不同的机票个性化推荐算法的测试，评估和分析。

# 面向大数据的机票个性化推荐系统设计

前边两章中，我们分别提出了基于用户偏好模型的机票个性化推荐算法以及基于选择模型的机票个性化推荐算法，两种算法都可以被分离为离线计算和在线计算的两个计算逻辑单元，其中离线计算单元是通过对用户机票数据的深度的分析和挖掘，根据算法模型为用户离线地计算出对应偏好信息；而在线计算单元则根据已计算好的用户偏好信息，为用户在线搜索机票时做出实时的预测和推荐。两种算法都需要对海量用户数据进行深度的分析和复杂的计算。

本章将结合大数据技术，介绍我们面向大数据的机票个性化推荐系统设计。首先，我们介绍了面向在线旅游服务的大数据平台，及旅游大数据中的数据的收集和处理流程，并提出基于大数据平台的机票个性化推荐系统设计。接着，我们还设计和实现了本文提出的两种机票个性化推荐算法的基于Spark的并行化实现。最后，我们还设计并执行了相关测试实验，对两种基于Spark的并行化的机票个性化推挤算法进行了性能的测量评估和比较。

## 面向大数据的机票个性化推荐系统框架设计

### 面向在线旅游服务的大数据平台

随着旅游信息化、智能化的推进和发展，越来越多的用户旅游数据通过移动互联网，物联网被生产出来。大数据技术的出现使得我们可以对海量的用户数据进行存储、处理以及深度的分析和挖掘。在第二章中，我们介绍了大数据的数据处理流程以及大数据平台的一些关键技术。这里，我们以某大型互联网旅游公司为例，介绍如何结合并使用当前主流的大数据技术，建立面向在线旅游服务的大数据平台。

图5-1展示了OTA的面向在线旅游服务的大数据处理体系架构。从图中可以看出，数据流自上而下，从开发的互联网，到业务应用层，再到数据收集层，最后到大数据层。数据收集层收集的数据包括：爬虫从互联网中第三方网站收集的旅游信息数据；内部的业务数据；以及网站运行状态和系统度量的数据；还有用户的行为数据。这些数据进入大数据层，一部分给Hadoop的HDFS或者分布式的数据库HBase存储，这些数据可以通过Hive已经MapReduce的作业离线地进行进一步的数据挖掘和分析；另一部分交给Storm进行实时的数据流处理，可以用来对用户行为进行实时的分析，以及通过对实时数据的异常检测进行应用的监测和预警。

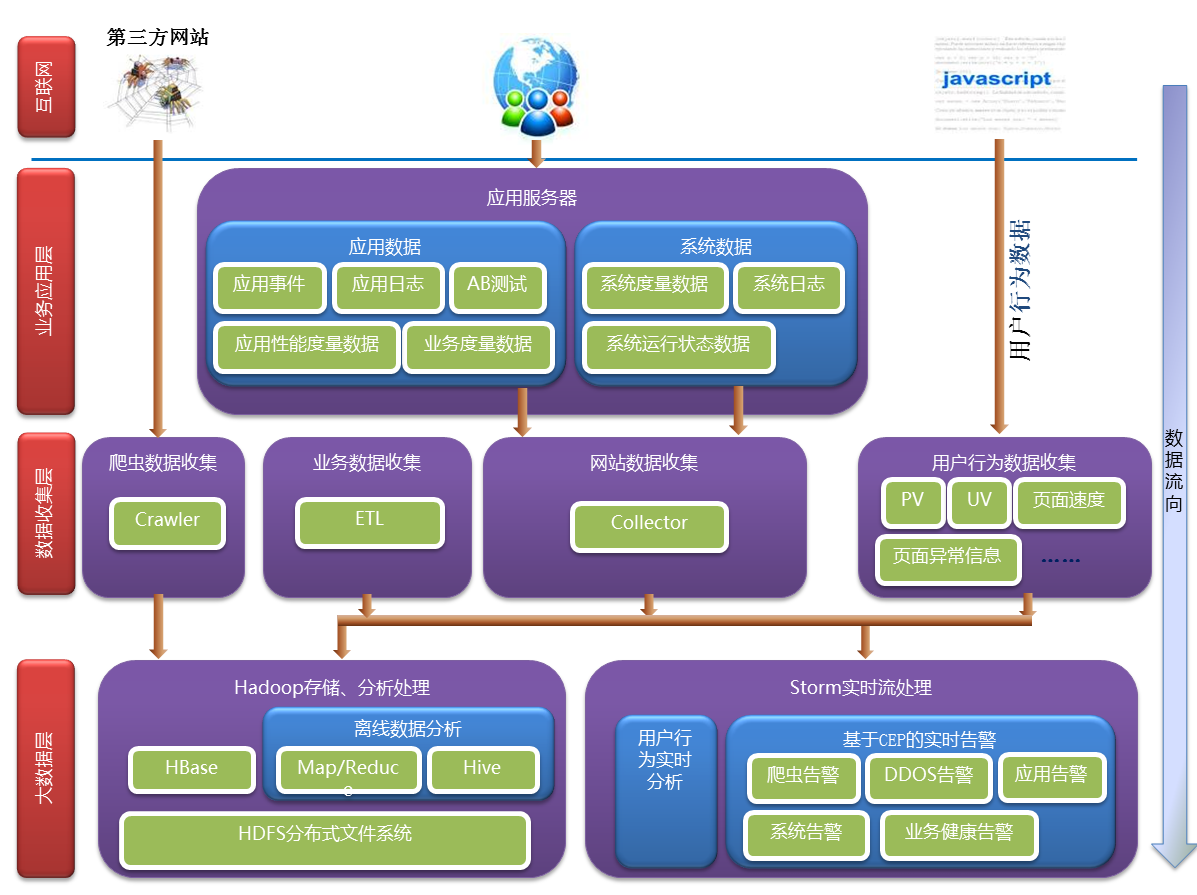


图 5-1面向在线旅游服务的大数据处理体系结构

Figure 5-1 Architecture of Big Data Process for online travel services

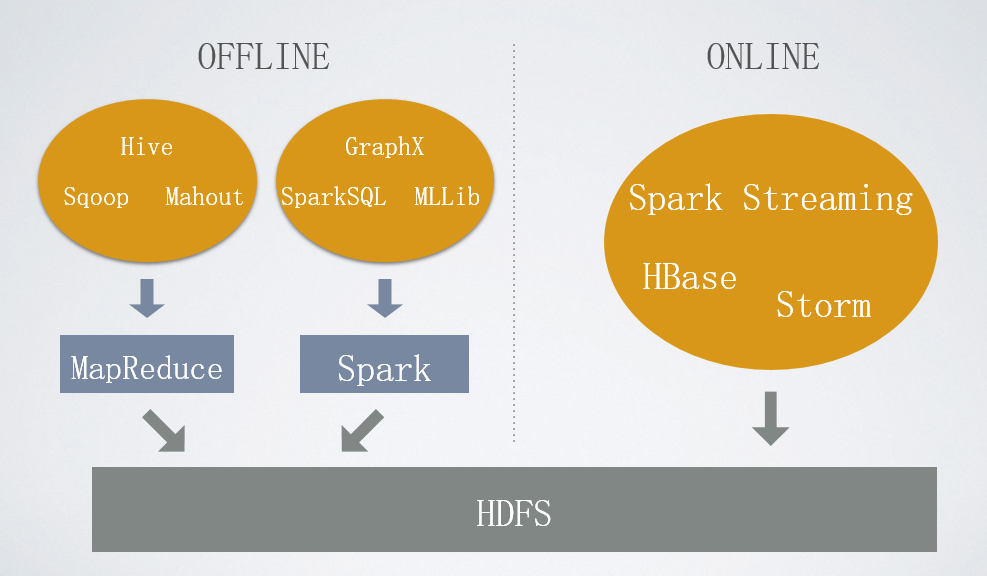


图5-2 面向在线旅游服务的大数据计算平台

Figure 5-2 Big Data computing platform for online travel services

图5-2展示了携程旅行网面向在线旅游服务的大数据计算平台的基本组件。从图中可以看出，最底层的为HDFS作为分布式文件系统以及数据仓库，在其上边分别建立了离线计算和在线计算的两个部分。离线计算部分中包括了基于MapReduce分布式计算框架的Hive，Sqoop以及Mahout，以及基于Spark分布式计算框架的SparkSQL，GraphX以及MLlib，用来对数据进行数据分析和数据挖掘；在线计算部分中包括了Spark Streaming和Storm用来进行数据流的实时处理，以及HBase作为分布式数据库能够高效可靠地对数据进行实时插入和更新。

### 面向大数据的机票个性化推荐系统设计

和传统的推荐系统一样，机票个性化推荐系统包含四个重要的组成部分，包括数据来源，用户模型，推荐算法以及应用展示推荐结果的用户界面。

根据上文我们所介绍的面向在线旅游服务的大数据平台，我们提出了一个面向大数据的机票个性化推荐系统的设计框架，如图5-3所示。我们将机票个性化推荐系统的总体框架分为三层，分别为数据层，推荐逻辑层以及应用层。

数据层包含两部分：一部分是机票个性化推荐系统的数据源—用户的订单数据以及用户选择的行为数据，另一部分是经过推荐算法离线学习得到的用户偏好数据。数据层中的数据可以存在大数据平台的HDFS中，对于用户的订单数据和行为数据，我们可以通过SQL引擎SparkSQL进行数据清洗和数据的预处理，并作为推荐算法中的输入数据；对于学习得到的用户偏好数据，我们可以通过建立分布式数据库HBase来保证实时的计算中能够得到高效即时的数据查询、读取性能。

推荐逻辑层包括推荐系统中的推荐模型和推荐算法部分。结合大数据计算平台和机票个性化推荐算法，我们在该层也建立了一个离线计算单元和一个在线计算单元。离线计算单元通过一个调度器定时的启动机票个性化推荐算法，通过用户的订单数据及行为数据来训练并更新用户的偏好模型，这里我们将我们的机票个性化算法结合了Spark分布式计算框架，使用户偏好能够通过高效的分布式计算进行并行化的学习和训练，在下一节中我们会详细介绍结合Spark的推荐算法的并行化设计和实现过程；在线计算单元主要根据偏好模型以及已经离线学习好的用户偏好数据，在线实时地计算用户对搜索结果中各个机票的偏好程度。

应用层包括了应用展示推荐结果的用户界面。在应用层中，当用户进行新的一次机票的搜索时，用户的信息以及机票搜索引擎返回的搜索结果会作为输入给推荐系统的接口，通过在线计算单元计算用户对搜索结果中每一张机票偏好程度，并返回Top N个偏好程度最高的机票作为推荐的结果。

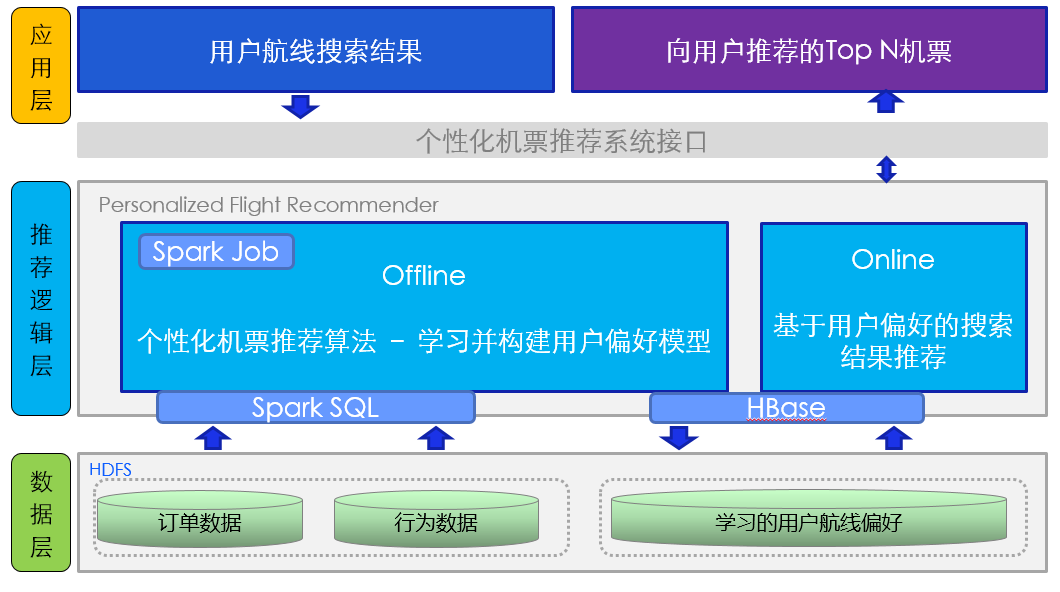


图 5-3 面向大数据平台的机票个性化推荐系统的框架

Figure 5-3 System framework for Personalized Flight Recommendation on big data platform

## 结合Spark的推荐算法并行化实现

在上一节中，我基于面向在线旅游服务的大数据平台，提出了一种通用的包含离线和在线计算单元的机票个性化推荐系统的框架设计。在离线计算单元中，我们结合了Spark分布式计算框架，通过高效的分布式计算来并行化地学习和训练用户偏好模型的。本节将详细介绍本文第三章和第四章中提出的两种个性化机票推荐算法的并行化的实现。

### 基于用户偏好模型的机票个性化推荐算法的并行化实现

在基于用户偏好模型的机票个性化推荐算法中，我们根据用户在各个航线上的历史订单数据来计算用户在航线上的偏好。由于对于不同用户在不同航线上的偏好计算是相互独立的，因此我们可以很简单地对于不同的用户及其不同航线上的偏好进行独立并行的运算。

此外，对于用户在不同特征上的偏好向量的计算也是相对独立的，我们也可以进行进一步的并行运算，算法5-1基于算法3-2展示了离线计算用户偏好的并行化算法。

|  |
| --- |
| **Algorithm 5-1:**  **Parallel Offline User Preference Calculation** |
| **Input： –** Ticket Set of Historical Flight Order of User u  **F** – Set of Flight Ticket Features |
| **Output：p** – Set of User Preference Vector for each feature  **w** – Set of weight for each feature |
| 1. **Calculate the entropy for each feature in F by :**   For in **F** **Parallel do**:    End For   1. **Calculate the weight for each feature preference:**   For in **F** **Parallel do**:    End For   1. **Calculate the preference vector for each feature :**   For in **F** **Parallel do**:      End For   1. **Return preference vector and weight for each feature** |

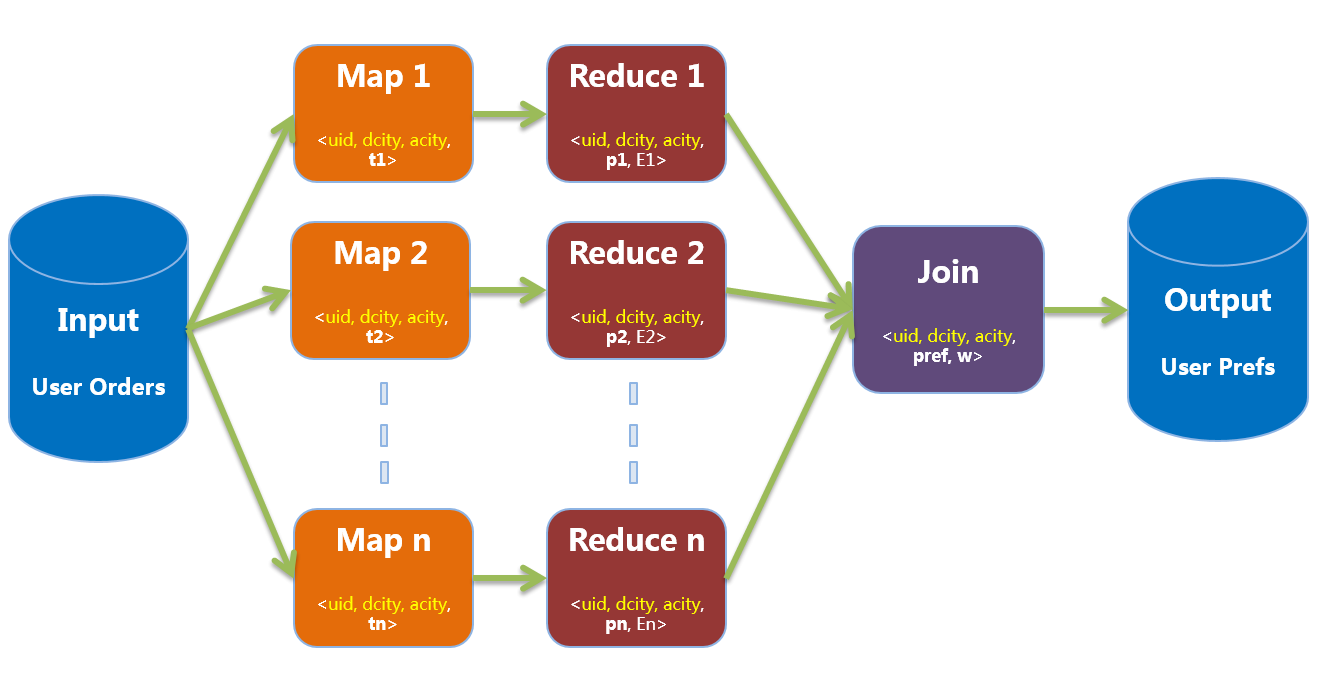


图 5-4基于用户偏好模型的机票个性化推荐算法的Spark并行化实现

Figure 5-4 Implementation of parallelized UserPref recommendation algorithm on Spark

在图5-4中，我们给出了基于用户偏好模型的机票个性化推荐算法在Spark并行计算框架上的实现。图中的每个方框代表着一个抽象的Spark分布式算子，方框中的尖括号表示经过算子运算后的分布式数据集（RDD）中元组所对应的形式。

从图中我们可以看出，用户订单的数据首先经过n个Map算子，每个Map对应一个机票的特征属性，在Map中数据集的元组可表示为<uid, dcity, acity, **tn**>，其中uid为用户的标识id，dcity和acity为航线的标识，**tn**为用户历史订单机票的第n个特征属性。在订单数据经过n个Map被拆分成各个特征属性后进入对应的n个Reduce算子进行用户在航线上的各个特征属性偏好的聚合计算。在Reduce中，对应的key为<uid, dcity, acity>，因此每个Reduce都可以得到用户在航线上对应特征上的所有数据，并计算其在该特征上的偏好向量**pn**及对应的信息熵值En。最后，一个统一的Join算子将联合用户在航线上对应的各个特征偏好向量，并计算得出在不同特征上的偏好权重。

### 基于选择模型的机票个性化推荐算法的并行化实现

在基于选择模型的机票个性化推荐算法中，我们采用了随机梯度下降（SGD）的算法来对模型进行迭代式的学习和训练。在这里，我们借鉴了Zinkevich等人[64]提出的并行化SGD的方法，其主要思想是将训练集随机拆分成与并行化的计算单元数量相等的k等份，并分别在k个计算单元中通过SGD来训练更新参数，最后，通过对k个计算单元中参数的聚合来更新总体的参数并进行下一轮的迭代训练。类似的，我们基于算法4-1，设计了针对于基于选择模型的机票个性化推荐算法的并行化训练方法（见算法5-2）。

在图5-5中，我们给出了基于选择模型的机票个性化推荐算法在Spark并行计算框架上的实现。与图5-4类似，图中每个方框代表着一个抽象的Spark分布算子。

首先，我们通过Map1和Reduce1来计算得到一段时间范围内的模拟搜索结果的数据。Map1 从订单数据中提取时间和航线的信息，输出元组形式为<time, dcity, acity, ticket>的RDD，Reduce1通过时间和航线的key，即<time, dcity, acity>，得到航线上相同时间段的所有用户订单，去重后便可得到该时间段的模拟搜索结果。这时，我们再通过一个Join算子联合map1的RDD和Reduce1的RDD，从而在元组中得到每个用户订单和其对应搜索结果，<ticket, search results>。

接着，我们可以根据算法5-2来进行并行的SGD参数训练。Broadcast首先将初始化的参数广播到集群中的每个计算节点；Map2将根据计算节点的数量对用户历史订单集合进行随机的划分，并分配对应的划分标识n作为键值，同时Map2还会对各个用户历史机票对应的模拟搜索结果进行bootstrap抽样，作为该次迭代的训练样本；Reduce2将收集具有相同划分标识n的训练样例，并进行独立并行的SGD的训练；最后，Reduce3将聚合各个计算节点训练得到的参数。更新后的参数将重新进行广播，并开始新的一轮迭代训练。当达到停止条件时，迭代中止，并将模型参数输出作为最终训练的结果。

|  |
| --- |
| **Algorithm 5-2:**  **Parallel Paired Choice Model – L2** |
| **Input： T –** Ticket Set of Historical Flight Order of all Users  **S** – Ticket Set of search results for each historical flight order  **F** – Set of Flight Ticket Features  **U** – Set of all users  **M** – Set of all airlines  – Regularization Parameter  r – bootstrap sampling rate  – initial learning rate  Nk – number of Latent Factors  MAXN – maximum number of iterations  Nc – number of nodes for parallel computing |
| **Output： –** set of to be estimated parameters in ,  which includes parameters for different users and different airlines |
| 1. , 2. Define 3. for it = 1 to MAXN: 4. Random partition **T** to k with equal size L to each node 5. for all **Parallel do**: 6. for in : 7. Define as the user of 8. Define as the airline of 9. Define as the flight of 10. Define as the search results for in **S** 11. Set as the bootstrap sampling samples from 12. for in : 13. if == : 14. continue 15. for in according to , , : 17. End for 18. End for 19. Aggregate estimated parameters from N node:      1. Return ,if it convergences or stop conditions are reached. 3. End for 4. End for |

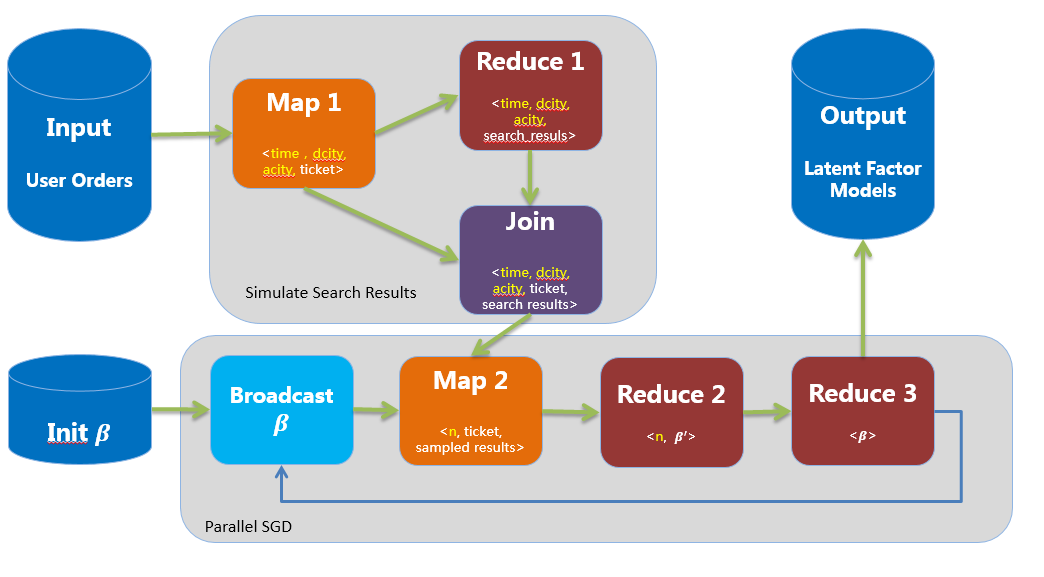


图 5-5 基于选择模型的机票个性化推荐算法的Spark并行化实现

Figure 5-5 Implementation of parallelized PCM recommendation algorithm on Spark

### 推荐算法并行化的性能测试

为了对上述并行化算法进行并行化性能测试，我们选取了数据集中约9000万的用户国内航线的订单数据放在Spark集群上进行性能测试。测试的Spark集群包含7个虚拟机节点，其中1个为Spark Master主节点，另外6个为Spark Slaves工作节点。每个节点的配置信息如表5-1所示：

表 5-1 Spark集群节点配置信息表

Table 5-1 Setup of Spark cluster

|  |  |  |
| --- | --- | --- |
| **Spark集群节点配置信息** | | |
| **硬件配置** | CPU型号 | Intel Xeon E312 |
| CPU主频 | 2 GHz |
| CPU核数 | 4 |
| 内存大小 | 7872M |
| **软件配置** | 操作系统版本 | CentOS Release 6.4 |
| Spark版本 | 1.20 |
| Hadoop版本 | Hadoop 2.2.0-cdh5.0.0-beta-2 |

图 5-6 基于用户偏好模型的机票个性化推荐算法在Spark集群上的并行化性能测试

Figure 5-6 Performance Test for parallelized UserPref recommendation algorithm on Spark

图5-6展示了基于用户偏好模型的机票个性化推荐算法对应的并行化实现（图5-4）在Spark集群上的性能测试。其中，横轴为给Spark任务所分配的计算核数，纵轴为完成该Spark任务所需要的时间。在测试过程中，由于数据量比较大，我们给每个节点统一分配了6GB的计算内存。从图中可以看出，随着核数的增加，算法运行的时间呈现出近似线性的递减。因此，该算法的并行化实现具有较好的并行性能。

对于图5-5中基于选择模型的机票个性化推荐算法的并行化实现，由于模拟机票搜索结果的步骤中我们需要非常大的内存，因此我们无法在该集群中进行全量数据集的测试。因此，我们只抽取了一小部分数据进行测试，并也获得了类似于图5-6中结果，如图5-7所示。

图 5-7 基于选择模型的机票个性化推荐算法在Spark集群上的并行化性能测试  
Figure 5-7 Performance Test for parallelized PCM recommendation algorithm on Spark

## 本章小结

本章主要结合大数据技术，介绍了面向旅游服务的大数据平台，并且提出了我们面向大数据的机票个性化推荐系统设计框架。此外，我们基于Spark并行计算框架，针对本文提出的两种机票个性化推荐算法进行了并行化的设计与实现，并且在Spark集群上进行了并行化性能的测试。结果显示，两种算法基于Spark的并行化实现都具有较好的扩展性（Scalability）和并行化效果。

# 全文总结与展望

## 本文工作总结

本文的主要工作体现在如下几个方面：

1. 本文提出了一种基于用户偏好模型机票个性化推荐算法，其通过分析和抽取用户历史机票订单数据的数据特征，并引入信息熵的概念，来计算用户在不同航线中各个机票特征上的偏好向量及其偏好权重。此外，针对用户在非活跃航线上数据稀疏的问题，提出了基于航线的协同过滤的算法，通过用户在相似航线上偏好的跨航线学习来强化用户在目标航线上的偏好。基于用户偏好模型的机票个性化推荐算法能够通过用户的历史订单数据离线地计算用户偏好，并根据其偏好能够快速高效地找出与用户历史订单中相似的机票从而为用户进行推荐。

2. 本文还提出了一种基于选择模型的机票个性化推荐算法，该方法根据选择模型，通过对用户在机票历史订单的成对选择分析，来建立用户选择机票时的效用目标函数及其优化问题。此外，还针对机票中价格波动性和价格敏感性的动态商品属性，提出了一种结合回归隐语义模型（RLFM）的效用函数模型。结合了回归隐语义模型的选择模型机票个性化推荐算法能够通过隐式空间特征，更精准地刻画出用户对不同航线中机票的偏好，尽管由于其模型的复杂性需要更长的时间进行迭代式的学习和训练，但该方法通常具有更好推荐效果。

3. 本文结合大数据技术，提出了一个通用的面向大数据的机票个性化推荐系统设计框架，包含了数据层，应用层，以及推荐逻辑层中的一个离线计算单元和在线计算单元。并且基于Spark并行计算框架，对本文提出的两种机票个性化推荐算法进行了并行化的设计与实现。

## 未来工作展望

本文对机票个性化推荐问题进行了研究和探索，分别基于用户偏好模型和选择模型，提出了两种不同的机票个性化推荐算法，并且都具有较好的推荐效果。由于时间有限，在总结现有工作的基础上，提出了一些在当前机票个性化推荐的研究中当前尚未解决的问题，以及未来的工作内容和研究方向：

1. 机票推荐的冷启动问题。针对用户在一些非活跃航线上的数据稀疏问题，本文在基于用户偏好模型的机票个性化推荐算法中，提出了一种基于航线的协同过滤的方法；在基于选择模型的算法中，我们在效用函数中增加了用户在其他航线上的公共偏好部分。然而对于在所有航线上都没有订单记录的新用户，我们并没有办法去判断用户的偏好。解决该冷启动问题的一种可行的办法是通过用户在其他旅游服务中的跨域的数据，例如酒店数据、景点门票数据等等，对用户进行跨领域的推荐。

2. 包含转程的国际机票的个性化推荐问题。由于国内机票订单大多数都是直飞航线的订单，因此，本文主要研究的是国内机票直飞航线上的个性化推荐问题。然而对于国际航线，大多数情况需要转程，这就给机票的个性化推荐增加了巨大的困难，而且用户在国际航线上的历史数据通常相对较少，数据稀疏的问题将更加明显。另外，包含转程的国际航线的搜索通常会有非常多种组合方案，相比于国内的航线具有更加明显的信息过载问题。因此，包含转程的国际航线的机票个性化推荐将是一个意义重大，但又具有很大挑战的问题。

3. 机票推荐中出行目的的预测问题。本文研究了用户在不同航线上的不同偏好，然而即使在同一航线上，用户也依然有可能因为不同的出行目的而直接影响用户对机票的购买行为。因此，研究对用户出行目的的预测，对机票推荐准确度的提升也具有重大的意义。

4. 机票推荐中共享账户的监测与识别。我们在研究和分析用户订单数据时，发现有一部分用户的历史订单中具有多个不同的乘机人，即存在着共享账户的问题。账户背后不同的用户常常具有不同的偏好，如何去分析和挖掘不同用户的偏好，并根据上下文信息来识别出共享账户背后的实际订票的用户，将能够进一步地提升推荐算法对这类共享账户的推荐效果。

参考文献

1. Google Flight. [EB/OL]. [2015-11-30]. https://[www.google.com/flights/](http://www.google.com/flights/).
2. Ctrip Flight. [EB/OL]. [2015-11-30]. <http://flights.ctrip.com/>.
3. Jannach D, Zanker M, Felfernig A, et al. Recommender Systems: An Introduction[J]. Int.j.hum.comput.interaction, 2010.
4. Resnick P, Varian H R, Resnick P, et al. Acm: special issue on recommender systems[J]. Planetary & Space Science, 1997, 48(10):945-954.
5. Ricci F, Rokach L, Shapira B. Introduction to Recommender Systems Handbook[J]. Recommender Systems Handbook, 2010, 1-35:1-35.
6. 王国霞, 刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用, 2012. 48(7): p. 66-76.
7. Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering[J]. Internet Computing IEEE, 2003, 7(1):76-80.
8. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. IEEE Computer, 2009, 42(8):30-37.
9. Coyle L. Making Personalised Flight Recommendations using Implicit Feedback[J]. University of Dublin Trinity College School of Computer Science & Statistics, 2004.
10. White T. Hadoop : the definitive guide[M]. O'Reilly, 2012.
11. 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19(01):1-15.
12. Baeza-Yates R A, Ribeiro-Neto B. Modern Information Retrieval[J]. Dept of Computer Science & Engineering Tatung Univ, 1999, 43(1):26–28.
13. Belkin N J, Croft W B. Information Filtering and Information Retrieval: Two Sides of the Same Coin?[C]. COMMUNICATIONS OF THE ACM. 1992:29-38.
14. Salton G., Automatic text processing: The transformation, analysis, and retrieval of[M]. Reading: Addison-Wesley, 1989.
15. Pazzani M, Billsus D. Learning and revising user profiles: The indentification of interesting web sites[J]. Machine Learning, 1997, 27(3):313-331.
16. Mooney R J, Roy L. Content-Based Book Recommending Using Learning for Text Categorization[C]. Proceedings of the fifth ACM conference on Digital libraries. ACM, 1999:195-204.
17. Breese J S, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[J]. Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2013, 7(7):43-52.
18. Delgado J, Ishii N. Memory-Based Weighted-Majority Prediction for Recommender Systems[C]. Research and Development in Information Retrieval. 1999.
19. Billsus D, Pazzani M J. User Modeling for Adaptive News Access[J]. User Modeling and User-Adapted Interaction, 2000, 10(2-3):147-180.
20. Hofmann T. Collaborative filtering via gaussian probabilistic latent semantic analysis.[J]. Proceedings of Workshop on Probabilistic Models for Medical Image Analysis Miccai ’, 2003, 13(4):177-184.
21. Marlin B.M. Modeling user rating profiles for collaborative filtering[J]. in Advances in neural information processing systems. 2003:627-634.
22. Zhou T, Jiang L L, Su R Q, et al. Effect of initial configuration on network-based recommendation[J]. EPL, 2008, 81(5):15-18.
23. Huang Z, Chen H. Analyzing Consumer-Product Graphs: Empirical Findings and Applications in Recommender Systems[J]. Management Science, 2007, 53(7): 1146-1164.
24. Tao Z, Jie R, Matús M, et al. Bipartite network projection and personal recommendation [J]. Physical Review E, 2007, 76(4):70-80.
25. Soboroff I M, Nicholas C K. Combining Content and Collaboration in Text Filtering[J]. Proceedings of the IJCAI’99 Workshop on Machine Learning in Information Filtering, 1999.
26. Burke R. Hybrid Web Recommender Systems[C]. The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science. 2007:377-408.
27. Burke R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted[C]. Interaction. 2002:331-370..
28. Joachims T, Freitag D, Mitchell T, et al. WebWatcher: A Tour Guide for the World Wide Web[J]. Proceedings of IJCAI, 1996, 47(9):770--775.
29. Riesbeck C K, Schank R C. Inside Case-based Reasoning[J]. Lawrence Erlbaum Associates, 1989.
30. Hu Y, Koren Y, Volinsky C. Collaborative Filtering for Implicit Feedback Datasets[C]. Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on. IEEE, 2008:263-272.
31. Jawaheer G, Szomszor M, Kostkova P. Comparison of implicit and explicit feedback from an online music recommendation service[C]. Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems. ACM, 2010:47-51.
32. Pan R, Zhou Y, Cao B, et al. One-Class Collaborative Filtering.[C]. IEEE International Conference on Data Mining. IEEE, 2008:502-511.
33. Sindhwani V, Bucak S S, Hu J, et al. One-Class Matrix Completion with Low-Density Factorizations[C]. Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE, 2010:1055-1060.
34. Pazzani M J, Billsus D. Content-based recommendation systems[C]. THE ADAPTIVE WEB: METHODS AND STRATEGIES OF WEB PERSONALIZATION. VOLUME 4321 OF LECTURE NOTES IN COMPUTER SCIENCE. 2007:325--341.
35. Chu W, Park S T. Personalized recommendation on dynamic content using predictive bilinear models[C]. Proceedings of the 18th international conference on World wide web. ACM, 2009:691-700.
36. Karatzoglou A, Baltrunas L, Shi Y. Learning to rank for recommender systems[C]. Acm Conference on Recommender Systems. ACM, 2013:493-494.
37. Koren Y, Sill J. OrdRec: an ordinal model for predicting personalized item rating distributions[C]. Fifth Acm Conference on Recommender Systems. ACM, 2011: 117-124.
38. Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian Personalized Ranking from Implicit Feedback[C]. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2012:452-461.
39. Shi Y, Karatzoglou A, Baltrunas L, et al. TFMAP: optimizing MAP for top-n context-aware recommendation[C]. Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012: 155-164.
40. Rvelin K, Kekaelaeinen, Inen J. IR evaluation methods for retrieving highly relevant documents[C]. International Acm Sigir Conference on Research & Development in Information Retrieval. ACM, 2000:41-48.
41. Gartner BigData. [EB/OL]. [2015-11-30]. <http://www.gartner.com/it-glossary/big-data>.
42. 刘智慧, 张泉灵. 大数据技术研究综述[J]. 浙江大学学报：工学版, 2014(06):957-972.
43. Keim D, Qu H, Ma K L. Big-Data Visualization[J]. IEEE Computer Graphics & Applications, 2013, 33(4):20-21.
44. King, R. 英特尔：将Hadoop“固化”到Xeon中. [EB/OL]. [2015-12-01]. <http://www.csdn.net/article/2013-02-27/2814274-intel-baking-apache-hadoop-into-xeon>.
45. The Apache Software Foundation. Apache HBase [EB/OL]. [2015-12-01]. <http://hbase.apache.org/>.
46. The Apache Software Foundation. Apache Hive. [EB/OL]. [2015-12-01]. <http://hive.apache.org/>.
47. The Apache Software Foundation. Apache Pig Latin. [EB/OL]. [2015-12-01]. <http://pig.apache.org/>.
48. The Apache Software Foundation. Apache Mahout. [EB/OL]. [2015-12-01]. <http://mahout.apache.org/>.
49. The Apache Software Foundation. Apache Zookeeper. [EB/OL]. [2015-12-01]. <http://zookeeper.apache.org/>.
50. The Apache Software Foundation. Apache Sqoop. [EB/OL]. [2015-12-01]. <http://sqoop.apache.org/>.
51. The Apache Software Foundation. Apache Flume. [EB/OL]. [2015-12-01]. <http://flume.apache.org/>.
52. The Apache Software Foundation. Apache Spark. [EB/OL]. [2015-12-01]. <http://spark.apache.org/>.
53. 罗成奎, 大数据技术在智慧旅游中的应用 [J]. 旅游纵览 (下半月), 2013. 8: p. 59-60.
54. Safavian S R, Landgrebe D. A survey of decision tree classifier methodology[J]. Systems Man & Cybernetics IEEE Transactions on, 1991, 21(3):660 - 674.
55. Quionero-Candela J., et al., Dataset shift in machine learning[M]. 2009: The MIT Press.
56. Jiang J, Zhai C X. Instance Weighting for Domain Adaptation in NLP[J]. Acl, 2007:264--271.
57. Pan S J, Yang Q. A Survey on Transfer Learning[J]. IEEE Transactions on Knowledge & Data Engineering, 2010, 22(10):1345-1359.
58. Von Neumann J and Morgenstern O, Theory of games and economic behavior[M]. 2007: Princeton university press.
59. Thurstone L L. A Law of Comparative Judgment.[J]. Psychological Review, 1994, 101(4):273-86.
60. Train K E. Discrete Choice Methods with Simulation[M]. Cambridge University Press, 2009.
61. Agarwal D, Chen B C. Regression-based latent factor models[C]. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009:19-28.
62. Carpenter B. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression[J]. Alias-i, Inc., Tech. Rep, 2008: p. 1-20.
63. Tsuruoka, Yoshimasa, Tsujii, Jun'ichi, Ananiadou, Sophia. Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty[J]. ACL-IJCNLP 2009, 2009:477-485.
64. Li L. Parallelized Stochastic Gradient Descent.[J]. Advances in Neural Information Processing Systems, 2010:2595-2603.

致谢

值此论文完成之际，谨向研究生期间所有支持、关心和帮助过我的人表示诚挚的感谢。

首先，感谢我的硕士导师曹健教授。在硕士的学习和研究期间，曹老师给了我非常大的帮助和支持。在曹老师的悉心指导下，我完成了本文的主要研究内容。曹老师严谨认真的治学态度，精益求精的工作作风，诲人不倦的高尚师德令我受益匪浅。曹老师教会了我对科学、对工作、对生活的积极向上的态度也将成为我未来人生道路上的宝贵财富，令我受益终身。

此外，感谢实验室长时间以来一起学习、成长的同学们。感谢胡亮博士，在课题研究过程中给了我很大的帮助和支持。在与陈誉文、赵亚峰、华逸群等同学一起合作参与的科研项目中，我们互帮互助，共同取得了一定的科研成果，同时也结下了深厚的友谊。此外，还要感谢在CIT实验室里所有曾给予给我帮助的同学们。我们在CIT这样的一个大家庭中共同成长，会是永远的朋友。

最后，衷心地感谢我的父母和我的朋友们，他们对我的关心和支持一直伴随着我的成长，他们对我的期望和鼓励更是我不断前进的动力。

附录一符号与标记

|  |  |
| --- | --- |
|  | 偏微分运算 |
|  | 取极小值 |
|  | 取极大值 |
|  | 取下确界  点乘 |
|  | 连加  连乘 |
| |**A**| | 集合**A**中元素的个数 |
| # | 数量 |
|  | 表示用户偏好关系，前者大于后者 |

附录二英文缩略语表

|  |  |  |
| --- | --- | --- |
| AUC | Area Under Curve | 曲线下面积 |
| BPR | Bayesian Personalized Ranking | 贝叶斯个性化排序算法 |
| CBR | Case-Based Reasoning | 案例推理 |
| HDFS | Hadoop Distributed File System | Hadoop分布式文件系统 |
| OTA | Online Travel Agent | 在线旅游代理 |
| PCM | Pairwise Choice Modeling | 成对选择建模 |
| KNN | K-Nearest Neighbor | K邻近算法 |
| LFM | Latent Factor Model | 隐语义模型 |
| LtR | Learning to Rank | 学习排序 |
| MAP | Mean Average Precision | 平均准确率均值 |
| MLE | Maximum Likelihood Estimation | 最大似然估计 |
| MSE | Mean Square Error | 均方误差 |
| OCCF | One-class Collaborative Filtering | 单类协同过滤 |
| RLFM | Regression-based Latent Factor Model | 基于回归的隐语义模型 |
| SGD | Stochastic Gradient Descent | 随机梯度下降 |

攻读硕士学位期间已发表的论文

**一已发表、录用的论文：**

1. Fangzhou Yang, Jian Cao and Dragan Milosevic. An Evolutionary Algorithm for Column Family Schema Optimization in HBase. IEEE International Conference on Big Data Computing Service and Applications (BigDataService), Page(s):439-445, 2015.（第一作者）