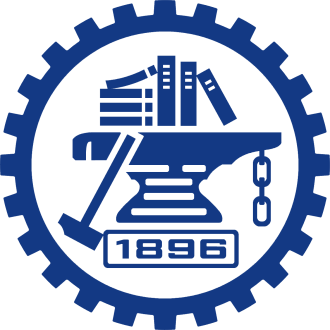
申请上海交通大学硕士学位[论文](#_OFDM技术简介_1)

基于强化学习的机票动态定价策略研究

****

**学校：** 上海交通大学

**院系：** 电子信息与电气工程学院

**班级：** B1703391

**学号：** 11703910026

**硕士生：** 刘泽霖

**专业：** 计算机科学与技术

**导师：** 曹健

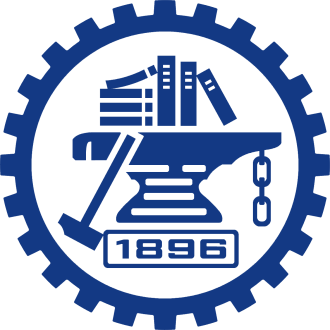
**上海交通大学电子信息与电气工程学院**

**2020年1月**

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

**Degree of Master**

Airline Tickets Dynamic Pricing Based on Reinforcement Learning

****

**Author:** Zelin Liu

**Specialty:** Computer Science

**Advisor:** Jian Cao

School of Electronic Information and Electrical Engineering

Shanghai Jiao Tong University

Shanghai, P.R.China

January, 2020

**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《机票价格预测系统的设计与实现》，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

**基于强化学习的机票动态定价策略研究**

摘要

随着互联网时代和现有交通体系的快速发展，

**关键词**：动态定价

Personalized Route Recommendation Based on Data Analysis

ABSTRACT

Keywords:

目录

[摘要 I](#_Toc469747835)

[ABSTRACT II](#_Toc469747836)

[第一章 绪论 1](#_Toc469747837)

[1.1 研究背景及研究意义 1](#_Toc469747838)

[1.2 国内外研究现状 3](#_Toc469747839)

[1.2.1 路径搜索的研究现状 3](#_Toc469747840)

[1.2.2 路径推荐的研究现状 4](#_Toc469747841)

[1.2.3 既有内容总结 5](#_Toc469747842)

[1.3 本文研究内容和结构安排 5](#_Toc469747843)

[1.3.1 研究内容 5](#_Toc469747844)

[1.3.2 文章结构安排 6](#_Toc469747845)

[第二章 相关技术 9](#_Toc469747846)

[2.1 路径规划 9](#_Toc469747847)

[2.2 推荐系统 11](#_Toc469747848)

[2.2.1 推荐系统概况 11](#_Toc469747849)

[2.2.2 常用相似度的计算方法 13](#_Toc469747850)

[2.2.3 常用评价指标 15](#_Toc469747851)

[2.3 缓存策略 16](#_Toc469747852)

[2.4 本章小结 17](#_Toc469747853)

[第三章 用户购买决策模型 19](#_Toc469747854)

[3.1 引言 19](#_Toc469747855)

[3.2 个性化路径推荐模型 20](#_Toc469747856)

[3.3 本章小结 23](#_Toc469747857)

[第四章 带约束的top-K路径搜索算法 25](#_Toc469747858)

[4.1 引言 25](#_Toc469747859)

[4.2 路径搜索算法KSPG 25](#_Toc469747860)

[4.2.1 路径搜索问题定义 25](#_Toc469747861)

[4.2.2 KSPG算法 26](#_Toc469747862)

[4.3 可到达距离计算方法 32](#_Toc469747863)

[4.4 实验结果分析 32](#_Toc469747864)

[4.4.1 实验准备 33](#_Toc469747865)

[4.4.2 实验结果与分析 35](#_Toc469747866)

[4.5 本章小结 36](#_Toc469747867)

[第五章 路径推荐算法 37](#_Toc469747868)

[5.1 引言 37](#_Toc469747869)

[5.2 特征预处理 37](#_Toc469747870)

[5.3 基于信息熵的路径推荐算法 38](#_Toc469747871)

[5.3.1 路径相似度计算 39](#_Toc469747872)

[5.3.2 基于信息熵的路径推荐算法 41](#_Toc469747873)

[5.4 实验结果分析 44](#_Toc469747874)

[5.4.1 实验数据和相关设置 44](#_Toc469747875)

[5.4.2 推荐评价指标 49](#_Toc469747876)

[5.4.3 结果分析 49](#_Toc469747877)

[5.5 本章小结 54](#_Toc469747878)

[第六章 路径搜索的缓存策略 55](#_Toc469747879)

[6.1 行程搜索数据分析 55](#_Toc469747880)

[6.2 行程缓存模块 56](#_Toc469747881)

[6.3 路径缓存模块 57](#_Toc469747882)

[6.3.1 路径缓存策略 58](#_Toc469747883)

[6.3.2 路径缓存结果分析 59](#_Toc469747884)

[6.4 路径缓存更新策略 60](#_Toc469747885)

[6.4.1 路径缓存更新策略 60](#_Toc469747886)

[6.4.2 缓存路径更新策略结果分析 62](#_Toc469747887)

[6.5 本章小结 63](#_Toc469747888)

[第七章 全文总结与展望 65](#_Toc469747889)

[7.1 本文工作总结 65](#_Toc469747890)

[7.2 未来工作展望 65](#_Toc469747891)

[参考文献 67](#_Toc469747892)

[致谢 71](#_Toc469747893)

[附录一 符号与标记 72](#_Toc469747894)

[附录二 英文缩略语表 73](#_Toc469747895)

[攻读硕士学位期间已发表的论文 74](#_Toc469747896)

1. 绪论
   1. 研究背景及研究意义

随着时代和科技的发展，交通为人类出行提供了安全、便捷的服务。现有的交通体系包括以公路、铁路、城市轨道地铁为主的陆地交通，海洋江河的水上交通，航空为主的空中交通等。同时，中国高铁的蓬勃发展，打造了“八横八纵”的高速铁路网络，使得整个交通运输网络规模日益庞大，结构更加复杂，出行更加便捷。交通已经成为人类生活、社会生产和经济发展中的必要工具。随着现代多样化交通工具逐渐成熟，国内OTA（Online Travel Agent）的迅速发展，以携程旅游[[[1]](#endnote-2)]、去哪儿网[[[2]](#endnote-3)]、Triplehop’s TripMatcher[[[3]](#endnote-4)]等为代表的在线搜索引擎，为乘客提供了越来越多的出行信息和高质量的出行服务。乘客可以自行在网页上搜索城际间的出行信息，筛选满足需求的出行工具，并完成最后的支付与交易。相比于传统的线下预定购买（例如：火车机票售票点）， OTA为乘客提供了全面、实时的信息，以及便捷查询、交易平台。

不同于传统的、价格稳定的出行工具（例如：火车、汽车），机票的价格随着时间非周期性频繁波动，各航空公司试图通过私有的、复杂的定价系统，动态调整机票价格从而达到收益最大的目标。同时，丰富的交通工具选择，热门城市间铁路班次、航班也越来越多，例如上海到杭州一天共有170趟火车可供选择；一条热门航线直达的航班搜索结果可能有几十，甚至上百条，若考虑中转，则可达成百上千种出行路径可供选择，同时每个航班包含了价格、出行时间、航司、退改签信息等大量特征属性，造成了严重的信息过载问题，。而且，由于乘客选择出行工具的复杂心理，在实际应用中，乘客通常会希望出行方案满足一些约束条件，例如换乘次数、出行时间、价格预算等，所以大多数情况下单一的联程路径往往不能很好的满足顾客的需求。

如今乘客面对多种交通工具、中转城市和换乘方案，OTA提高服务质量的关键问题在于如何快速的选择最优且满足个人需求的城际出行方案。随着生活智能化的推进，基于大数据分析的推荐技术研究也应运而生，通过对乘客的历史数据分析，有效的挖掘乘客兴趣偏好和需求，并推荐最可能感兴趣的路径方案。推荐可以帮助乘客在众多的搜索结果中快速筛选可能感兴趣的路径，减少乘客的选择时间，且OTA服务提供商也可以因此潜在的提高转化率。而路径推荐针对个人提供个性化的服务，使得人们能够及时安排规划行程，交通资源得以合理的分配，交通安全得到大幅度提高，为人们的城市生活提供更好的服务。

个性化推荐[[[4]](#endnote-5)][[[5]](#endnote-6)]作为当今大数据领域重要的发展方向之一，已经被广泛的研究，并大量应用到工业实践中。大数据技术可以有效的从海量数据中挖掘有价值的信息，实现数据模型与业务的有效结合，例如Amazon[[[6]](#endnote-7)]的书籍推荐、MovieLens [[[7]](#endnote-8)]的电影推荐。传统的推荐中，物品是独立存在的，具有固定的属性，而出行路径并不是单一物品的推荐，而是对于多种不同交通工具路径组合的推荐，交通工具是受时间影响的动态商品，价格、时间是实时变化的，路径也是动态变化的，因此传统的静态物品推荐算法并不能很好的直接应用到出行路径及其他一系列时间敏感商品的个性化推荐中。在个性化路径推荐问题中，复杂性具体表现在以下几个方面：

* **波动性：**路径是时间敏感的商品，某些特征会在短时间内发生明显波动，例如：各航空公司试图通过私有的、复杂的定价系统，动态调整机票价格，造成价格随时间非周期性的波动，即使同一个航班，在同一天内价格也会稍许波动，而出发日期的不同更会造成较大的价格差。
* **多样性：**交通工具的多样性使得不同航班、高铁车次、汽车等组成的出行路径可达成百上千种；而且每条路径拥有多种特征，出发和到达时间、价格、航司、换乘数、退改签政策、机型等。多样性增加了路径查找的难度，使得乘客的选择更加复杂。
* **特征敏感性：**乘客在选择出行方案时，路径的波动性会直接影响最终的购买结果。不同乘客的购买决策因素也不相同，例如：价格敏感、时间敏感等。乘客总是倾向于选择性价比高，且满足自己要求的出行路径。

基于以上路径特征，推荐问题在此场景下变得十分困难，但同也说明了研究此问题的重要研究意义和价值。在学术领域，针对城际间出行路径的个性化搜索、推荐研究并不是很多，已知的大多数方法主要集中在旅游的行程安排、换乘点的选取等问题上，而且各问题的研究均独立进行，也没有将路径的推荐、搜索算法等直接应用到实际服务中的案例。

同时，近些年来，大数据技术迅速发展，并渗入到金融、生活、出行、医疗等众多生活领域中，人工智能、机器学习也已经成为IT的热门发展方向。将数据分析在路径搜索中具体化，完善城市间的交通体系，构建个性化的路径推荐系统，并优化潜在的路径搜索、推荐算法，高效搜索符合乘客需求偏好的出行路径，是为乘客提供高质量服务的前提保证。

本文设计了一个完整的城际间个性化路径推荐服务模型，在路径搜索时，通过分析乘客的历史订单数据特征，根据推荐算法对路径排序，并在搜索及推荐的基础上，加入缓存机制，提高搜索效率。所以综上所述，基于大数据的相关技术，研究个性化路径搜索与推荐，无论在学术领域，还是行业应用中，都有着十分重要的意义。

* 1. 国内外研究现状
     1. 路径搜索的研究现状

在交通领域，路径搜索与规划问题被广泛研究，针对本文的研究课题，首先从路径规划方面介绍研究现状。

路径规划最早可追溯到Hassler Whitney提出的TSP旅行商问题[[[8]](#endnote-9)]，目标是找出满足约束条件的最短路径，使得旅行商从一点出发，拜访所有城市回到原点，且每个城市限制只能拜访一次。目前，路径规划已被广泛应用于诸多领域，在科技领域包括：无人机动态航迹规划、机器人自主避障、导弹拦截等；在生活领域中主要包括交通和旅游两方面，涵盖GPS导航、最优驾驶路径优化、物流配送等。传统的路径规划问题通过已知路径信息，并根据乘客提供的出行需求，快速搜索起始起点和目标终点间的路径，并返回可行路径，总体大致可分为两个环节：

1. 路径网络建模：该环节的目标是建立路径规划模型，根据环境信息构建路径网络，将实际应用场景转换为抽象的符号模型，并进行求解；
2. 路径规划：在网络模型的基础上，针对模型中的目标函数，应用算法对可行路径进行搜索，以达到目标函数最优；

在交通网络中，文献[[[9]](#endnote-10)]建立了多模式旅游模型（MTS），模型中乘客可以选择不同的交通工具出行，包括铁路、汽车、地铁等。将换乘节点和路段抽象为城际网络，对乘客给定的源-终点（O-D）、时间和换乘次数等约束条件，通过划分子网进行最短路径搜索，最终通过GIS系统等平台展示路径搜索结果。在文献[[[10]](#endnote-11)][[[11]](#endnote-12)]中交通网络都固定不变，但每条边的权重用随机变量表示，权重大小随时间变化，其中文献[10]中路径规划的目标是找到权重和期望值最小的路径，算法共分为两步：定义时间周期，在每个周期中计算所有点到终点的最短路径；基于以上最短路径得到路径权重的下限值；而文献[11]是在给定出发时间的情况下查找最优路径，该问题是NP难问题，故使用近似算法在保证准确率的前提下，可以在伪多项式时间求解问题。陈佳丽等人[[[12]](#endnote-13)]将公交网络中空间位置相近的公交站合并，并视为交通网络中的节点，而节点间边的权重表示为路段的长度和道路拥堵的情况，在路径选择时以换乘次数最小为目标，尽可能缩短出行距离，并在可接受的步行距离范围内，减少换乘次数。

多式联运是一种至少由两种运输方式组合而成的现代运输方式，目的是为了充分利用不同运输方式优点，同时降低成本。相关的研究问题包括转运节点、路径的选择等。多式联运的路径中，影响的因素包括运输方式、成本等，而建模的目标函数主要为四个方面：最小化运输成本、最短运输时间、最短运输距离及最少运输车辆。Reddy等人[[[13]](#endnote-14)]第一次提出以最小化运输成本为目标，求解整数规划问题，得到成本最低路径的中转点，其中运输成本主要包括运输费用和转运费用。文献[[[14]](#endnote-15)]针对最优运输组合方式进行研究，首先预设某些路段、节点间的运输方式，再计算到固定接入点的距离，从而将路径搜索问题缩小至一个较小的网络中，反复计算直至找到最短路径以及其他路段的运输方式。刘舰等人[[[15]](#endnote-16)]不仅考虑了运输成本和时间，还加入了运输风险指数，目标是将运输成本和风险最小化，求解过程中将多目标转换为单一目标，并使用人工免疫算法求解。目前国内外大多数研究把最小化运输成本作为路径规划的目标，部分将时间转化为成本，或者作为目标函数的惩罚因子，但无论是运输工具的选择还是联运路径规划，多式联运主要的运输对象是货物。

国内外路径规划问题的研究还有很多，网络建模、路径选择算法与优化等许多研究思想都可以借鉴，但是在传统的路径规划中，多数模型设定单/多目标函数进行求解，但建模时未考虑出行交通工具特点和服务质量，也没有考虑出行乘客的偏好、舒适度等因素。

* + 1. 路径推荐的研究现状

个性化推荐[4][5]作为解决信息超载的有效解决方法，被广泛应用于学术研究和工业领域中，尤其在电子商务领域获得了巨大的商业利润。推荐通过分析用户的需求和偏好兴趣，对信息进行过滤，并返回用户感兴趣的物品。

路径推荐的场景主要包括交通和旅游。在交通中，国内外研究集中在路径导航和出租车路径推荐方面。研究中大多以汽车GPS轨迹数据为基础，在优化驾驶时间的同时，最大化驾驶路径产生的效益及与用户驾驶偏好的相似程度。文献[[[16]](#endnote-17)]中的T-driver系统对行车路径进行规划，结合实际的交通、天气情况，为不同的驾驶员返回不同的驾驶路径，但在路径搜索中只考虑了时间因素。文献[[[17]](#endnote-18)]中使用高斯混合模型从GPS轨迹数据中挖掘用户的驾驶偏好，包括时间、转向等驾驶习惯，同时定义满意度评分函数，为用户推荐局部最优和全局最优的行车路径。文献[[[18]](#endnote-19)]比较了不同用户的驾驶习惯差异，将行驶轨迹与用户的偏好相联系，提出一种基于行驶轨迹聚类的推荐算法，分析历史轨迹中经过的行驶点并进行单层聚类分析，建立轨迹模型，并根据用户车辆的位置推荐可行路径。文献[[[19]](#endnote-20)]通过预估概率最大的载客点，推荐出租车司机行驶路径，减少空载时间，并从乘客角度出发，推荐更容易打车的地点。

游客在规划行程时需要收集、整合大量的信息，而在现有的旅游推荐中，根据收集的标签、照片、社交数据，可以准确、高效的为游客推荐旅游景点和观光路径，提供便捷的旅游服务。Y.Huang等人[[[20]](#endnote-21)]在贝叶斯网络的基础上，结合基于内容的协同过滤算法，为游客推荐感兴趣的旅游景点。文献[[[21]](#endnote-22)][[[22]](#endnote-23)][[[23]](#endnote-24)]主要研究时间跨度较长的行程安排，例如多个景点间的参观顺序，以最大化用户满意度为目标，提供个性化的旅游路径推荐；而文献[[[24]](#endnote-25)][[[25]](#endnote-26)]中研究基于室内无线定位装置（例如：RFID）收集历史乘客的观光数据，并对景点内部的游览路径进行个性化推荐，使用的场景包括主题公园、博物馆等。

* + 1. 既有内容总结

在交通、旅游产业高速发展的今天，在路径规划的同时，如何提高乘客出行满意度是各大服务提供商亟待解决的问题。而以上国内外研究中，路径规划、多式联运运输和路径推荐算法研究都有一定的局限性，并不能很好的解决城际间个性化路径推荐问题，主要表现在三个方面：

1. 路径网络：传统的交通、旅游路径规划问题中，网络整体变化很小，但城际间的出行路径是多种交通工具的组合，每种交通工具都会具体的班次、车次或者航班等信息，此类信息是随时间动态变化的，所以构建基础的城际间出行路径网络是文章研究的基础。
2. 推荐算法：乘客自身很难刻画对推荐路径的满意程度，只能从隐式反馈的数据中挖掘，例如：浏览记录、历史订单等。路径规划和多式联运的对象不是人，不同乘客在选择路径时偏好不相同，因此需要一种个性化的路径推荐方法，保证乘客的满意度。
3. 缓存策略：路径搜索中在满足乘客偏好的同时，加入路径的缓存策略，尽可能减少路径的搜索时间，提高整体的服务质量
   1. 本文研究内容和结构安排
      1. 研究内容

乘客的购买行为是由一系列决策过程构成的，由多方面因素共同影响，乘客根据自身需要根据当前的交通状态，做出判断选择的过程，主要的影响因素为：1）乘客的偏好特征，例如换乘次数、出发时间、价格等，也包括经验判断等；2）交通状态，出行时的交通信息直接影响了路径的特征。所以乘客的购买是多因素决策过程，具有很大的随机性。

基于已有研究，针对城际间的个性化路径推荐问题，考虑多种交通换乘方式、乘客出行需求和隐式特征偏好，提出一种城际间个性化路径推荐服务模型，根据历史订单数据分析乘客偏好，搜索乘客满意度最高的最优路径，为出行提供决策支持。文章的主要工作致力于以下三个方面：

1. 路径搜索：构建城际间交通换乘网络，提出一种带约束的启发式路径搜索算法，在路径搜索前，由乘客给定出发到达城市、出发时间区间，且乘客可以提出一些约束条件，例如总耗时、价格区间等，在保证效率的同时，搜索前条满足约束的最优出行路径。
2. 路径推荐：路径是随时间变化的动态商品，所以选择合理的特征刻画乘客偏好，并根据乘客的历史订单记录计算乘客偏好，同时针对乘客历史订单稀疏性的问题，定义乘客相似度计算方法，找出top-n相似乘客，利用相似乘客的所有历史订单，提出一种基于信息熵的乘客偏好模型推荐算法，计算路径的推荐排序。
3. 路径缓存：由于出发-到达城市、中转车站较多，搜索效率降低，所以缓存机制目的是在保证乘客满意度的情况下，降低路径搜索时间。基于查询搜索数据和乘客的订单数据，根据搜索频率、推荐度制定行程、路径的缓存策略，以及缓存路径的更新策略，提高整体的搜索效率。
   * 1. 文章结构安排

本文共有七个章节，各个章节的结构与内容如下：

第一章绪论，首先简单阐述了文章的研究背景和课题的研究意义，并介绍了城际间个性化出行路径推荐问题，及需要面临的挑战。通过总结国内外目前对路径规划、多式联运问题及推荐系统的已有研究成果，明确个性化出行路径推荐的必要性。最后，概括性的描述文章主要的研究内容和整体的结构安排。

第二章相关技术，首先总结归纳已有的最短路径算法，并对路径规划问题做了详细介绍，同时分类介绍目前已有的推荐算法和相似度计算方法，总结推荐算法的评价指标。最后列举了几种目前常用的缓存策略，并对比不同策略的优缺点。

第三章个性化推荐模型，首先介绍基于数据分析的个性化路径推荐模型，其中包括与乘客的交互，路径搜索、推荐和缓存四个部分及其之间的相互联系。基于以上四部分，详细介绍了路径推荐的整体交互过程和推荐流程，同时描述了每个部分需要解决的问题、使用到的数据集，且明确个性化推荐模型的使用场景，为后文进一步对路径搜索、推荐算法和缓存策略的评估、实验分析奠定基础。

第四章带约束的top-K路径搜索算法，本章提出的一种路径启发式搜索算法KSPG，在路径搜索前，由乘客给定出发到达城市、出发时间区间，且乘客可以提出一些约束条件，例如总耗时、价格区间等，根据实时价格信息，构建交通换乘网络，KSPG算法按照路径权重由小到大依次生成前条满足约束的最短路径，并作为推荐的备选路径。最后，基于真实的实时机票、高铁价格数据，评估算法的搜索效率。

第五章路径推荐算法，合理刻画乘客、路径的特征，基于信息熵计算乘客在不同特征上的偏好权重，定义路径相似度计算方法，提出一种基于信息熵的路径推荐算法；对于乘客历史订单稀疏问题，定义用户相似度计算方法，利用相似乘客的历史订单完成路径排序。最后根据真实的机票、高铁历史订单数据，对个性化路径推荐算法进行实验评估，比较分析推荐结果。

第六章路径缓存策略，共研究三个问题：行程缓存替换策略，中转路径缓存策略，及缓存路径的更新策略。基于路径的搜索频率、推荐程度和抽样等概念，在保证顾客满意程度的情况下，指定不同的替换、更新策略，有效的提高路径搜索效率，减少路径搜索时间。

第七章全文总结与展望，首先回顾、总结主要研究工作，并针对研究中存在的不足，提出未来进一步研究的方向和重点。

1. 相关技术
   1. 路径规划

路径规划算法是路径规划的核心，其中常用的算法有求解带权重的有向图最短路径、整数规划、动态规划等，另外基于生物仿生学，启发式算法也被很好的运用到复杂动态环境的路径规划问题中。

最短路径是图论中的经典问题，也是大多数路径规划研究问题的核心，主要可分为两大类：1)狭义最短路径：对于源点和终点已确定的情况，在目标网络中找出代价最小的路径，经典的算法有Dijkstra算法、Floyd算法、Auction算法，目前依然有许多研究者关注[[[26]](#endnote-27)][[[27]](#endnote-28)]，研究方向主要集中在动态网络、并行搜索等，目的是提高搜索效率；2)广义最短路径：即K最短路径问题，在源点和终点间搜索前K条最短路径（KSP），在通信、运输和物流等诸多领域广泛研究[[[28]](#endnote-29)][[[29]](#endnote-30)]，下面简单介绍几种经典的最短路径计算方法。

* Bellman-Ford算法

Bellman-Ford算法[[[30]](#endnote-31)]是解决单源最短路径的一种常用算法，基本原理是对原始图进行次松弛，每次松弛访问相邻节点，得到所有可能最短路径解，从中选择代价最小的作为最短路径解，算法具体过程如下：

1. 假设源点为，数组大小为，存储各节点到的最短距离，初始化数组，如果没有直接相连的边，则；
2. 建立路径数组，记录每个节点在最短路径上的父节点，初始化为NULL；
3. 松弛操作：，如果，则更新，且；
4. 重复松弛操作，直至数组不再改变退出循环，最多重复次；
5. 最后第次松弛操作中，如果，则图中存在负权环。

经典Dijkstra算法是从深度优先角度搜索最短路径，而Bellman-Ford算法是广度搜索，且图中权值可以负数，但是算法复杂度太高，为，因此改进后得到一些最短路径的优化算法，例如SPFA[[[31]](#endnote-32)]。

* 算法

算法是静态网络的一种启发式最短路径搜索算法，在给定的网络空间中，评估每一个搜索位置，并选择最佳位置进行下一步搜索，直至到达目标。在算法中，虽然一直选择最佳节点进行搜索，但是并没有舍弃其他节点，而是在每一次的评估选择时，把当前节点和以前的节点共同评估，从中选择当前的最佳节点进行下一步的搜索，所以一定能求得最优解。

在启发式搜索中，位置评估变得十分重要，评估函数通常表示为，其中表示源点到节点的最短路径值，表示节点到目标节点的启发距离值。从计算中可知，是已知的，而是未知的，如果，则没有任何启发信息，需要计算所有节点到源点的最短距离，所以只要小于节点到目标节点的最短路径，则可以求得最优路径。启发距离决定了搜索效率，越小，路径搜索效率越低，计算最优路径时需要搜索更多的节点。目前算法在游戏的路径搜索中已被广泛使用。

* Yen算法

Yen算法[[[32]](#endnote-33)]是解决无环图中KSP问题的经典算法，采用偏离路径的思想搜索任意两点间的K条最短路径。假设从源点到目标点存在两条路径、，两条路径中一定存在同一节点，满足、中到的路径完全相同，但从节点开始、的路径不相同，则是的偏离路径，且是偏离节点。Yen算法具体过程如下：

1. 初始化路径列表：采用经典的最短路径算法（Dijkstra、Bellman-Ford等），计算源点到目标节点的最短路径，添加到路径列表中，假设目前列表中已有条路径，计算；
2. 计算偏离节点：如果，则路径的第二个节点为偏离节点；如果，首先将路径偏离节点的前一节点，到目标节点前一节点的所有中间节点组成偏离节点集合；
3. 假设偏离节点为，则，计算到目标节点的最短路径，替换中路径，并与拼接构成路径的一条候选路径；
4. 计算偏离节点集合的所有候选路径，从中选择最短路径作为，添加到路径列表中。重复以上过程，直至得到条最短路径。

目前，基于Yen算法也提出了多种优化算法[[[33]](#endnote-34)][[[34]](#endnote-35)]，但由于路径扩展时，都需要计算偏离点到目标节点的最短路径，所以Yen算法运行时间较长。

* 标号算法

标号算法是求解一般KSP问题的方法之一，共分为两类：标号纠正、标号设置。其中标号纠正算法最早起源于Double-Sweep算法，分为向前和向后扫描两部分：1）向前扫描：对每个节点，下标从小到大进行搜索，找到节点的前一节点，如果从源点到的最短路径经过，则重新计算预估值，并被下一次迭代使用；2）向后扫描：与向前扫描相似，节点下标从大到小搜索，检查与节点相邻的后一个节点。多次迭代直至节点标号值不改变，则完成路径搜索。该算法最坏情况下需要次扫描，时间复杂度为。

目前路径规划问题除了求解最短路径外，还可以使用优化模型进行求解，包括整数规划为代表的精确算法和蚁群、粒子群算法为代表的启发式算法。文献[[[35]](#endnote-36)]对城际多方式出行的路径规划问题进行求解，构建多方式出行网络，并在网络内划分核心区域，基于时间、费用、换乘、舒适度四个方面在区域核心内进行路径规划和搜索，提出广义费用最小的出行路径优化模型。目前国内外在城际间出行路径推荐部分研究较少，而在大多数的研究中也并未考虑路径和用户偏好的匹配程度。除此之外，文献[[[36]](#endnote-37)]中提出使用skyline查询方法求解路径规划问题，将路径网络抽象为有向图，每条边有多种属性值，并在搜索最短路径时加入2个剪枝策略和评估函数，但当网络扩大时，搜索时间和空间仍将迅速增加。

* 1. 推荐系统
     1. 推荐系统概况

Resnik和Varian[[[37]](#endnote-38)]在1997年第一次提出推荐系统概念定义，如今推荐系统已经成为科研及互联网领域的研究热点，是解决信息过载问题的有效方法。在个性化推荐中，由系统挖掘用户的兴趣偏好，根据用户的需求和兴趣，推送用户感兴趣的产品和信息。推荐系统不仅提供了高质量的推荐服务，与用户间也建立了互利共赢的关系。推荐系统包括用户建模、物品建模和推荐算法三部分，而数据源是推荐系统的基础，是获取用户信息和推荐的关键，一般输入数据包括以下四种：1)用户的自然属性和社会属性，包括年龄、职业、收入等；2)用户交互式输入，包括主动填写的信息（例如搜索数据、感兴趣的类别）和用户对推荐物品的满意度（例如评分、调查问卷）；3)用户浏览、搜索和购买行为数据，通过浏览内容、搜索购买频率等可以很好的反应用户的需求和兴趣偏好；4)推荐物品信息，推荐对象决定了推荐模型的选择，商品通常关心用户对不同商品的评分，而网页一般评估内容与用户兴趣的相似关系。而目前推荐系统可大致分为两类：交互式推荐系统、自动推荐系统。

* 交互式推荐系统中用户参与度较高，可以和系统实时交互，并根据自身偏好对推荐的路径、行程进行选择和打分。文献[[[38]](#endnote-39)]中构建了一个交互式系统TripTip，基于之前选择的景点标注用户偏好，根据偏好推荐给用户下一个旅行景点。此类系统中用户的信息输入是主动的，称为显示反馈，例如：用户对浏览的物品、电影进行打分。此种方式的优点是系统可以更简单、直接的了解用户的偏好和需求；而缺点是实效性太高，现实中很少用户愿意反馈自己的喜好，而且用户耐心是有限的。
* 自动推荐系统根据各类数据，例如：购买、浏览、书签标记数据，追踪用户行为，评估用户偏好，并根据推荐算法生成推荐的路径。文献[[[39]](#endnote-40)]基于贝叶斯网络和层次分析方法，建立一个旅游推荐系统，系统集成各类异构的在线旅游信息，为用户在陌生城市推荐旅游景点，并收集物理信息作为用户对推荐景点的反馈，无需任何的用户交互行为。此类系统中用户的交互较少，只能从行为数据中挖掘用户的偏好预测用户感兴趣的商品，称为隐式反馈。此种方式的优点是不会干扰用户的正常操作，减少用户的负担；但缺点是不能直接了解用户对物品的喜好程度，模糊的信息数据也不一定可以直接反映用户的需求和特征偏好。

无论是交互式还是自动推荐系统，推荐算法一定是推荐系统的核心，通过对近年来的研究和总结，大体可分为四类：基于内容的推荐、协同过滤、基于网络的推荐、混合推荐。

* 基于内容的推荐根据物品本身抽取特征信息表示物品，并将用户兴趣特征与物品特征进行比较，寻找具有相似特征的物品作为推荐结果，其中物品特征与用户特征相似度是推荐策略中的关键。 第一小节中提到，路径是实时变化的动态物品，文献[[[40]](#endnote-41)]中提出一种航班机票的推荐算法，机票的价格随着时间改变，而基于案例推理（Case-Based Reasoning）将乘客的历史订单作为案例，根据历史订单计算当前所有航班的相似度得分，如果乘客购买航班与推荐航班不符合，则优化调整航班相似度计算参数，最后将乘客此次选择作为新的案例case加入到推荐算法中。
* 协同过滤(Collaborative Filtering, CF)包括基于用户的协同过滤(UserCF)、基于物品的协同过滤(ItemCF)和基于模型的协同过滤(ModelCF)。UserCF给用户推荐有公共兴趣特征的用户喜欢的物品，反应的是社会中有相似兴趣群体的热门商品；而ItemCF给用户推荐和之前喜欢、购买的物品相似的物品，反应的是用户个人的兴趣爱好；ModelCF的重点在建立用户模型，可以用统计模型、最大熵模型等进行建模，根据用户模型对物品评分。协同过滤的缺点在于推荐结果的好坏取决于数据集的质量，而且一般情况下，数据会存在稀疏性等问题。
* 基于网络的推荐中将用户和对象都抽象成网络中的一个节点，例如在文献[[[41]](#endnote-42)][[[42]](#endnote-43)]中，如果对象选择物品，则图中节点和间存在一条边，若系统共有个用户个物品，则构成一个共有个节点的二分图，推荐中把物品看为可分配的资源，并将资源分配给喜欢的用户，则物品分配给物品的资源为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-1) |

其中函数表示节点的度。如果用户曾经购买过某物品，则物品初始资源为1，否则为0，从而每个用户的初始资源分配情况可以简化为维向量，蕴含用户的个性化信息，通过计算得到物品最终的资源分配矩阵，对于用户没有购买过的商品，按照资源分配矩阵中的值由大到小进行排序，越大则表示用户越喜欢，推荐度越高。

* 混合推荐是指将不同类型的推荐算法进行组合，每种推荐各有优缺点，而混合推荐可以取长补短，得到更符合用户需求的推荐结果，可从两个角度考虑：1)混合多种推荐算法计算结果得到最终推荐列表，混合的过程可以使用投票机制[[[43]](#endnote-44)]、预测打分[[[44]](#endnote-45)]等模型；2)混合多种推荐算法，基于某种推荐策略，融合其他的推荐算法，得到最终的推荐结果。

目前国内外研究的推荐算法多种多样，还有基于排序[[[45]](#endnote-46)]或关联规则分析[[[46]](#endnote-47)]的推荐算法，未来研究也会主要集中在以下几个方面：1)用户偏好和物品特征的提取；2)推荐算法安全性、多样性的研究；3)数据稀疏性，及推荐系统冷启动问题。

* + 1. 常用相似度的计算方法

基于内容的推荐和协同过滤算法的核心都是相似度计算，相似度直接反应了用户对物品的喜爱程度，也是用户间相似程度的衡量标准，结果将直接影响最后物品的评分和推荐的结果。相似相关系数及余弦相似度都是非常经典的相似度算法。下面分别介绍三种比较常用的方法。

皮尔逊相关系数用于描述两个变量间的线性关系紧密程度，即共变性强度，相关系数越高，则用一个变量预测另一个变量取值的精准度越高。若相关系数为0，则表示两个变量间没有关系；如果相关系数为负数，两个变量间呈负相关；而相关系数绝对值越小，两个变量相关性越弱。在基于用户的协同过滤中，评判用户­和的相似度时可以用皮尔逊相关系数计算，假设用户­和同时给物品集合打分，有：

|  |  |  |
| --- | --- | --- |
|  |  | (2-2) |

其中表示用户对物品的评分，表示用户对物品的评分，表示用户对所有物品的平均打分，同理表示用户的平均打分。但是使用皮尔逊系数有一定的限制条件，例如：变量间存在线性关系，且连续、相互独立、符合正态分布。

余弦相似度通过两个向量间夹角余弦值作为相似度的衡量标准，假设有两个维向量和，则计算公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (2-3) |

计算结果取值范围是，其中-1表示完全相反，1表示完全相同，0表示完全独立。余弦相似度常用于文本挖掘中的文档对比，统计文档中每个词出现的频率，没出现过的词表示为0，则两个文档的维向量夹角余弦值即是文档相似度取值；在协同过滤中，固定物品数为，用户和对个物品的评分可以被看成两个维向量，而用户的相似度则由评分向量的夹角余弦值表示。但是余弦相似度并未考虑不同用户的评分标准不同，有些用户习惯性所有打分都很高，有些打分普遍偏低，并不能很好的处理此类分数差距较大的现象。故可以使用评分的平均值和，消除打分标准差异的问题，计算方法如式(2-3)，在余弦相似度上加入归一化，用以处理用户评分标准不统一的现象，但实则是皮尔森相关系数。

|  |  |  |
| --- | --- | --- |
|  |  | (2-4) |

Spearman相关系数又称为秩相关系数，计算和皮尔森相关系数相同，但不同于余弦相似度直接使用评分计算，而是将评分数据按照定长划分为多个区间等级，根据评分的区间等级值进行计算相似度。文献[[[47]](#endnote-48)]中曾比较过Spearman和皮尔森相关系数，实验表明二种相似度计算方法推荐的效果相差并不多。

杰卡德相关系数用于度量两个集合间相似度的一种指标，分别计算两个集合的交集元素个数所占并集元素个数的比例：

|  |  |  |
| --- | --- | --- |
|  |  | (2-5) |

集合则为用户为物品的打分集合，杰卡德相关系数更适合离散特征的相似度计算，并没有考虑离散特征的具体取值。

目前虽然以上介绍的多种相似度、相关系数计算方法都可以获得不错的推荐结果，在商业中也广泛使用，但仍有许多不足，其中数据稀疏问题最为关键。对于所有的物品集合，用户评分标注的个数是十分少的，评分矩阵数据的确实会严重影响相似度计算和推荐结果，例如：如果两个用户共同打分的商品只有1个，则杰卡德相关系数计算非常不准确，余弦相似度为1；如果用户给所有物品打分相同，将无法使用皮尔森相关系数计算。

* + 1. 常用评价指标

推荐系统根据用户的偏好对物品进行推荐，在推荐系统中，常见的评价方式分为在线和离线两种。在线评价通过在线的推荐实验，通过收集用户的实时反馈（包括评分、问卷调查）衡量推荐结果的好坏，A/B测试是工业上常用的在线实验方法：对于推荐方案A和B，只有一种变量不同，一部分用户使用A方案，一部分用户使用B方案，记录并对比不同方案的实验效果，评估哪个方案更符合推荐的目标。虽然在线评价可以直观的评估推荐结果满意程度，但从设定实验环境，到实验上线、结果收集都需要耗费大量的人力、物力成本。所以目前大多数研究进行离线实验评估用户是否喜欢推荐的物品，但用户的评分尺度也有差异，不同的用户即使评分相同，满意度也并不相同，因此需要选择合理的评价指标。目前绝大多数评价指标针对推荐准确度进行评估，下面介绍一些常见的评价指标。

* MAE

对于物品集合固定的显示反馈推荐算法，可以计算用户实际评分和算法预估评分之间的误差，使用推荐评分的准确度来衡量推荐结果的好坏，最经典的指标是平均绝对误差（mean absolute error，MAE）[[[48]](#endnote-49)]，计算测试集所有推荐预估评分与实际评分偏差绝对值的平均，如式(2-6)，其中表示推荐物品集合、为用户对物品的真实评分、为推荐计算的物品评分。因为MAE计算方法简单，所以被许多推荐系统广泛应用[[[49]](#endnote-50)]。另外类似的评价指标有MSE（均方误差，式(2-7)）、RMSE（均方根误差，式(2-8)）等。

|  |  |  |
| --- | --- | --- |
|  |  | (2-6) |
|  |  | (2-7) |
|  |  | (2-8) |

* AUC

AUC用于衡量推荐系统可以在多大程度上区分用户喜欢和不喜欢的商品，是一种计算分类准确性的指标，且推荐列表长度不受限制。AUC即为ROC曲线覆盖的面积，横坐标为不相关商品比例，纵坐标为相关商品比例，由于横纵坐标取值范围都是0-1，所以AUC取值范围也是0-1。根据推荐商品顺序注意判定用户是否喜欢该物品，若用户喜欢则曲线上移，若不喜欢则右移。因此，AUC越大表示推荐准确度越高，可以设定阈值当AUC时，表示推荐结果优于随机推荐。虽然AUC指标涵盖了推荐列表中所有商品的判断，但是并未考虑商品在列表中排序的位置情况。

* 排序准确度

对基于排序的推荐算法，使用评分相关的评价指标衡量推荐结果的好坏显然并不合适，此类算法需要比较推荐物品的排序和用户对物品排序的一致程度，例如：系统A中用户最喜欢的物品在推荐列表中排名第2，但在系统B中排名第5，显然系统A推荐算法优于系统B。因此基于排序的推荐算法，文献[[[50]](#endnote-51)]中提出ARS（average rank score）衡量推荐系统排序准确度，即对于用户而言，物品的排序得分为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-9) |

其中为物品在推荐列表中的序号，为推荐物品的总数，ARS值越小，表示用户喜欢的商品排序靠前，推荐效果越好。该评价指标不需要用户对物品的任何评分信息，因此可以很好的衡量不同的推荐算法在相同数据集上的推荐结果。

此外，在推荐系统中还有许多其他的评价指标，例如：覆盖率[[[51]](#endnote-52)]适用于为用户找出所有感兴趣物品的推荐系统、多样性包括推荐物品的多样性和推荐不同用户不同物品的多样性，选择合适的评价指标在推荐系统中十分重要。

* 1. 缓存策略

如今互联网信息高速膨胀，搜索服务需要在较短的时间内返回准确的查询结果，而缓存作为一种提高搜索性能的方法，已经被广泛应用于CPU、操作系统内存、web缓存等计算机领域中，提高缓存命中率可以显著有效的缩短搜索时间，提升服务性能。由于缓存容量是固定的，为了提高命中率，需要对未来数据的访问情况进行预测和判断，替换将来访问概率小的数据块。

目前主要的缓存策略分为静态和动态两种。静态缓存策略是一种离线计算方法，通过离线统计数据块的访问频次，对缓存内容进行更新，在此种策略下，缓存内容在一段时间内不会发生变化。动态缓存策略是在分析搜索数据实时更新的情况下，对缓存内容进行更新，当缓存容量不够时，根据规则需要选择部分被替换的缓存内容，并填充新的数据项。经典的缓存策略包括FIFO、LFU、LRU算法等，例如：LFU算法根据数据的访问频次判断是否需要被替换，但访问频次高的数据如果一直不被访问，仍然会占用缓存容量，降低命中率；LRU算法中认为最近被访问的数据，将来访问概率也大，因此将缓存结果的最后一次查询时间作为选择替换数据块的依据，在文献[[[52]](#endnote-53)]中对LRU及其变种（FBR、LRU/2等）算法比较命中率结果，在缓存容量较少时可以获得较高的命中率。

基于概率分布的缓存策略，通过分析每个数据块被查询的频率分布，使用概率预测模型分析不同数据块将来的访问情况，并缓存访问概率大的数据块从而提高命中率。除了以上介绍的缓存算法，许多缓存的研究集中在流行度上，定义流行度的计算方法，流行度越高，数据被访问的概率则越大。文献[[[53]](#endnote-54)]中将视频分为多段，根据网站用户的访问频次作为流行度，根据流行度定义视频文件的流行等级，并依据视频段的访问频次和视频文件流行等级，制定缓存替换策略，该算法的缺点是视频文件一般较大、视频段数量较多，导致缓存替换时计算量较大，降低了算法效率。

* 1. 本章小结

本小节中主要对路径规划、路径推荐和缓存策略三个方向的相关工作进行详细阐述。路径规划中介绍了经典的和启发式的最短路径算法，包括经典的KSP问题求解方法，并简单描述了路径规划中的skyline查询；在路径推荐中，主要对现有路径推荐算法、显示/隐式推荐系统进行总结，并介绍了相关的相似度计算方法和推荐算法评价指标；最后介绍了缓存策略的经典算法和目前研究的基本情况。

1. 用户购买决策模型
   1. 引言

旅客模拟器是一种可以模拟环境中旅客选择行为的工具。选择行为得到了广泛的研究。例如，[10]提出了一种服务质量差距模型作为选择模型的效用函数。 Cao J.等人提出了描述乘客对航班偏好的模型[11]。此外，还开发了一套乘客模拟器。其中，旅客起源地模拟器（PODS）是最著名的一种，它可以模拟航空业中各种可能的旅行选择中个人旅行者的选择[9]。具体来说，在PODS中应用Logit选择模型来处理乘客的偏好。诸如重新计划之类的其他因素被替换为无效功能，并添加到给定航班的总成本中。 PODS需要输入订单号。在我们的模拟器中，请求数量将基于历史数据进行预测。此外，我们的模拟器可以根据最新数据调整其固有模型。

旅客模拟器是一个向强化学习模块提供反馈的平台，与这些模型有关的一些重要概念如下：

1）查询机票的日期和飞行日期：发生购票动作时有两个日期。查询机票的日期表示旅客搜索机票并在满意的情况下预订机票的日期，而航班日期则表示航班起飞的日期。

2）旅客类型：根据旅客的属性和预订机票的方式，他们可以分为五类[18]。这五种类型是直接预订机票并自行旅行的直接客人，预订商务，旅行社，旅游团体和分销商机票的商务客人。由于每个班级都有自己的属性，因此模拟器将生成不同类型的不同乘客。

3）购买比例：乘客查询票后，即使他们发现这些票可以满足他们的要求，也只有其中一些最终下订单。最终订购数量与对某些机票感到满意的乘客数量之间的比例称为购买率。

* 1. 基于特征分布的用户购买决策模型

3.2.1用户购买决策框架设计

Blablabla，如图所示

3.2.2用户生成模型

3.2.3用户选择模型

基于对历史顺序的分析和其他研究[11]，以下三个特征是在模拟旅客对机票的偏好时最重要的特征：

{\ bfseries 1）航空公司：}

 这代表了乘客最喜欢的航空公司，并且可以用一个向量表示，其中每个维度代表一个航空公司，值代表乘客对这家航空公司的偏爱。

{\ bfseries 2）旅行类别：}

此功能代表乘客的首选机票等级。 W.o.l.g，在我们的研究中，课程分为经济，商业和商业。\\

{\ bfseries 3）价格：}

  此功能代表乘客的价格敏感性。由于不同航班的价格范围各不相同，这意味着绝对价格不能表示乘客的价格偏好。在这里，我们引入了一种称为价格敏感度的度量，以衡量人们偏爱便宜机票的程度。对于给定的出发和到达城市对，价格敏感性等于：

\ begin {equation}

P\_ {sen} = \ frac {p\_ {high}-\ overline {p}} {p\_ {high}-p\_ {low}}

\ end {equation}

其中$ p\_ {high} $和$ p\_ {low} $分别代表同一航班的最高价和最低价。 $ \ overline {p} $表示乘客选择的机票的平均价格。

我们使用$ Prf\_u = <A\_u，C\_u，p\_u> $表示乘客$ u $的偏好，其中$ A\_u，C\_u，p\_ {sen u} $分别是航空公司偏好，舱位偏好和价格敏感性。

此外，航空公司的偏好表示为：

\ begin {equation}

A\_u = [a\_1，a\_2，a\_3，...，a\_n] \ quad s.t. \ quad 0 \ leq a\_i，\ sum\_ {i = 1} ^ {n} a\_i = 1

\ end {equation}

其中$ n $等于历史订单中出现的不同航空公司的数量。

类首选项定义为：

\ begin {equation}

C\_u = [c\_1，c\_2，c\_3，...，c\_m] \ quad s.t. \ quad 0 \ leq v\_i，\ sum\_ {i = 1} ^ {m} c\_i = 1

\ end {equation}

其中$ m $等于历史订单中出现的不同类别的数量。

类似地，票证可以用$ Tck\_t = <A\_t，C\_t，p\_t> $表示。 $ A\_t $是$ n $维列向量，指示此机票中的航空公司信息。对于$ A\_t $的$ i $ th元素，当该机票由$ i $ th航空公司提供时，它为1，否则为0。$ C\_t $是一个$ m $维列向量，指示此机票中的类信息。对于$ C\_t $的$ j $ th元素，当此票证是$ j $ th类票证时，则为1，否则为0。$ p\_t $是价格度量标准，也使用等式计算。 1。\\

％\ subsubsection {乘客实用程序功能}

{\ bfseries乘客实用程序功能}

机票$ t $上乘客$ u $的效用函数表示为$ U\_u ^ t $，定义为：

\ begin {equation}

\ begin {split}

U\_u ^ t＆= \ omega\_1（A\_u \ cdot A\_t）+ \ omega\_2（C\_u \ cdot C\_t）+ \ omega\_3 F（p\_u，p\_t）\\

F（x，y）＆= \开始{cases}

1＆\ text {x $ \ geq $ y} \\

\ frac {2} {1 + e ^ \ frac {2（y-x）} {x}}＆\ text {x $ \ textless $ y}

\ end {cases} \\

\ sum\_ {i = 1,2,3} \ omega\_i＆= 1

\ end {split}

\ end {equation}

乘客效用函数的值反映了乘客对给定票证的满意程度。并且我们将$ U ^ \* $定义为确定是否满足乘客的阈值。换句话说，当且仅当$ U\_u ^ t \ geq U ^ \* $时，乘客$ u $才对机票$ t $感到满意。

％\ subsubsection {购买率估算}

{\ bfseries购买率估算}

购买率随时间变化。假设查询日期和飞行日期的购买率表示为$ BR（qd，i）$，其中$ qd $表示查询机票的日期，$ i $表示查询日期和航班日期之间的天数。我们对给定查询日期和不同航班日期的购买率感兴趣。假设如果航班日期晚于查询日期MaxConv天，则购买率将变为零。因此，可以使用线性函数估算购买率：

\ begin {equation}

BR（qd，i）= \ alpha \* \ frac {MaxConv-i} {MaxConv}，\ quad i = 0,1,2，...，MaxConv

\ end {equation}

其中，\\ alpha $是衰减系数，可以根据实际订单数据进行估算。

假设$ d\_2 $和$ d\_1 $分别代表查询日期$ qd $前两天和一天。我们使用$ Num\_p（qd，i）$表示对机票满意的乘客数量，而$ Num\_O（qd）$表示实际在日期$ qd $下的订单数量。由于衰减系数是自相关的，因此可以估算日期为{{qd} $的衰减系数，如下所示：

\ begin {equation}

\ alpha\_ {qd} = \ theta \ alpha\_ {d\_1} +（1- \ theta）\ alpha\_ {d\_2}

\ end {equation}

其中，\\ theta $是自相关系数。并且在我们收集了数量为$ Num\_O（qd）$的实际每日订单之后，我们可以在日期$ {qd} $上更新衰减系数的值，因为它满足：

\ begin {equation}

Num\_O（qd）= {\ sum\_ {i = 0} ^ {MaxConv} Num\_p（sd，i）\ cdot（\ alpha\_ {qd} \ frac {MaxConv-i} {MaxConv}}）

\ end {equation}

这意味着：

\ begin {equation}

\ alpha\_ {qd} = \ frac {Num\_O（qd）} {\ sum\_ {i = 0} ^ {MaxConv} Num\_p（sd，i）\ cdot \ frac {MaxConv-i} {MaxConv}}

\ end {equation}

3.2.4模型总览

图2显示了模拟器框架。给定查询日期$ qd $，$ MCv $天，乘客数量，效用阈值$ U ^ \* $，衰减系数$ \ alpha\_ {qd} $和相关历史数据，我们需要预测日期的订单数\ emph {qd}。

模拟器将生成一组乘客，其数量为$ S\_ {qd，i} $，代表在日期$ qd $和航班日期为$ i $天后搜索航班的乘客数量。根据乘客类型分布生成每个乘客，并根据历史数据随机生成其偏好模型。例如，假设有3个航空公司$ a\_1，a\_2，a\_3 $，这三个航空公司的出现率分别为$ 0.1、0.3 $和$ 0.6 $。等式中表示的航空公司偏好。根据该分布随机生成2。

该实用程序功能应用于每个乘客，以模拟选择过程。每个乘客选择最大化$ U\_u ^ t $的机票$ t $，如果$ U\_u ^ t \ geq U ^ \* $，则此乘客称为满意乘客。查询日期为$ qd $且航班日期为$ i $天后的满意乘客的数量表示为$ Num\_p（sd，i），i = 0,1，... MCv $。因此，可以使用公式计算查询日期\ emph {qd}上的预测总订单号。 7.最后，在获得日期$ qd $之后的实际订单号之后，我们可以根据等式更新衰减系数。 8。

* 1. 基于生成对抗的用户决策模型

前文

* 1. 本章小结

本章节首先介绍了面向乘客出行的个性化路径推荐模型，介绍与乘客的交互内容，路径的缓存、搜索和推荐部分，及其之间的相互联系。同时详细介绍了模型中的名词含义，及使用的具体场景，为后文的路径搜索、推荐算法和缓存策略的评估、实验分析奠定基础。

1. 带约束的top-K路径搜索算法
   1. 引言

在上一章中，我们主要介绍了个性化路径推荐模型，而路径推荐的核心在于路径的搜索，乘客的需求和约束是搜索的输入，而路径搜索的输出是推荐和缓存的关键。

由于出行路径的多态性，且不同航班价格实时动态变化，路径搜索是个性化推荐的关键部分。因此，本章将重点提出的一种路径启发式搜索算法KSPG，在路径搜索前，由乘客给定出发到达城市、出发时间区间，且乘客可以提出一些约束条件，例如总耗时、价格区间等。假设共有n个常用的中转车站，t种交通工具，该启发式搜索算法目的是在较短时间内搜索出前K条满足约束的最优、最短路径。

* 1. 路径搜索算法KSPG
     1. 路径搜索问题定义

由中转车站构成的换乘网络可用有向有权图表示：每一个中转车站表示中的一个点；而表示一种可行的交通出行方案，例如：高铁、飞机、汽车等，可以将乘客从送至，每条边的方向表示出行路径的方向；每一条边的权重表示为此种交通工具QoS值，例如：价格、出行时间等，其中有些值是动态变化的。所以当乘客给定查询、约束条件后，在给定的交通网络中，路径搜索问题则转变成一个寻找top-K条满足约束的最短路径问题，即KMCSP（K-Multiple Constrained Shortest Paths），下面介绍KMCSP的相关概念。

假设表示一个有向有权图，其中表示图的顶点集合，表示图的所有有向边集合。每条边有种QoS值表示为，其中用 表示为该边的权值：

|  |  |  |
| --- | --- | --- |
|  |  | (4-1) |

在给定源点和终点后，表示所有可行路径集合。乘客在给定个约束条件后，KMCSP的目标是从到的路径集合中，找到条最短路径，每条路径均满足式(4-2)的约束，其中是计算路径第r种QoS权值的函数，式(4-3)中表示路径的总权值。

|  |  |  |
| --- | --- | --- |
|  |  | (4-2) |
|  |  | (4-3) |

假设，该问题则退化为MCSP问题，是一个整数规划问题，找出满足约束的最短路径，可表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (4-4) |
|  |  | (4-5) |
|  |  | (4-6) |

同时已经有研究证明KMCSP和MCSP均为NP完全问题[x]。

MCSP问题是在图中找到从s到t的权重和最小的路径，使得其他任何一条路径且，都满足。而KMCSP是最短路径问题的扩充，除了生成满足约束的最短路径以外，还要依次生成次短路径，直到找到第条满足约束的最短路径。其中路径集合 中的条路径需要满足以下条件：

* 条路径依次按顺序生成，即是生成的第一条路径，在后生成；
* 条路径，均满足；
* 条路径均满足R个约束条件；
* 条路径为s到t所有路径中，满足约束的top-K的最短路径，一定是图中从s到t符合条件约束的最短路径。
  + 1. KSPG算法

上一小节给出了KMCSP的详细定义，根据乘客输入的条件和实时的交通信息构建有向有权网络，根据给定的约束条件搜索可行路径集合。第二章中曾介绍了算法的具体过程，传统算法的优点是需求可控性，启发函数在算法的搜索过程中尤为重要，好的启发函数可以使得算法一直朝着最优路径方向进行搜索，提高搜索效率，从而更快的找到符合要求的路径。本文基于算法，提出一种启发式搜索算法KSPG（K Shortest Paths Generator），可以快速的搜索前K条满足约束的最短路径。

在本小节中，首先定义一些集合符号：表示任意点之间的可行路径集合；表示以为出发点，到达任意点的可行路径集合；表示以为出发点、为终点的所有可行路径集合；表示从出发点到终点的任意一条路径。此外，下面将给出一些重要定义。

**可行路径集合**（feasible path set）。出发点到终点的路径构成可行路径集合的条件为：

|  |  |  |
| --- | --- | --- |
|  |  | (4-7) |

**下限距离**（lower-bound distance）。从点到，在给定约束条件时，需满足：

|  |  |  |
| --- | --- | --- |
|  |  | (4-8) |

**可到达距离**（admissible distance）。从点到，在给定约束条件时，需满足：

|  |  |  |
| --- | --- | --- |
|  |  | (4-9) |

其中两个R维向量A和D间，符号表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (4-10) |

显然从式(4-9)不等式可看出，是一种可行的可到达距离取值。

**投影距离**（projected distance）。给定约束条件、起始点s和终点t后，对于中的其中一条路径，投影距离函数表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (4-11) |

**可到达备选路径集合**（admissible header set, AHPS）。给定点和约束，可到达的备选路径集合是：

|  |  |  |
| --- | --- | --- |
|  |  | (4-12) |

即集合中的路径是搜索结果，另外表示可到达备选路径的前K条路径集合。

**引理4.**.为KMCSP问题的解。

**证明：**假设*p*是一条从*s*到*t*的路径，则有：

|  |  |  |
| --- | --- | --- |
|  |  | (4-13) |

且，故得证：

|  |  |  |
| --- | --- | --- |
|  |  | (4-14) |

即为KMCSP问题的解。

**引理4.**.，路径均为扩展获得。

**证明：**由定义可知，是由点扩展生成的一条路径，定义可以用经过的点表示：

|  |  |  |
| --- | --- | --- |
|  |  | (4-15) |

即从到经过个点，构成了路径。

**引理4.**.如果一条路径不是由可到达备选路径集合中的路径扩展生成的，则该路径一定不属于可行路径集合：

|  |  |  |
| --- | --- | --- |
|  |  | (4-16) |

**证明：**假设给定源点和终点，路径不属于，即满足。假设是点的一条扩展边，则对于有：

|  |  |  |
| --- | --- | --- |
|  |  | (4-17) |

则，满足：

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | (4-18) |
|  |  |  |
|  |  |  |

所以由不等式(4-18)知也不属于可到达备选路径集合，故得证。

从以上引理可得知，所有的可行路径解都是由可到达备选路径集合扩展生成。下面描述KSPG算法的详细步骤。算法KSPG以算法为基础，与剪枝技术结合求解KMCSP问题。KSPG算法从路径开始搜索，潜在的会搜索到所有属于的路径，但是根据约束加入减枝技术，可行的路径解只能由即中的路径扩展生成；同时，中路径投影距离最小的路径最先扩展，直到获得中条带约束的可行路径解，或AHPS中无可扩展路径，算法结束。算法中，生成的所有路径均是的子集。

算法中减枝技术会加快路径搜索的速度，算法4.1伪代码详细描述了KSPG过程，假设给定出发点、终点，约束。算法KSPG中选择投影距离最小、权重最大的路径进行扩展，使得算法能更高效的进行路径搜索，KSPG主要的步骤如下：

* 初始化计算：

已知图，首先计算每一个点的可到达距离,其中和点对有关，如果未知，可将其设为。目前有很多已知的算法用于计算可到达距离，在下一小节中会详细描述。同时，初始化AHPS集合，包含路径。

* 路径扩展：

假设目前有可到达备选路径列表为，由路径扩展生成。中，路径依据投影距离构建极小堆，如果路径的投影距离相等，则根据由大到小排序，并在删除、插入路径操作时维护此最小堆。

在路径扩展中，首先从中选择投影距离最短的路径，将点所有的可扩展边与生成新的路径，并将符合剪枝要求的路径添加至，并移除。当算法找到条可行路径解时，路径扩展结束。

* 带约束最短路径判定：

每次从中选择的路径，首先需要判定是否为符合约束的最短路径解，若是，则加入到中并从中移除。

* 剪枝函数：

在扩展路径添加至之前，需要判断是否属于，如果不属于，则删除该路径，否则添加至。 显然，所以当为空时，中所有的路径都已经被扩展。

* 算法终止条件：

算法KSPG只要满足以下两个条件之一立即停止：1)已经找到条最短路径；2)中路径为空。

|  |
| --- |
| **算法 4.1:**  **KSPG（K Shortest Paths Generator）** |
| **输入： –** a weighted directed graph  – the source vertex  – the target vertex  – the number of shortest path to generated  – the R QoS values of each edge  – the R constrains given by passengers |
| **输出：** – K paths with the lowest sum of QoS Values |
| 1. **For**   // admissible length from to ;  // admissible length from to ;  ;  END For   1. **初始化:**   ;  ;  ;  ;  ;   1. **While and AHPS is not empty do:** 2. HeaderSort(AHPS); 3. the first path of AHPS; 4. remove the first path of AHPS; 5. **if** **then**: 6. Insert into ; 7. ; **goto 3;** 8. **end if** 9. ExpandEdges ; 10. **while** ExpandEdges is not empty **do:** 11. remove an edge from ExpandEdges; 12. ; 13. **if**  has a cycle **then:** 14. **goto 12;** 15. **end if** 16. **for**  **do:** 17. **if then:** 18. **goto 12;** 19. **end if** 20. **end for** 21. Insert into AHPS; 22. **end while** 23. **end while** 24. return ; 25. **Function**   **return ;**   1. **Function** *HeaderSort* ( AHPS ):   **return** The path with the minimum and maximum ; |

搜索算法通常从以下三个方面进行评估：

* 最优性：如果问题有多种解，算法是否保证可以求得最优解；
* 时间复杂度：算法运行消耗的时间；
* 空间复杂度：算法运行消耗的内存。

**引理4.**. KSPG算法具有最优性。

**证明：**假设有两条满足约束可行路径解和，且满足。首先我们通过反证法证明在之前扩展生成。

假设路径在之后扩展生成，将表示为，即从算法中可知，存在边一定在*AHPS*中，且路径在之后扩展，同时满足：

|  |  |  |
| --- | --- | --- |
|  |  | (4-19) |

且是一条可行路径解，从引理4.3可推导：

|  |  |  |
| --- | --- | --- |
|  |  | (4-20) |

显然式(4-20)与假设相悖，故假设在之后扩展生成不成立。所以证得如果，则在之前扩展生成。故在KSPG算法中，top-K条满足约束的最短路径是按照路径总权重由小到大依次扩展生成，所以KSPG具有最优性。

KSPG算法生成条可行路径解，是路径搜索中约束的个数，是*AHPS*中总共扩展的路径数量，是条路径中最多的边数，是图中最大的出度数。算法的空间复杂度为，时间复杂度为。如果为指数增长，则KSPG也为指数时间复杂度。KSPG结合算法和剪枝策略，比效率高。当满足式(4-21)不等式时，KSPG是准多项式时间复杂度。因此可到达距离取值越大，KSPG算法时间复杂度越低。

|  |  |  |
| --- | --- | --- |
|  |  | (4-21) |

综上所述，KSPG算法结合约束条件，首先扩展最可能生成可行解的路径，并结合剪枝策略，删除不满足中间约束的可到达备选路径，减少*AHPS*中可扩展的路径，从而降低复杂度。最终算法搜索得到条QoS和最小且满足约束的可行路径解作为推荐的候选路径。

* 1. 可到达距离计算方法

目前，上一小节式(4-9)中可知下限距离是一种可到达距离计算方式，虽然对扩展路径的选择、减枝作用很大，但是计算复杂度太高。

**Uniform距离**。Uniform距离即为，是一种可行的可到达距离取值。基于Uniform距离的KSPG算法会扩展所有的路径，按照路径权重由小到大依次扩展。虽然在路径搜索的过程中无需其他计算操作，但是Uniform距离对剪枝策略几乎没有作用，所以合适的可到达距离的取值应该介于Uniform距离和二者界限之间。

**Dijkstra距离**。对于每一种QoS值，Dijkstra算法可计算出一条最短路径，由此条最短路径权重值构成的向量表示为，结合式(4-8)中的定义，对有：

|  |  |  |
| --- | --- | --- |
|  |  | (4-22) |
|  |  | (4-23) |

显然式(4-23)一定成立，则有：

|  |  |  |
| --- | --- | --- |
|  |  | (4-24) |

等同于：

|  |  |  |
| --- | --- | --- |
|  |  | (4-25) |

所以Dijkstra距离一定是一种可到达距离。

同时在式(4-4)(4-5)(4-6)中，MCSP被标示为整数规划问题，而该整数规划的松弛问题解也是一种可到达距离。在相关的文献中，提出了许多算法用于计算MCSP松弛问题下界的准确解和近似解[[[54]](#endnote-55)]，拉格朗日松弛算法[[[55]](#endnote-56)]也可以计算MCSP整数规划问题的下界。显然，以上方法均可作为可到达距离的计算方法。

* 1. 实验结果分析

在本小节中，我们对KSPG算法进行实验结果分析，为了更好的展示KSPG算法在路径搜索中的效果，本章的实验中共选取6条最热门的行程，出发-到达城市分别为：北京-上海、北京-重庆，三亚-上海，三亚-北京，北京-广州、北京-杭州。这些城市对间既有直飞的航班、直达的高铁，例如：上海到北京，有直飞航班或高铁直达，可以从上海乘坐高铁到杭州，由杭州飞往北京等多种出行方案。其中城市对间的中转城市如表4 - 1所示，例如：北京-上海的中转城市有南京、杭州、石家庄、天津。

|  |  |
| --- | --- |
| **城市对** | **中转城市** |
| 北京-上海 | 南京、杭州、石家庄、天津 |
| 北京-重庆 | 石家庄、成都、天津 |
| 上海-三亚 | 海口、南京、杭州 |
| 北京-三亚 | 海口、石家庄、天津 |
| 北京-广州 | 深圳、石家庄、天津 |
| 北京-杭州 | 南京、石家庄、天津、上海 |

表4 - 1 路径搜索实验-换乘车站

Table 4 - 1 Transfer Stations of the Route Searching Experiment

一般在实际应用中，乘客首先会给定出发、到达城市和预期出发时间，同时给定可以接受的最长出行时长、最高出行费用、最多的换乘次数等条件约束。为了验证本章中提出的路径搜索算法在实际应用中的效果，我们将使用通过爬虫记录的飞机、高铁实时价格数据，还原真实的路径搜索场景，对算法结果进行评估。首先为了直观上理解数据的属性，选取部分数据集进行介绍。

* + 1. 实验准备
* 机票价格数据

机票价格数据涵盖了从2014年10月15日到2015年4月28日，上海、北京、广州、香港、三亚、成都、杭州等城市间共36条航线上不同航班的实时机票价格，其中包括以上海到北京为代表的商务航线、以北京到三亚为代表的旅游航线等，每条航线包括多个航班号。每个航线中包含多个航班，对于每一条航班，每一个出发日期，

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **抓取**  **日期** | **起飞前天数** | **0:01-2:00** | **2:01-4:00** | **4:01-6:00** | **…** | **20:01-22:00** | **22:01-0:00** | **最低价格** |
| 2014.11.16 | 30 | 353.0 | 353.0 | 353.0 | … | 333.0 | 333.0 | 333.0 |
| 2014.11.17 | 29 | 353.0 | 353.0 | 343.0 | ... | 319.2 | 353.0 | 319.2 |
| 2014.11.18 | 28 | 353.0 | 353.0 | 339.2 | … | 337.2 | 339.2 | 337.2 |
| … | … | … | … | … | … | … | … | … |
| 2014.12.13 | 2 | 830.0 | 830.0 | 769.0 | … | 830.0 | 830.0 | 769.0 |
| 2014.12.14 | 1 | 830.0 | 769.0 | 769.0 | … | 1100.0 | 769.0 | 769.0 |

表4 - 2航班KN5977出发前30天最低价格数据概览

Table 4 - 2 Lowest Flight Price of Flight KN5977 Crawled During 30 Days before Departed

都通过网络爬虫平均每两小时抓取一次携程网[[56]](#footnote-1)上的实时机票价格数据，每个航班至多只抓取距离起飞前30天的机票价格，即在1月1日会记录1月1日至2月1日内出发的所有航班价格。在构建实时交通换乘网络前，需要对现已有数据预处理，剔除冗余、不一致、不相关的数据。由于机票的价格数据与航班、抓取时间等多个维度相关，为了便于理解和实验，分别构建每条航线的最低机票价格矩阵，即对于一条固定的航线，抓取价格的日期即可视为订票日期，并根据起飞日期和距离起飞日期前天数，对机票价格数据进行整合。

例如北京到上海的KN5977航线，出发日期为2014年12月15日，出发前30天最低价格矩阵如表4 - 2所示，即给定本航班起飞日期和距离起飞前天数后，矩阵中单元格中的价格表示在抓取日期当天对应时间段内，抓取到的该航线所有航班中机票价格的最小值，例如：第1行第3列单元格的价格为353元，表示在2014年11月16日0点至2点，预定2014年12月15日航班KN5977的机票最低价；同理第一行最后一列表示，2014年11月16日任何时段预定2014年12月15日航班KN5977的机票价格最低值为333元。

* 高铁价格数据

高铁价格数据涵盖的时间区间为2014年1月1日至2015年12月30日，高铁的票价相对固定，主要是车次出发-到达时间的记录。数据表4 - 3是北京到天津的高铁、动车价格表，对比第一、二行后可知，G387次高铁列车在2015年6月30日之前是08:42出发、09:17到达，从7月1日开始改至08:25出发、09:05到达，但价格没有改变，均为54.5元。

|  |  |  |  |
| --- | --- | --- | --- |
| **车次** | **出发-到达时间** | **日期** | **票价** |
| G387 | 08:42-09:17 | 2014.01.01-2015.06.30 | 54.5 |
| G387 | 08:25-09:05 | 2015.07.01-2015.12.30 | 54.5 |
| C2209 | 10:09-10:47 | 2015.04.20-2015.05.17 | 54.5 |
| … | … | … | … |

表4 - 3高铁票价数据概览

Table 4 - 3 High-Speed Train Peking-Tianjin Price

* + 1. 实验结果与分析

实验中包括中转共涉及12个城市，每个城市表示为有权有向图中的一个点，根据预期出发的时间，获得爬虫记录的机票、高铁价格数据，并构建换乘交通网络，最终根据路径搜索中CPU耗时评估KSPG算法，实验中对比Uniform和Dijkstra两种可到达距离的搜索效率，并与深度优先搜索算法进行比较。

在2014年10月15日到2015年4月28日内随机选择一个出发时间和出发、到达城市，模拟一次路径搜索，共实验5000次。实验机器为Intel i7-4770处理器，内存8G，实验中有权有向图的每条边权重只包括时间和价格两种，换乘时各中转站间的代价依照实际所需的价格、耗时进行定义，统一标准化去除量纲影响，实验目标是生成前条权重和最小且满足约束的路径，记录搜索的总耗时。表4 - 4中比较了基于Uniform距离、Dijkstra距离的KSPG和深度优先算法的实时路径搜索耗时（单位：秒），其中深度优先搜索需要列举所有的可行路径，再根据权重筛选出满足约束的路径，并按照权重小到大进行排序后选择路径，所以的取值几乎不影响搜索耗时。但无论是Uniform距离还是Dijkstra距离，KSPG算法在搜索效率上都优于深度优先，KSPG算法在路径搜索时并不需要遍历所有可行路径解，而是在保证约束的情况下进行剪枝，并依照权重由小到大依次生成满足约束的路径，减少路径搜索时间。对比可得，Dijkstra距离明显在搜索效率上优于Uniform距离，Dijkstra算法虽然计算复杂度较高，但可以很好的对路径进行剪枝，同时将可到达备选路径集合（*AHPS*）中的路径按照投影距离由小到大依次扩展，提高搜索效率。

|  |  |  |  |
| --- | --- | --- | --- |
| **算法**  **K值** | **KSPG-Uniform** | **KSPG-Dijkstra** | **Deep First Search** |
| **K=1** | 28.93 | 26.84 | 50.89 |
| **K=50** | 30.78 | 29.02 | 51.27 |
| **K=100** | 33.12 | 31.83 | 51.56 |
| **K=150** | 33.42 | 32.57 | 50.48 |
| **K not limited** | 37.52 | 38.45 | 52.35 |

表4 - 4 KSPG算法5000次实验运行时间（单位：秒）

Table 4 - 4 KSPG Algorithm 5000 Running Time Comparison (time unit: second)

为了进一步对KSPG算法搜索效率进行评估，使用Dijkstra算法作为可到达距离，并随机选择5000个时间点，在上海-北京上进行路径搜索实验，共四个中转站。图4 - 1中，横坐标表示搜索窗口，其中搜索窗口的定义为：若搜索路径前给定具体的出发时间为14:00且搜索窗口为，则KSPG算法中路径的搜索范围是内出发飞机、高铁路径，其余时间段的出行路径不会出现在推荐结果中；纵坐标表示搜索前100条满足约束的最短路径耗费的时间。从图中可以看出，KSPG算法在搜索效率上优于深度优先算法，且随着搜索窗口的增大，可行路径解增多，导致搜索时间逐渐增多。

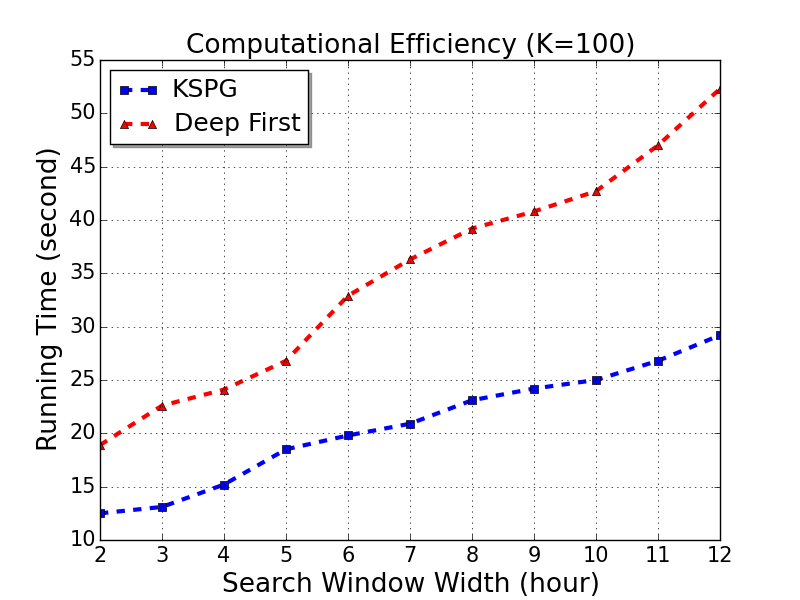
****

图4 - 1 上海-北京路径搜索耗时比较

Figure 4 - 1 Path Searching Time Comparison between SHA to BJS

* 1. 本章小结

本章中提出一种带约束的启发式路径搜索算法KSPG，通过实时交通数据，将城际间的换乘交通网络视为有权有向图，且每条边的权值由特征值表示，例如：价格、时间等。在给定源点、终点和约束条件后，KSPG按照路径权值由小到大依次生成前条满足约束的最短路径，作为推荐的备选路径，最后在4.4小节中评估KSPG算法的搜索效率。

1. 路径推荐算法
   1. 引言

在第三章中，我们介绍了个性化路径推荐模型，第四章中提出一种带约束的启发式路径搜索算法。而正如第一章中描述的，路径特征自身具有多样性，同一行程中可以包括不同航班、高铁车次、汽车等组成的出行路径，且路径是随时间变化的动态商品，每条路径都不是一个固定的商品，导致没有一个固定的商品集合，因此许多传统的推荐算法（例：矩阵分解、基于用户/物品的协同过滤算法）均无法很好的运用到路径推荐中。同时每条路径拥有多种特征，出发和到达时间、价格、航司、换乘数、退改签政策、机型等，而乘客在选择出行方案时，总倾向于选择满足自己要求的出行路径，路径特征的波动会直接影响乘客是否购买，而不同乘客有不同的购买决策因素，例如：价格敏感、航司敏感等。

在本章中，基于路径搜索过程中生成的条最优备选路径，提出了一种基于乘客偏好模型的个性化路径推荐算法，定义用户和路径、路径和路径、乘客和乘客间的相似度计算公式，对乘客的行程路径搜索结果进行推荐，同时提出一种基于乘客最近邻（Neighbor-Based）的路径推荐方法。

* 1. 特征预处理

在3.2小节中介绍了路径的特征属性值，包括航空公司，出发、到达时间，总耗时长，总票价，换乘次数及换乘间隔总耗时。属性特征包括：连续型和离散型，其中总耗时长、总票价、换乘间隔总耗时、出发/到达时间、换乘次数是连续型特征，而航空公司、舱位等级、退改签政策是离散型特征。

对于连续属性，需要在使用前应将不同属性的取值正则化，消除量纲的影响，使得取值范围均在。首先将价格转换为统一度量标准，由于在不同的行程上搜索返回结果的航班、高铁车次差异性，使得价格的衡量标准并不相同，例如：从上海到北京的航班500元属于较为普遍的平均价格，但是从广州到哈尔滨的航班500元应属于低价机票。因此引入业内常用的价格敏感指数，消除不同航班间价格的差异性，如式(5-1)中，表示航班机票的原价，表示当前搜索航班返回结果中的最低价格，表示乘客购买机票的价格，可以看出敏感指数的取值范围是，如果乘客购买的机票价格正好是当前时间点的最低价，则表示非常敏感；如果是以原价购买的机票，则表示非常不敏感，且价格敏感指数仍为连续取值。

|  |  |  |
| --- | --- | --- |
|  |  | (5-1) |

再将总耗时长标准化，标准化公式如(5-2)所示，其中是待归一化的时长，是总耗时长的最大值，为最小值。

|  |  |  |
| --- | --- | --- |
|  |  | (5-2) |

现假设共有种属性，其中前种是连续型属性，后种是离散型属性。一条路径的特征向量表示为，乘客的特征向量由历史订单数据生成，表示为，其中连续型特征为多个订单中特征值标准化后的平均值，离散型特征为订单中不同取值的数量统计，在每个属性上为唯一取值。因此，对于不同类型的特征应该采用不同的计算方法分析乘客的偏好情况。

* 1. 基于信息熵的路径推荐算法

信息用于消除了人们对事物的不确定性，而信息熵是信息论中信度量信息量大小的一种方法，表示信息在传播过程中产生的不确定性。熵原本是物理名词，表示热力学中密闭系统的稳定状态，例如：密闭空间中气体a、b混合达到稳态，如果此时需要将气体a、b分离，则必须依靠外界（给系统加入能量）使得a、b分离，系统更加有序化，熵值降低。因此可以使用信息熵表述信息不确定程度，同时本文中定义的特征信息熵完全由原始数据计算生成，故可以用于度量特征取值的有序化、稳定程度，信息熵越小，特征稳定性越高、不确定性越低；反之，不确定性越高，信息越混乱。

Shannon信息熵是常用的计算方法之一。对于离散随机变量，共种可能的取值集合，信息熵定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (5-3) |

其中表示出现的概率函数，可以为非的任何值，一般取，且当时，。从以上公式可以看出，当只有一种取值时，即时，取值最小，当大于一种取值，即时，每个取值出现概率相同时，最大为，则取值范围是，故消除量纲影响，将信息熵进行标准化后，计算方法如下：

|  |  |  |
| --- | --- | --- |
|  |  | (5-4) |

此外，对于一维数据集合，信息熵的计算与式(5-4)相似，方法如下：

|  |  |  |
| --- | --- | --- |
|  |  | (5-5) |

且若时，。当数据集合中m个数值完全相等时，信息熵最大为1，表示该集合中数据没有区分度；当只有一个数值不为0、其余都为0时，信息熵最小为0。

基于以上介绍，分析发现特征的信息熵，在选择出行路径时，可以很好的表示乘客在离散特征上的选择喜好程度，信息熵越小表示乘客在此特征上选择的确定性越强、稳定性越高。所以在路径推荐中，我们基于特征的信息熵计算乘客的偏好模型，计算推荐备选路径与乘客历史订单数据的相似度，根据相似度对备选路径进行排序，相似度越高，则表示乘客对该路径满意度越高，其中最重要的是相似度计算方法。

* + 1. 路径相似度计算

第二章中介绍了几种推荐中常见的相似度计算方法，此类计算方法广泛在商品集固定的推荐中使用，而本文中路径是实时动态变化的，是基于当前时刻交通信息的路径搜索和推荐，相似度是其中十分重要的步骤，计算结果直接影响了推荐路径的排列顺序。根据上文中定义的信息熵，我们介绍路径推荐中使用到的相似度计算方法。

假设乘客历史订单数据集合为，向量表示乘客在各个特征上的取值，其中为前列离散型特征，每个特征是一个一维数组，是在订单数据上该特征的统计值，例如：国内客运航司共24家，则航司特征值是长度为24的一维数组，每个值代表历史订单中该航司出现的数量；为列连续型特征取值，每个特征为一个连续型数值，是所有历史订单特征值的平均。

首先介绍路径相似度计算方法。假设路径共定义种特征，前列为离散型，后列为连续型，则两条路径的相似度可以定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (5-6) |

相似度的计算把离散、连续型特征分开计算，且和的取值范围均为；是定义的权重常数，表示为离散特征的重要性，取值范围是；为特征相似度的权重。

在路径推荐中，乘客偏好是隐性的，即使是乘客本身也无法形象的描述出自己的偏好，同时权重也体现了乘客对特征的偏好程度，权重越高，特征重要性越高。通常普适性的权重无法准确区分乘客特征的差异，不同乘客对路径的特征偏好也并不相同，因此表示乘客在此特征上偏好程度，影响最终乘客的选择结果。传统的权重确定方法，例如：层次分析法、经验法，人为因素对相似度的计算结果影响较大，最终影响最终推荐效果。因此下文中将基于信息熵确定特征的权重，权重值由数据样本直接计算产出，无人为因素干涉，具有一定的客观性。

下面先介绍离散型特征的相似度：

1. 特征权重的计算方法。使用式(5-4)计算中个离散特征标准化之后的信息熵值，得，则对于乘客，各特征的权重定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (5-7) |
|  |  | (5-8) |

从上式中可看出，特征取值相差越小，则信息熵越高，特征偏好权重越低，说明乘客在此特征上的偏好更小，在特征上选择确定性越弱。当权重为0时，表示该特征对乘客的偏好没有提供有用的区分信息，在偏好模型中不予以考虑。而对于权重高的特征，应重点关注，说明该特征提供的信息越多。

1. 定义相似度，其中、是路径和在特征上的离散值，相似度定义如式(5-9)，如果特征离散值不相同，则相似度为0。

|  |  |  |
| --- | --- | --- |
|  |  | (5-9) |

从第二章的介绍中可知，连续特征的相似度计算可分为基于距离和非距离两种。基于欧氏距离的定义，路径和为连续型特征取值向量，标准化后转换成模为1的向量，即可以使用修正的余弦相似度计算，计算如式(2-2)，虽然余弦相似度可以很好的衡量向量的相似度，但是忽略了乘客在各特征上的偏好问题，即所有特征的权重是相等的。

因此与离散特征权重计算方法相似，可以用公式(5-5)计算中个连

续型特征的信息熵，得，与离散特征权重计算相似，使用公式：

|  |  |  |
| --- | --- | --- |
|  |  | (5-10) |
|  |  | (5-11) |

计算每个属性的权重，下面在得到各连续型特征信息熵权重的基础上，介绍两种连续型特征的相似度计算方法：

1. 基于欧氏距离公式，在余弦相似度的基础上，加入乘客特征的偏好权重，如下式：

|  |  |  |
| --- | --- | --- |
|  |  | (5-12) |

1. 与离散型特征计算类似，提出一种基于距离的相似计算：

|  |  |  |
| --- | --- | --- |
|  |  | (5-13) |

其中表示在特征上的最大值，为最小值。

* + 1. 基于信息熵的路径推荐算法

在上一小节中，详细介绍了两条路径的相似度计算方法，本节中主要介绍基于信息熵的路径推荐算法，简写为IERA(Information Entropy Recommendation Algorithm)。在第五章中，KSPG算法基于乘客输入的需求和约束，搜索出条最优路径作为推荐的备选路径。针对乘客，IERA算法分为以下几个步骤：

1. 历史订单数据集的特征取值正则化；
2. 计算每个特征的信息熵，并基于信息熵计算每个特征相应的权重；
3. 计算备选路径与历史订单数据中每条路径的相似度平均值；
4. 将中条路径按平均值大小排序，即为备选路径的推荐顺序。

|  |
| --- |
| **算法****5.1:**  **基于信息熵的路径推荐算法(IERA)** |
| **输入： –** 乘客历史订单数据集  **L** – 路径特征总数  d – 路径离散特征总数  K – 备选路径的条数  – 备选路径集合  – 备选路径的评分数组 |
| **Output：按乘客偏好排序后的推荐路径集合** |
| 1. **基于订单，计算乘客各特征的信息熵:**   For j in (1,…,d):    End For  For j in (d+1,…,L)    End For   1. **分别计算每个特征相应的权值：**        1. **对中每一条路径分别与**历史订单数据集中每条路径**计算相似度，取平均值：**   For in :  定义相似度数组：arr\_sim  For f in :  End For    End For   1. **将备选路径按arr\_score评分由高到低输出，即为备选路径的推荐顺序。** |

从伪代码算法5.1中可知在推荐时，对于乘客，条备选路径中每条路径都要和历史订单中的所有路径使用式(5-6)进行相似度计算，并维护大小为的最大堆，因此时间复杂度为，其中为乘客历史订单数量，当为常数时，复杂度简化为。

通过对大量乘客的历史数据分析，发现历史订单数据十分稀疏，包含的出行路径也非常少。稀疏性影响了整个路径推荐的过程。因此，可以通过计算乘客间历史出行订单的相似度，找出top-n个相似度最高的乘客，使用这些相似乘客的所有订单数据计算各个特征的信息熵权重，共同对备选路径进行排序。下面定义乘客间相似度的计算方法，根据历史订单集合和，分别统计乘客、的特征向量、，且标准化处理消除量纲影响，乘客相似度计算如下：

|  |  |  |
| --- | --- | --- |
|  |  | (5-14) |

其中前项是离散特征，对于，和分别是、在特征上的统计向量，每个特征的相似度可使用Jaccard距离或余弦相似度进行计算，

|  |
| --- |
| **算法****5.2:**  **基于top-n相似乘客的信息熵路径推荐算法** |
| **输入：**  F – 历史订单数据集  L – 路径特征总数  d – 路径离散特征总数  K – 备选路径的条数  – 备选路径集合  – 备选路径的评分数组 |
| **Output：按乘客偏好排序后的推荐路径集合** |
| 1. **计算top-n个相似度最高的乘客，构建订单集合**   For in U：    arr\_user\_sim[] =  End For  取arr\_user\_sim中前n个乘客历史订单，构成集合   1. **基于订单，计算乘客各特征的信息熵:** 2. **分别计算每个特征相应的权值：** 3. **对中每一条路径分别与历史订单数据集中每条路径计算相似度，取平均值，保存至arr\_score中，** 4. **将备选路径按arr\_score评分由高到低输出，即为备选路径的推荐顺序。** |

而后项特征值都为乘客所有订单中相应特征取值的平均，则和均为维向量，相似度可以使用余弦相似度计算。

基于top-n相似乘客的信息熵路径推荐算法，在计算信息熵权重前，加入了相似乘客的查找过程，并用相似乘客的所有历史订单数据进行条备选路径排序。具体的伪代码见算法5.2，算法时间复杂度为，其中是乘客总数量,是所有相似乘客的订单总数量。

* 1. 实验结果分析

乘客的购买偏好不仅仅和航班自身的属性相关，同时也与其他可选的出行路径有一定关系，所以为了评估在本章中提出的基于信息熵的路径推荐算法，实验将使用各航班不同时间的实时最低价格数据，及第四章中的KSPG路径搜索算法，并结合乘客出行的历史订单数据，包括乘客的机票、高铁订单，对乘客搜索的出行路径进行推荐。使用真实的价格、订单数据，还原乘客购买时可选的路径，充分利用数据间的关联性，以更加真实的情况对路径搜索和推荐进行实验，并对推荐结果进行分析。

* + 1. 实验数据和相关设置

首先为了直观上能清楚的理解数据及属性，下面解释和分析数据集的字段描述、属性和分布情况。历史订单数据包括国内直飞航线和部分高铁的历史订单数据，所有订单数据均来自于国内某家大型OTA。国内直飞航线的订单时间为2013年1月至2015年7 月，数据中各航线订单数呈长尾型，共包含198个城市间的航线订单数据。

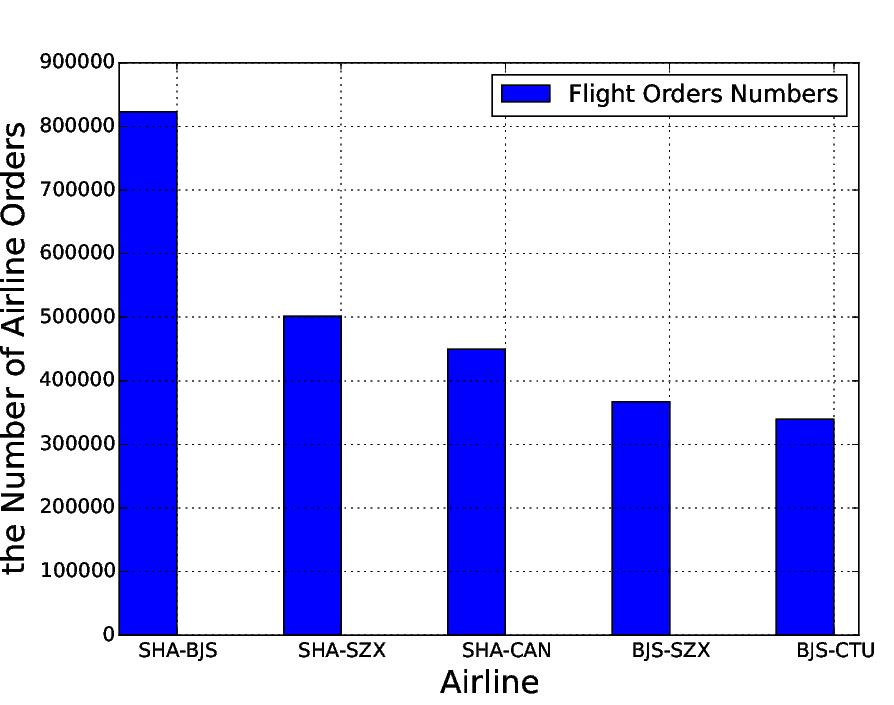


图5 - 1 行程的搜索占比

Figure 5 - 1 Route Searching Percentage

航线可分为商旅、旅游等不同种类。同时避免渠道商、代理商影响推荐的准确性，故将总数超过10个订单的乘客从实验数据集中剔除。图5 - 1中用直方图列举了订单数最多的五条航线分别是：上海-北京、上海-深圳，上海-广州、北京-深圳，北京-广州，其中上海-北京航线的订单量最多，单程的订单总数超过了80万张。订单的主要字段和含义列举在表5 - 1中，每条订单由id为唯一标识。

|  |  |
| --- | --- |
| **字段名称** | **字段意义** |
| Id | 每条直飞订单的唯一标识属性 |
| 出发城市 | 航班的出发城市 |
| 到达城市 | 航班的到达城市 |
| 出发机场 | 航班的出发机场 |
| 到达机场 | 航班的到达机场 |
| 出发时间 | 航班的出发时间 |
| 飞行时间 | 航班总共的飞行时间 |
| 预定时间 | 预定该航班机票的时间 |
| 航司 | 航班所属的航空公司 |
| 机型大小 | 乘坐的飞机机型：大、中、小 |
| 价格 | 机票的票面价格 |

表5 - 1 飞机订单数据格式

Table 5 - 15 - Format of Flight Orders

高铁的历史订单与国内直飞航线订单相关联，时间为2014年12月至2015年7月，其中主要字段列举在表5 - 2中，其中本论文中实验用到的高铁订单id均在国内直飞航线的订单中出现，即该订单中的乘客，选择的出行方案是先乘坐高铁再坐飞机，或先坐飞机再通过高铁到达目的地。此种情况是当转乘城市与出发城市或到达城市非常近时的一种出行选择，从传统的业务经验上分析，大部分的乘客是价格、时间敏感的，在确定行程且时间宽裕、允许的情况下，大多数乘客会选择价格合理，且满意的出行方案。

|  |  |
| --- | --- |
| **字段名称** | **字段意义** |
| Id | 订单id号，是每条订单的唯一标识 |
| 出发城市 | 高铁的出发城市 |
| 到达城市 | 高铁的到达城市 |
| 出发高铁站 | 高铁出发的车站 |
| 到达高铁站 | 高铁到达的车站 |
| 出发时间 | 高铁的出发时间 |
| 乘坐时间 | 高铁的乘坐时间 |
| 价格 | 高铁的票面价格 |

表 5 - 2高铁订单数据格式

Table 5 - 2 Format of Highway Orders

飞机+高铁的出行订单数据集共涉及212个城市，包括高铁站和机场所在的所有城市，共组成4083条行程，已售的订单总量是长尾型分布，如图5 - 2，其中最多的是北京到上海，占总订单量的2.9%。订单总量前10条行程中的城市已在图5 - 3中用红点进行标注，共22个城市。一般情况下，换乘的高铁站与出发城市或到达城市距离较近，例如：上海到北京，分别可以从石家庄、天津、杭州、南京高铁换乘飞机；北京到广州的行程，可以从深圳、石家庄、天津换乘。而乘客一般都是价格、时间敏感的，因此乘客在允许的时间范围内，愿意选择支付较低的价格，选择飞机+高铁的组合方式出行。

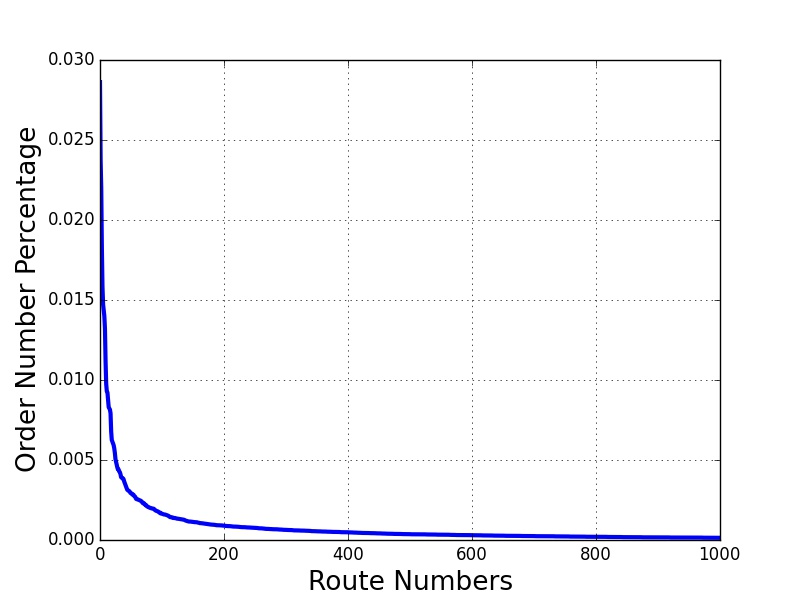


图 5 - 2飞机+高铁订单量统计

Figure 5 - 2 Flight & High-speed Train Order Distribution

在4.4小节中，我们在6个城市对间评估路径搜索算法KSPG的计算效率进行，本章中我们针对上海-北京、北京-重庆、上海-三亚三组城市进行实验，其中上海-北京属于商业性质，上海-三亚属于旅游性质，而北京-重庆二者兼顾。实验中，使用真实的订单数据评估出行路径推荐算法，乘客可以选择直飞航班或直达



图5 - 3主要换乘车站分布图

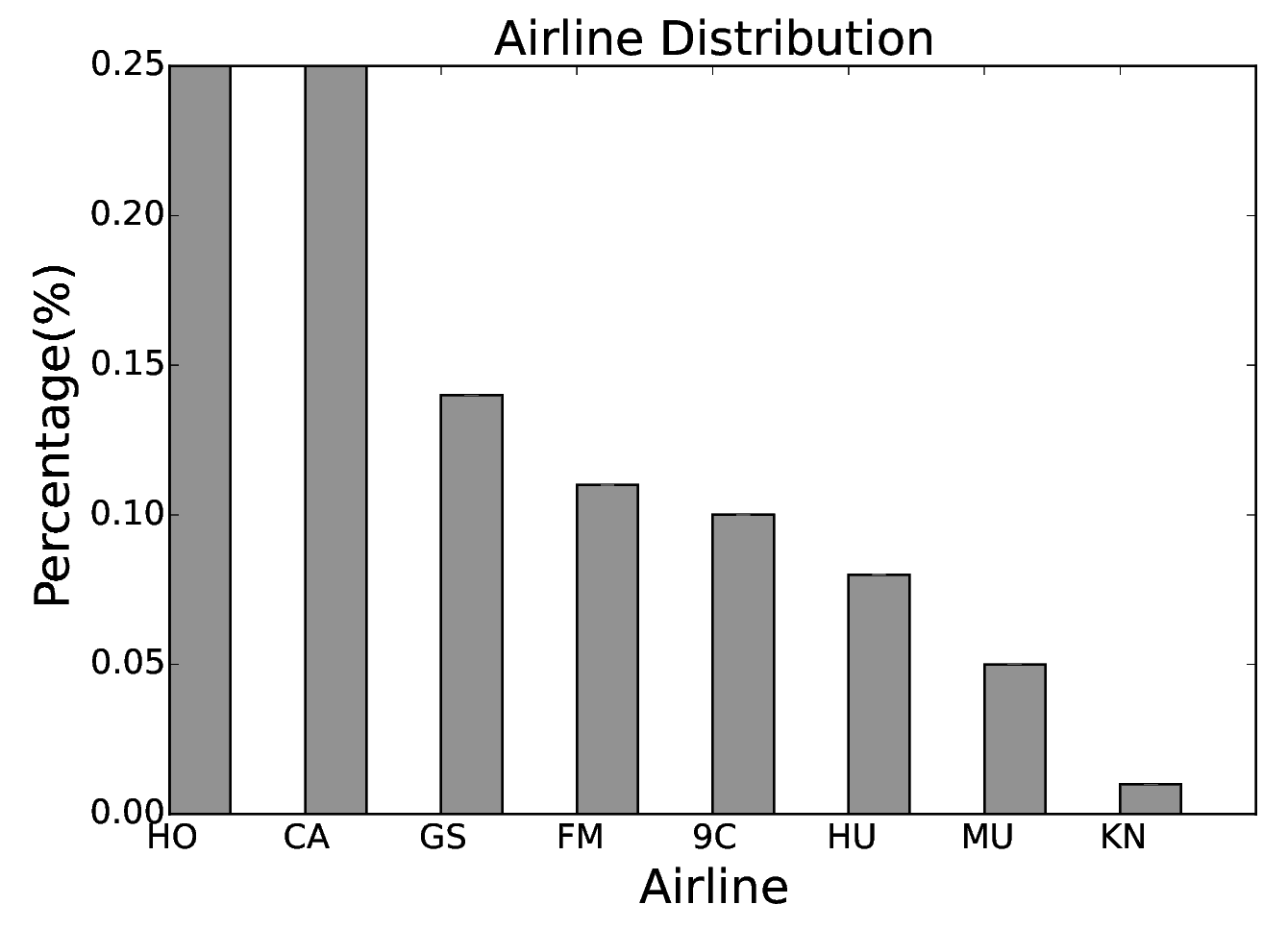
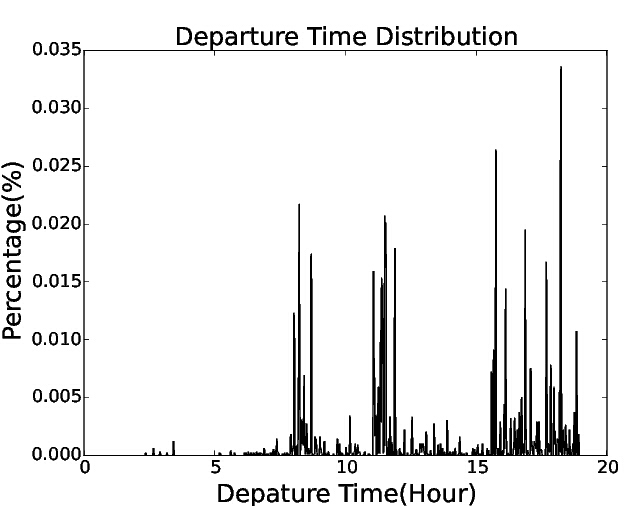
Figure 5 - 3 the Transfer Station Distribution on the Map

的高铁，也可先乘坐高铁转乘飞机，或先坐飞机再通过高铁到达目的地。我们从数据集中筛选出2000个至少3次购买飞机+高铁订单的乘客进行实验，其中乘客和路径共包括7个相同的特征：航司、出发时间、到达时间、总耗时（分）、总票价（元）、换乘次数、换乘总时间（分），其中离散特征为航司、出发时间、到达时间和换乘次数，其余为连续属性。

为了可以模拟真实的路径搜索购买场景，我们使用在4.4.1小节中介绍的飞机、高铁实时价格数据，根据历史订单中的出发-到达城市、价格和出发时间日期等信息，还原历史订单购买时，乘客的路径搜索场景，更准确的对测试集推荐结果

进行评估。

以上海-北京的历史订单数据为例进行分析，中转为南京、杭州、石家庄和天津，从历史的飞机+高铁订单统计得到71.2%的乘客会从天津换乘，10.4%从石家庄，9.7%从杭州，只有8.7%的乘客选择从南京换乘。图5 - 4表示的是上海到北京所有飞机+高铁订单大致的数据分布情况，以出发时间、总票价、总耗时、航司四个方面为代表进行展示，纵坐标表示对应横坐标取值所占的百分比。从图中可以看出，总票价大体呈现正态分布，大致范围是200到800块，占比最高的应该是300至500元区间；出发时间大部分在白天，不超过晚上9点，而上午8点、中午12点及傍晚比例居多；总耗时大致在4-6小时左右；共有8家航司可供选择，其中吉祥航空（HO）和中国国际航空（CA）比例最高。因此实验中我们采取两种评估方案：1）只使用天津一个换乘点进行实验；2）使用全部4个换乘点，根据历史数据分布，生成飞机+高铁订单数据作为补充数据集，与真实的订单数据共同进行实验。对于其他两组实验的换乘城市，可以采取同样的方法进行数据补充和实验。

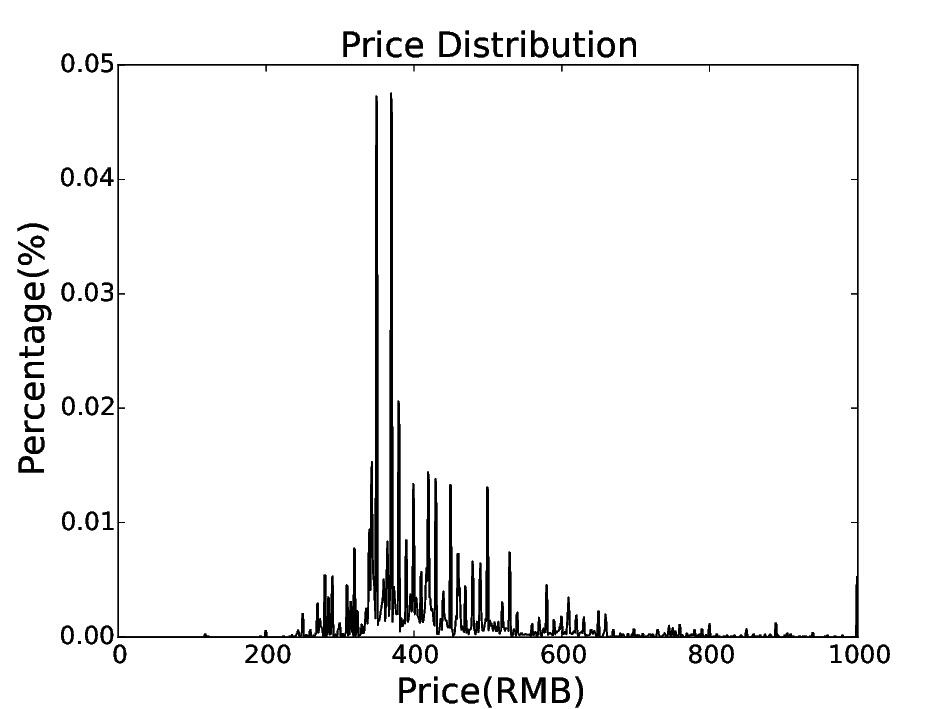
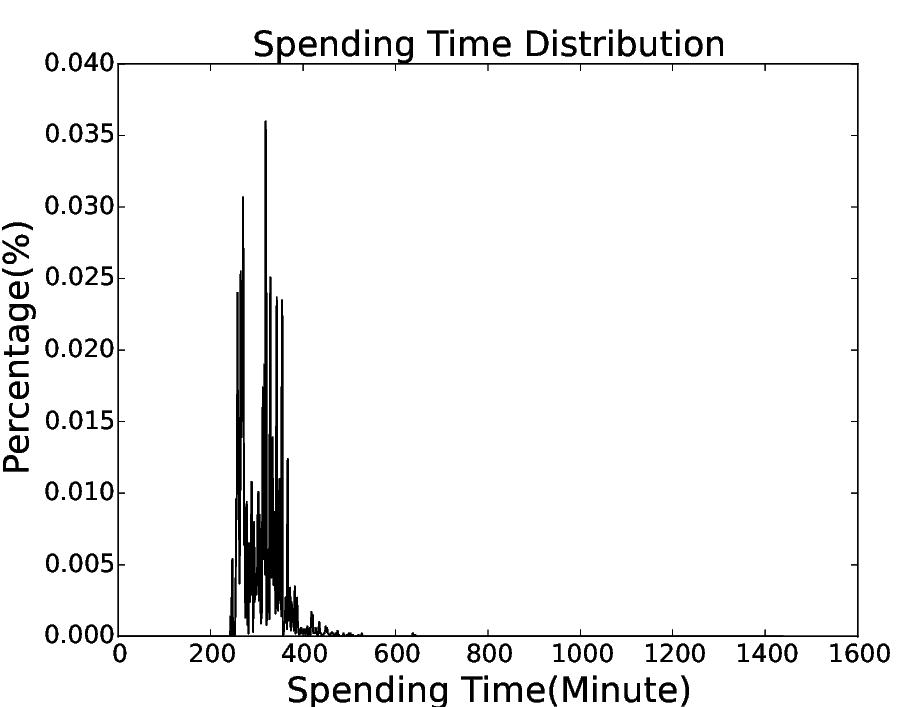
 

图5 - 4南京、杭州、石家庄换乘订单数据分布图

Figure 5 - 4 Distribution of History Orders Which transfer in Nanjing, Hangzhou and Shijiazhuang

本章中，仿真实验首先将历史订单数据划分为两部分，2015年4月30日之前的所有订单数据作为推荐算法训练集，而2015年5-7月的飞机+高铁订单数据作为测试集，测试集中每个订单都可以视为乘客的一次查询，由于机票价格在一天的时间内浮动较小，因此根据订单的出发、到达城市和飞机、高铁实时价格数据，构建交通换乘网络，每条边的权重设置为价格（元）和时间（分），网络中换乘车站间产生的费用凭借经验值确定，同时消除量纲影响将权重标准化，使用KSPG搜索前条备选路径，根据该乘客训练集中的订单数据，由推荐算法返回排序后的路径作为最终的推荐结果。

* + 1. 推荐评价指标

在基于排序的隐式反馈推荐系统中，我们无法获得乘客对推荐路径真正的评分和满意度，因此关于评分准确度或者相关性等评价指标，在此场景下均不适用。所以为了评估推荐算法的准确度，首先要制定评价指标。

测试集订单都是乘客真实的购买记录，评价指标则是用于衡量推荐路径列表和乘客订单的统一程度。推荐中，路径与传统的物品并不相同，推荐路径集合是实时搜索反馈的结果，而乘客只会购买列表中最满意的一条。乘客在选择过程中，耐心是有限的，乘客的满意程度通常与路径在推荐列表中的位置有很大关系，而推荐的准确率可以定义为订单路径在推荐列表中排序的序号，因此我们根据路径推荐问题，在实验中使用MAP（Mean Average Precision）作为评价指标，定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (5-15) |
|  |  | (5-16) |

其中表示测试集中乘客数量，是乘客测试订单的数量，表示订单在推荐条路径中的排名，即是2.2.3小节中的ARS评价指标，若订单路径不在推荐列表中，则为。

在定义的评价指标中，乘客的平均准确率是所有测试订单排序中序号的均值，而所有乘客的MAP是乘客平均准确度的均值，且有MAP越小，则乘客在路径搜索时满意度高的路径排名越靠前，推荐结果越好，因此可以很好的反应乘客的偏好和推荐路径排序之间关系。

* + 1. 结果分析

利用上述所说的训练、测试数据集和评价指标，我们对本章中基于top-n相似乘客的信息熵路径推荐算法进行实验和评估。在推荐算法实验中，备选路径是确定值，由KSPG算法搜索出条时间、价格最优的路径作为推荐备选路径，当没有限制时，

表示对所有可行路径进行推荐排序。

首先，分析特征的信息熵。图5 - 5是从历史飞机+高铁购买订单中筛选出的30000

个乘客，每个乘客至少有三次历史订单记录，对每个乘客的离散特征航司和连续属性价格敏感度进行信息熵的计算分析，横轴是信息熵的取值，纵轴表示信息熵值对应的乘客数量。左图中可以看出约1100个乘客在航司特征上信息熵值为0，从式(5-3)中可知表示此类乘客出行购买的均为同一家航空公司，在该特征上偏好十分明显，而其余乘客的信息熵大部分在0.2至0.8范围内；而右图为价格敏感度的信息熵，属于连续属性，根据式(5-4)可知，几乎没有乘客信息熵为0，而大部分的信息熵在0.2至1范围内，若信息熵为1则表示所有订单的价格敏感度取值完全相同，在计算特征相似度时，该特征在路径上没有区分度。我们设定特征的偏好阈值，当计算的信息熵值大于此阈值时表示乘客在该特征上有偏好，共对航司、价格、总耗时、出发时间四个特征进行计算，图5 - 6是30000个乘客的特征偏好个数分布图，约5000个乘客在四个特征上都有偏好，90%的乘客至少含有一个特征偏好，而只有一个特征偏好的乘客数量最多。因此基于以上分析，信息熵可以作为乘客特征的衡量指标，且基于信息熵的权重可以很好的量化乘客在特征上的偏好程度，为路径的推荐奠定基础。

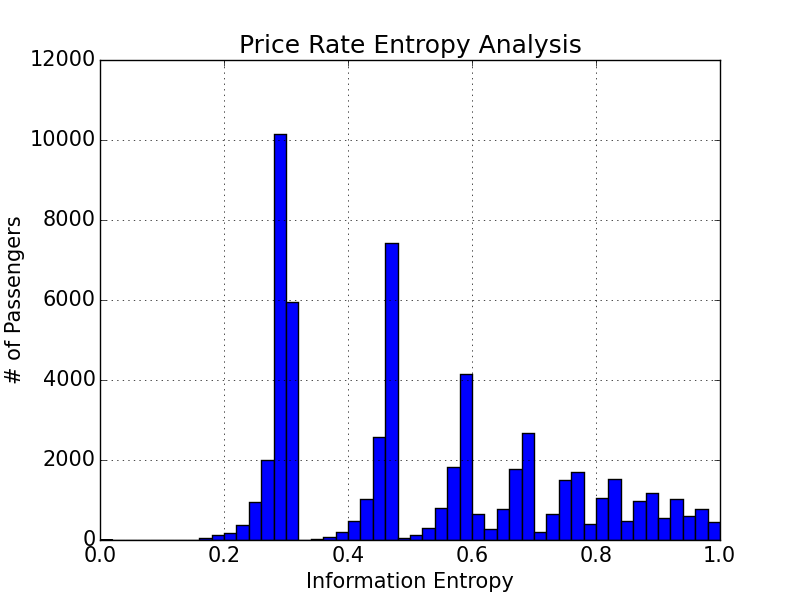
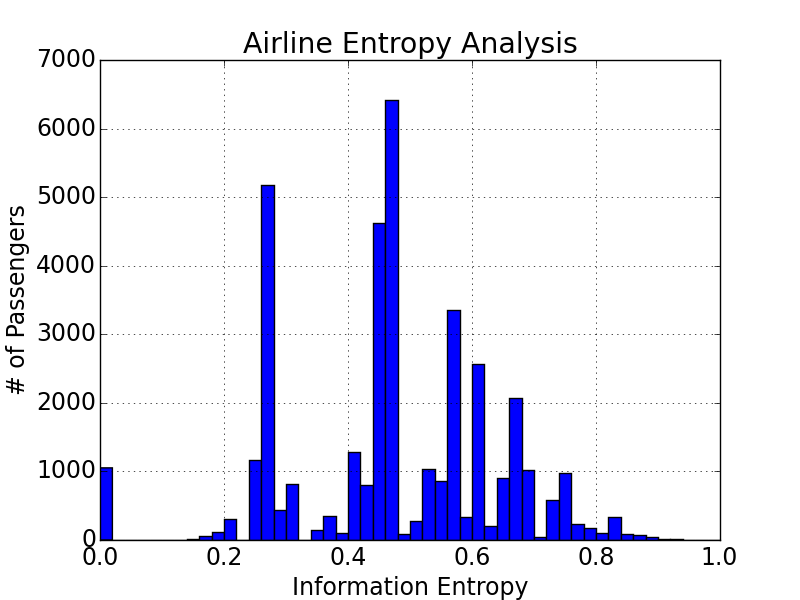


图5 - 5 航司、价格敏感度特征的信息熵比较

Figure 5 - 5 Comparison of the Information Entropy on Airline and Price Rate

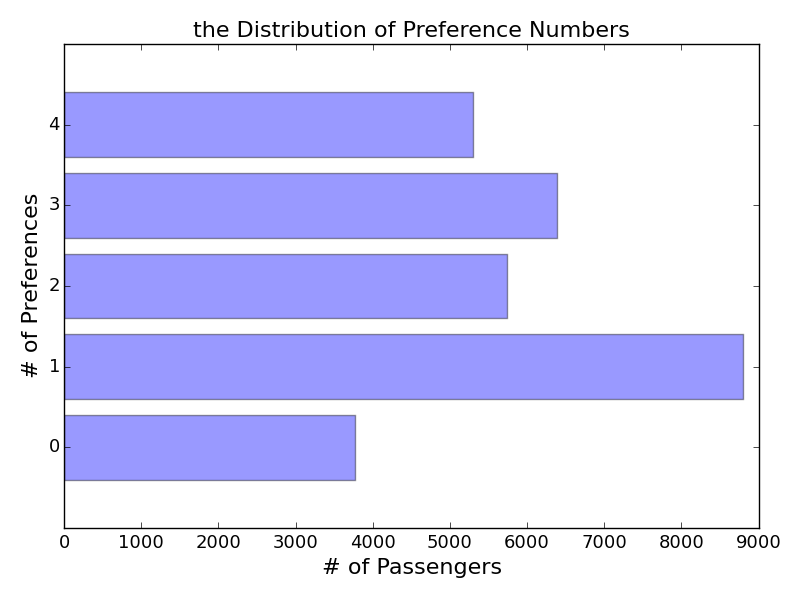


图5 - 6 乘客偏好个数分布

Figure 5 - 6 the Distribution of Preference Numbers

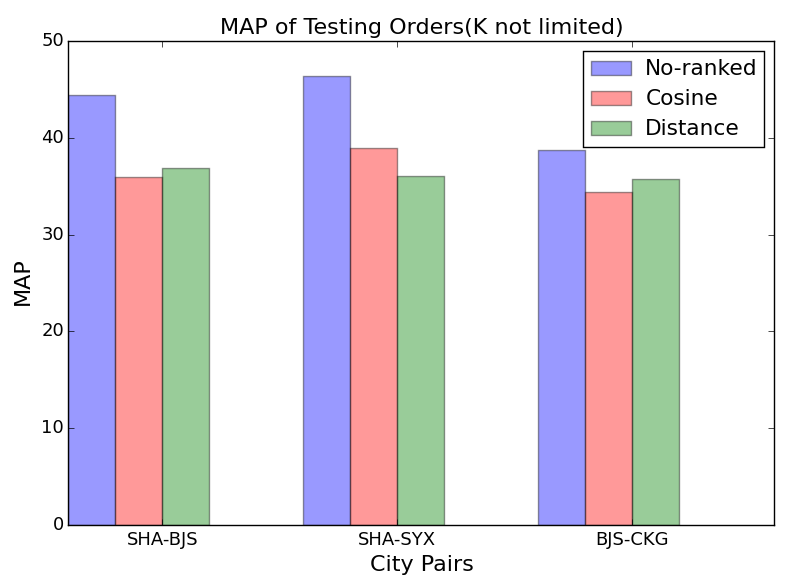


图5 - 7 基于信息熵的路径推荐算法结果MAP比较

Figure 5 - 7 MAP of Route Recommendation Based on Information Entropy

因为测试订单是乘客自己对路径进行组合、筛选和够买，而由于高铁班次、航班数量众多在，造成信息过载、可行路径增多，乘客的选择很大可能并非当时的最优选择。图5 - 7中展示了三组城市间路径推荐算法5.1的MAP结果评价值，其中紫色表示为KSPG算法中路径的生成顺序即是推荐顺序，按照时间、价格最优依次排序，此结果作为推荐算法评估的基准值；红色表示推荐算法使用式(5-12)计算连续特征的相似度，即基于信息熵系数的余弦相似度，图中标签简写为Cosine；绿色表示推荐算法使用式(5-13)计算连续特征相似度，基于信息熵的距离相似度，图中标签简写为Distance。从柱状图的结果上我们可知，在三个城市对间路径推荐算法都有较低的MAP，即乘客在搜索时，可以通过浏览较少的路径，查找到满意的结果。同时，对比上海-北京和上海-三亚两条行程后发现，在上海-北京的推荐结果上距离相似度优于余弦相似度，而上海-三亚行程上相反，主要原因可能在于上海-北京的出行商务出差居多，且机票的动态特征波动较小，基于信息熵的余弦相似度可以较好的衡量路径间相似度。从结果对比中可以看出，乘客在路径选择时，并不一定是完全倾向于价格、时间最低的路径，乘客最终选择的路径与航司、出发时间、到达时间等特征也有密切的关系。

在图5 - 8中，我们对基于top-n相似乘客的信息熵推荐算法进行评估，图中横坐标表示的是搜索窗口，纵坐标是算法评价指标MAP值，横坐标的含义在第四章实验

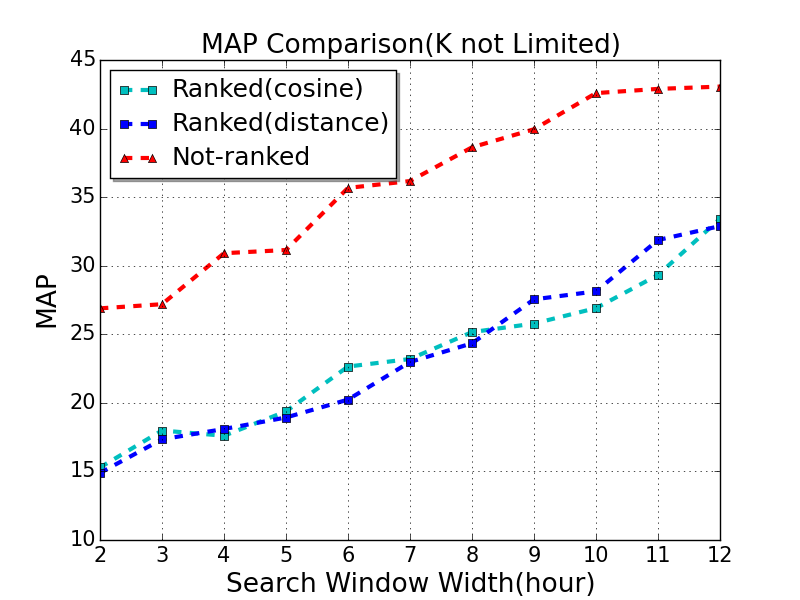


图5 - 8基于top-n相似乘客的路径推荐算法结果

Figure 5 - 8 Top-n Nearest Neighbor Route Recommendation Algorithm Result Analysis

中进行过详细说明。从图表中都可以看出，对比于未排序路径的基准值相比，路径推荐算法可以取得较好的推荐效果。从图中也可看出，随着搜索窗口的增大，MAP逐渐升高，代表乘客满意度逐渐下降，原因是搜索窗口的增大则可行路径数量增多，而存在一部分路径更符合乘客的偏好，导致该部分路径在推荐列表中排名靠前，从而测试订单在推荐路径列表中靠后，排名下降导致MAP升高。而从表5 - 3中我们从推荐路径的数量进行评估，通过比较可知值的大小会影响最终的推荐效果，因为可行路径的解空间十分庞大，乘客在购买订单时无法有效的解决信息过载问题，筛选出最满意的路径。同时，对比5.3.1小节中提出的两种基于信息熵的相似度计算方法，在推荐的MAP值上差距不大，但5.3.3小节中基于top-n相似乘客的IERA算法明显优于IERA算法。因此可知，通过查找并使用前个相似乘客的所有历史订单数据进行推荐可以一定程度上解决历史订单稀疏性的问题，且能够进一步提升乘客的路径推荐效果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 算法  K取值 | Non-ranked | IERA-Cosine | IERA-Distance | SIERA-Cosine | SIERA-Distance |
| K=50 | 35.14 | 28.12 | 27.96 | 25.41 | 25.86 |
| K=100 | 37.83 | 30.90 | 31.02 | 28.57 | 27.16 |
| K=150 | 38.95 | 33.51 | 32.28 | 30.24 | 29.34 |
| K not limited | 43.06 | 36.42 | 36.24 | 33.43 | 32.89 |

表 5 - 3 路径推荐算法MAP比较

Table 5 - 3 Comparison of the Route Recommendation MAPs

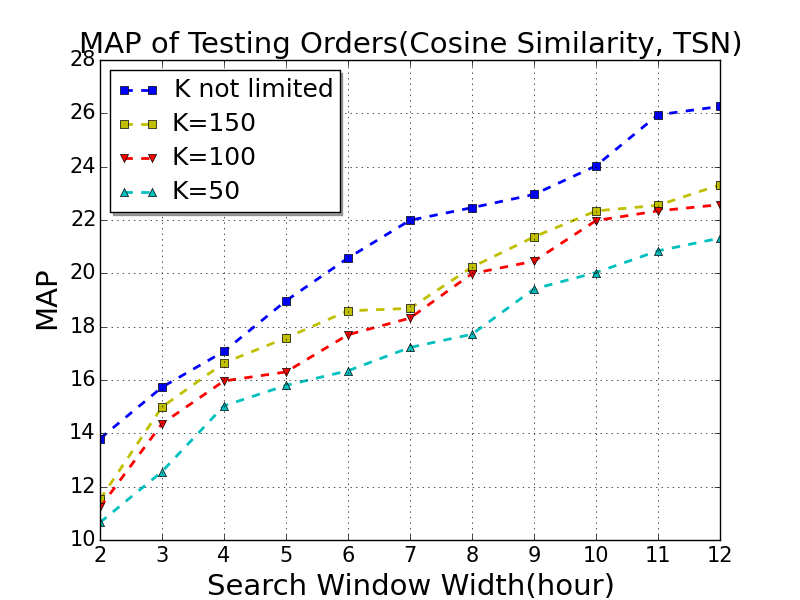
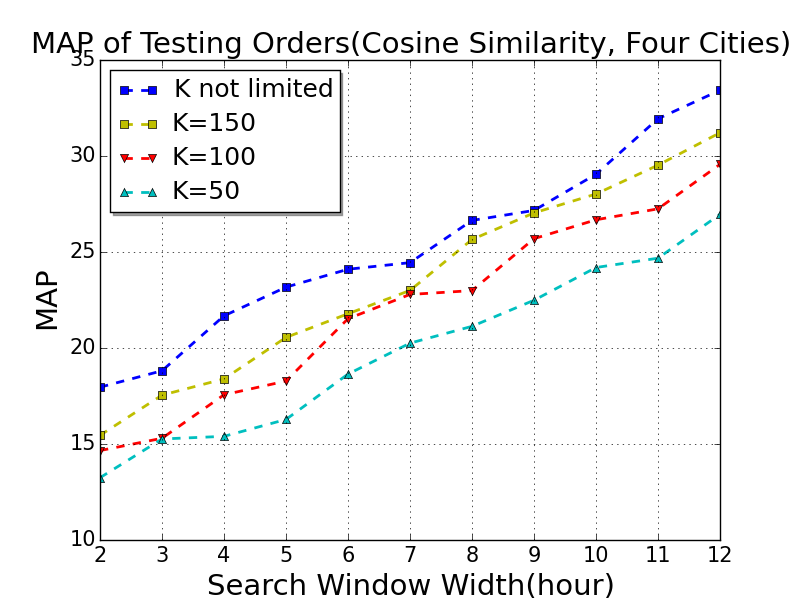
 

图5 - 9 上海-北京路径推荐算法结果比较

Figure 5 - 9 MAP of the Route Recommendation between SHA and BJS

为了进一步评估基于top-n相似乘客的信息熵路径推荐算法，即算法5.2，我们选择上海-北京的实验结果进行详细分析。在图5 - 9的两幅图中，横坐标表示搜索窗口，纵坐标为MAP值。其中左图表示第一种实验方案，即只有天津作为中转站；右图为第二种方案，即通过根据历史订单数据分布扩充数据集，使用四个中转站进行实验。两张图对比后可知，由于中转站数量的增多，导致可行路径数量的增长，路径解空间增大，使得乘客很难选到最满意的出行路径，导致MAP值明显的升高。从两幅图中均可以看出，随着搜索窗口的增大，可行路径数量的增多，MAP升高，客户客满意度逐渐下降。同样推荐路径数量的增加也会导致推荐结果MAP的升高，但图中的趋势可以看出，影响程度小于搜索窗口的改变，由于路径是根据时间和价格权重从小到大依次生成，从业务经验上可以理解为大部分乘客还是时间、价格敏感型，在满足偏好的同时，选择价格较低的路径。

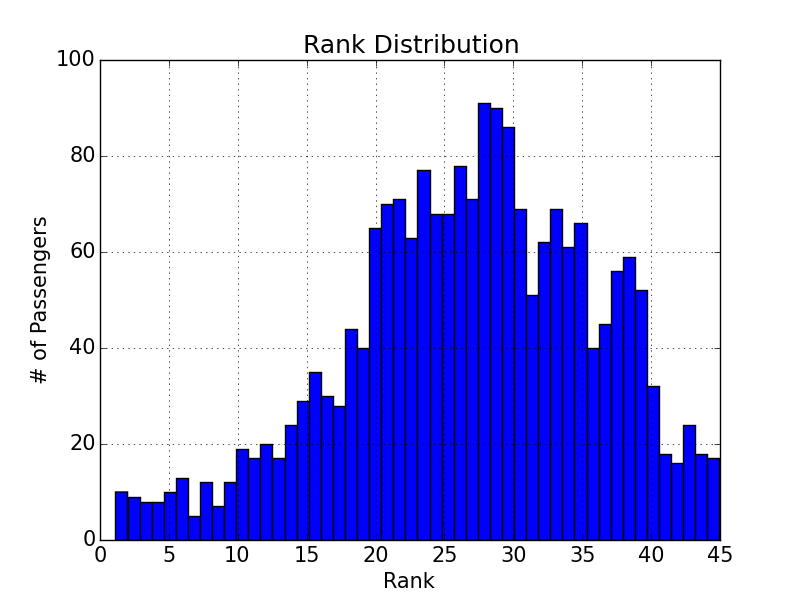


图5 – 10 上海-北京推荐算法rank分布

Figure 5 - 10 Distribution of Testing Orders’ Rank between SHA and BJS

图5 - 10为上海-北京路线所有订单在推荐列表中的位置统计值，实验共选择四个中转车站、且不限制搜索窗口时，路径推荐的平均准确度为26.82，横坐标表示值，纵坐标表示每个值对应的乘客数量，从图中可以看出大部分的订单数据在推荐列表中在20至40的位置，在30左右的乘客数量最多，但是仍有部分乘客测试订单的推荐效果不是非常理想，推荐*rank*较大，主要是因为乘客的特征偏好并不明显，无法很好的根据历史订单信息进行推荐。

* 1. 本章小结

本章提出一种基于信息熵的路径推荐算法，通过分析乘客历史订单数据的连续、离散特征信息熵，计算分析乘客在不同特征上的偏好权重，定义路径-路径相似度计算方法，并对KSPG算法中搜索的前条备选路径进行推荐排序。同时针对乘客历史订单稀疏性问题，定义乘客相似度函数，提出一种基于top-n相似乘客的信息熵路径推荐算法，对路径的实时搜索结果进行路径推荐。

在5.4小节中，使用飞机、高铁的实时价格数据和真实历史订单数据，构建历史订单的真实搜索场景，对乘客的信息熵及推荐算法进行实验，评估分析推荐效果。

1. 路径搜索的缓存策略

以上两个章节中，详细介绍了路径搜索和推荐算法，并根据历史订单数据对算法结果进行评估，但是随着查询流量爆发式增多，为了提高服务质量、保证乘客满意度，通过缓存技术减少搜索的响应时间尤为关键。本章节中，首先对行程搜索数据进行统计研究，针对出发-到达城市的搜索流行度，设计行程的缓存替换机制，并从搜索命中率角度评估缓存结果。再者，分析出发-到达城市间存在多个换乘节点的情况，从路径的推荐热门程度和抽样角度设计路径缓存策略，乘客查询时，系统先通过和缓存路径比较，筛选满意度高的换乘点再进行实时路径搜索，减少查询时间，并使用历史机+铁订单评估路径缓存策略。最后，由于路径是实时动态变化的，所以分析乘客搜索的时间间隔分布规律，定义行程缓存路径信息的新鲜度计算方法，根据历史搜索频次，以最大化缓存路径的平均新鲜度为目标，制定缓存路径的更新策略，保证路径信息的实时有效性。

* 1. 行程搜索数据分析

数据源同样来自于国内某家大型OTA，表6-1中展示了网站出行搜索记录包含的字段，每个字段的具体含义是：ip表示搜索的ip地址，SearchTime代表搜索时间 ，DepartureTime表示乘客希望的出发日期，Acity和Dcity表示出发到达城市。

|  |  |
| --- | --- |
| **字段名称** | **字段意义** |
| Ip | 搜索ip |
| Dcity | 行程出发城市 |
| Acity | 行程到达城市 |
| Searchtime | 行程搜索时间 |
| DepartureTime | 行程出行时间 |

表6 - 1出行搜索数据格式

Table 6 - 1 Format of Route Search Data

任意选择连续10天时间的搜索数据进行统计分析，总搜索量是21426377次，搜索行程共35038条，城市共358个。图6 - 1中，左图为搜索量最多前5000条行程的搜索量分布图，整体呈长尾分布，其中搜索量排名前三的行程分别是：上海到北京，北京到上海，及北京到成都，搜索量分别为144366、131865、131044；右图中展示了行程搜索数据的分布情况，其中蓝线表示行程数和搜索比例的关系，例如：前5000

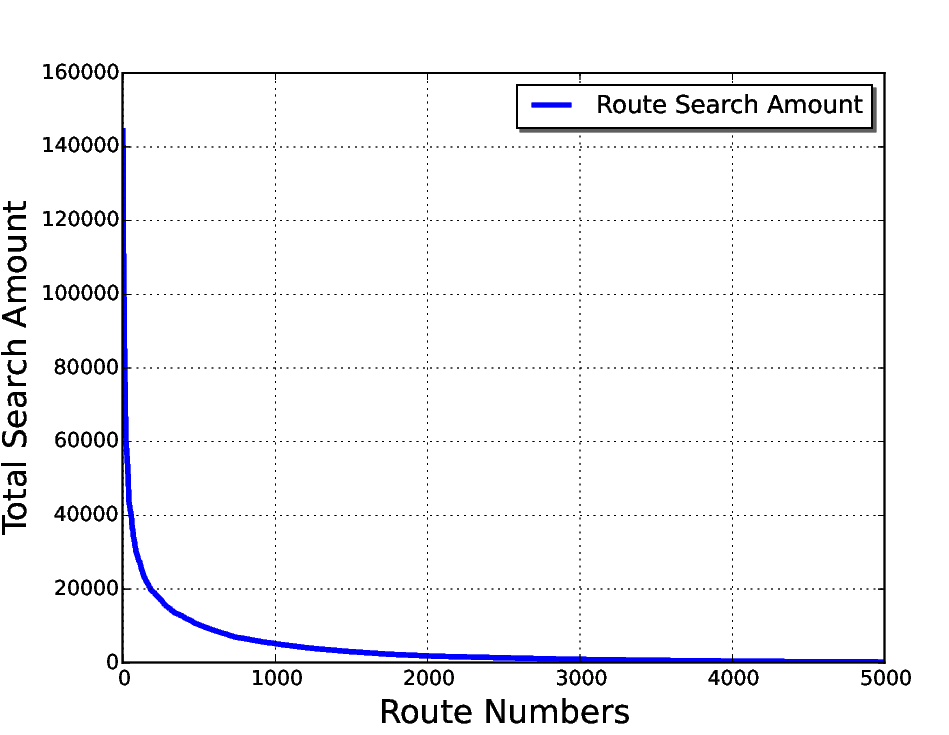
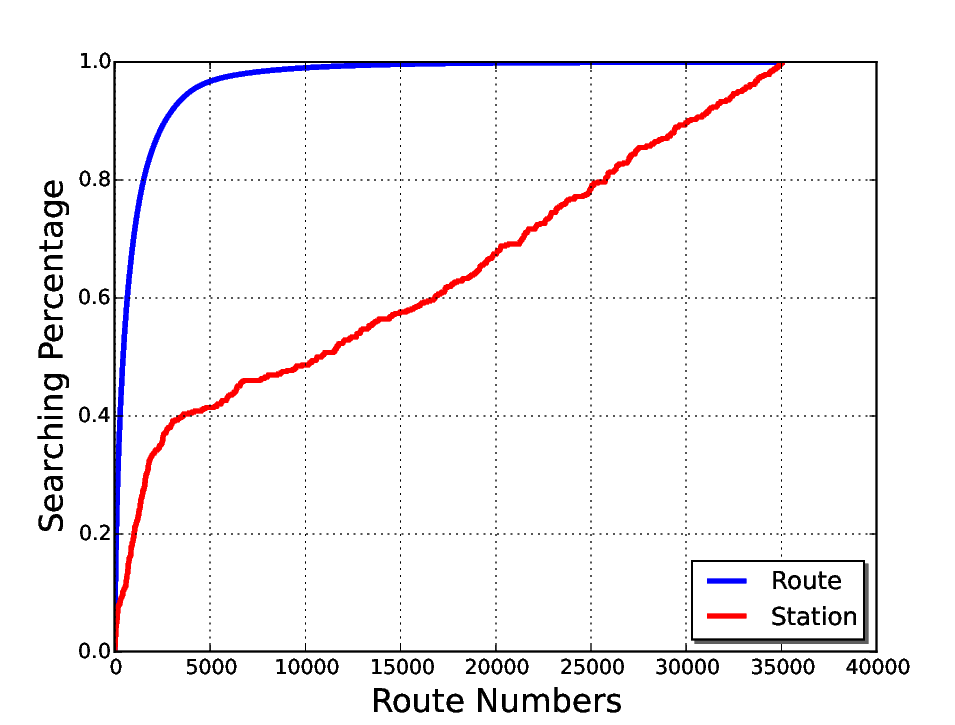
 

图6 – 1 行程的搜索分布情况

Figure 6 - 1 Route Searching Amount Distribution

条行程（约占行程总数的14%）的搜索量占总搜索量96.7%，图中红线表示城市的个数，大致呈线性增长，前5000条行程的斜率明显大于后30000条。

* 1. 行程缓存模块

由于行程路径的搜索结果是实时动态变化的，如果将所有行程数据全部缓存，既无法保证实时性，也浪费了硬件资源，而且在实际情况中，硬件资源的容量也不是无限的。优秀的缓存替换机制需要预测行程将来的搜索频率，从而将搜索概率较大的行程缓存。本小节中，针对定容的缓存空间，以城市对间的路径查询结果为单位，基于搜索热度和经典的替换算法LFU，提出一种基于时间窗口的缓存替换机制。

式(6-1)表示搜索查询时系统的平均响应时间，其中表示搜索命中时查询响应时间，表示搜索未命中请求服务器的响应时间，代表缓存命中率，计算如式(6-2)，为搜索的总次数，为cache命中的次数。如果缓存命中率越大，则系统响应时间越少，搜索性能提升，故使用缓存的命中率作为衡量指标，既可以减少乘客搜索的时间，也可以降低服务器的负载。

|  |  |  |
| --- | --- | --- |
|  |  | (6-1) |
|  |  | (6-2) |

缓存容量是固定的，根据第一小节中的分析，我们暂定缓存容量最多为5000个行程数据，即出发-到达城市对。当内容占满缓存且未命中时，需要根据替换机制选择被替换的对象，替换相应的缓存内容，可以使用操作系统内存管理中经典的替换算法对缓存的行程城市进行替换。但是由于行程的搜索是高密度的，每次没有命中的行

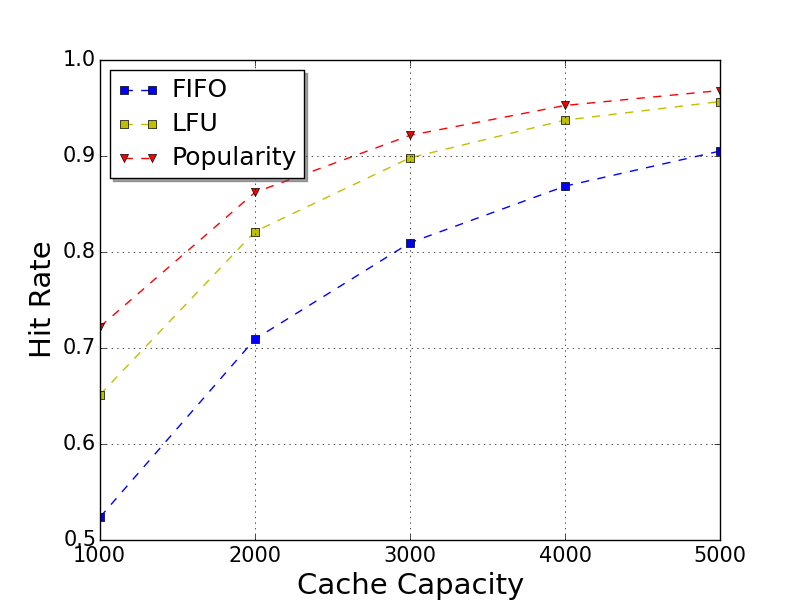


图6 – 2 缓存替换机制命中率分析

Figure 6 - 2 Hit Rate Comparison of Cache Strategy

程查询都需要被替换，缓存高频率的换入换出也会导致效率低下。

因此本小节提出基于时间窗口的LFU缓存替换机制，该机制中首先定义缓存替

换的时间窗口，统计在每个时间窗口内不同行程的查询次数，在每个时间窗口结束时，根据缓存容量和窗口内的查询次数由大到小更新行程的出发-到达城市，对于每次搜索的行程，如果出发-到达城市在缓存中，表示命中，否则表示未命中。

使用行程搜索数据对缓存替换机制进行实验模拟，时间窗口为5个小时，即每隔5个小时统计查询次数作为搜索热度(popularity)，结果如图6 - 2，横坐标为缓存的行程数量，从图中可以看出各缓存机制的命中率都很高，且均随着缓存容量的增大而提高，而基于时间窗口的LFU缓存替换机制命中率均略高于经典的替换算法，当缓存行程数量为5000时，基于时间窗口的LFU缓存替换机制命中率高达97%，说明绝大对数乘客搜索的行程都十分相似。

* 1. 路径缓存模块

由于交通的迅速发展，热门城市间铁路班次、航班也越来越多，若考虑中转，可组合成百上千种出行路径。因此对于第一小节中缓存的所有出发-到达城市对，缓存所有可行路径并不现实；同时为了增加查询效率，考虑以下场景：从城市A到达城市B，一般需要一次数据库查询，如果考虑中转城市C和D，每加入一个中转则增加2次查询，实际应用中一次数据库的查询时间大概是500ms，假设根据第五章的路径推荐算法，计算ACB路径上前10条最优的航班、高铁组合，再查询D中转路径并计算路径的相似程度，但结果比ACB路径前10条最优路径的相似度都要低，查询中D中转站消耗了两次查询时间（AD，DB），但整体A到B前10条推荐路径结果并未发生改变。所以本小节中希望解决的问题是：在保证推荐路径的满意度前提下，如何减少中转城市数量，提高路径搜索效率。

* + 1. 路径缓存策略

我们假设乘客的查询是不间断的，因此暂不考虑缓存路径更新问题。基于上一位乘客路径搜索结果，不同中转站缓存有代表性的路径或信息，当下一位乘客查询时，首先根据缓存路径对中转站进行筛选，选择满意度最高的中转城市，并根据第四、五章的算法进行路径搜索和排序，选择最满意的出行路径。

表示基于某中转站搜索的路径集合，依据价格和总耗时之和由小到大排列，及KSPG算法路径搜索的结果。假设缓存路径的数量为，介绍三种缓存路径选择方法。

* 随机选择

最简单的方法是随机选择，计算效率高，但并不能包含路径的重要信息。

* Gibbs采样

所谓采样就是观察以什么样的概率分布发生什么样的事情。在第五章中，对出行订单统计量分析可知，价格、时间大致符合正态分布和均匀分布，因此可以将价格、时间看做随机变量，使用最小二乘法对路径价格拟合高斯分布的期望和方差，再使用Gibbs采样方法对价格、时间的联合分布进行采样，Gibbs采样中根据条件概率可以推导等式：

|  |  |  |
| --- | --- | --- |
|  |  | (6-3) |

即可使用式(6-4)生成价格、时间的联合概率分布采样点集合。

|  |  |  |
| --- | --- | --- |
|  |  | (6-4) |

* 基于推荐度的路径缓存策略

第五章中介绍了基于信息熵的路径推荐算法，推荐算法可以很好的预测乘客的行为。而对于所有乘客，路径推荐排序越高，则该路径的热门度也越高、被乘客选中的概率越大，所以具有较高的缓存价值。

定义6. 为乘客集合，为路径集合，表示乘客对路径的打分，是乘客的历史订单集合，为订单数量，则路径被乘客的推荐程度为：

|  |  |  |
| --- | --- | --- |
|  |  | (6-5) |
|  |  | (6-6) |
|  |  | (6-7) |

其中为乘客总人数，式(6-6)中相似度计算与第五章中路径相似度计算方法相同，具体见式(5-6)至式(5-13)，同时使用消除不同乘客评分标准不同的问题，因此路径被所有乘客推荐的程度为：

|  |  |  |
| --- | --- | --- |
|  |  | (6-8) |

乘客在路径搜索时，系统会将基于信息熵排序的路径推荐给乘客，同时推荐排序中的路径相似度又被转化为乘客对各路径的推荐程度，因此根据式(6-8)更新各路径的推荐程度。如果路径的推荐程度越大，说明对该路径满意的乘客越多，则将来此路径被选择的可能性就越大，被系统缓存的价值则越高。最后，系统将路径集合根据推荐程度由大到小依次排序，选择前条作为缓存路径。

* + 1. 路径缓存结果分析

根据路径缓存策略，每个中转站上都缓存一定数量的路径，当乘客查询时，根据缓存的路径对中转站进行筛选。实验中假设乘客查询是不间断的，因此使用机票、高铁的实时价格数据更新每个中转点上缓存的路径。同样，和5.4小节中实验的行程相同，选择上海-北京、上海-三亚、北京-广州三个出发-到达城市对，中转城市如表4 - 1所示，且实验中乘客至少有三次历史订单且至少包含一次飞机+高铁订单。

实验中对于每个飞机+高铁订单，首先使用乘客的历史订单，对每个中转站的缓存路径计算相似度，选择路径平均相似度最高的中转城市，如果该城市与乘客订单的中转站一致，则表示该订单命中。图6 - 3中对不同的路径缓存方法结果进行评估，分别为：1）随机选择；2）根据价格、时间总和由小到大选择前条路径；3）Gibbs采样；4）基于推荐程度的路径缓存。图中横坐标表示缓存路径占该中转站可行路径总量的比例，纵坐标表示飞机+高铁订单中转站选择正确的比例，从图中可以看出随

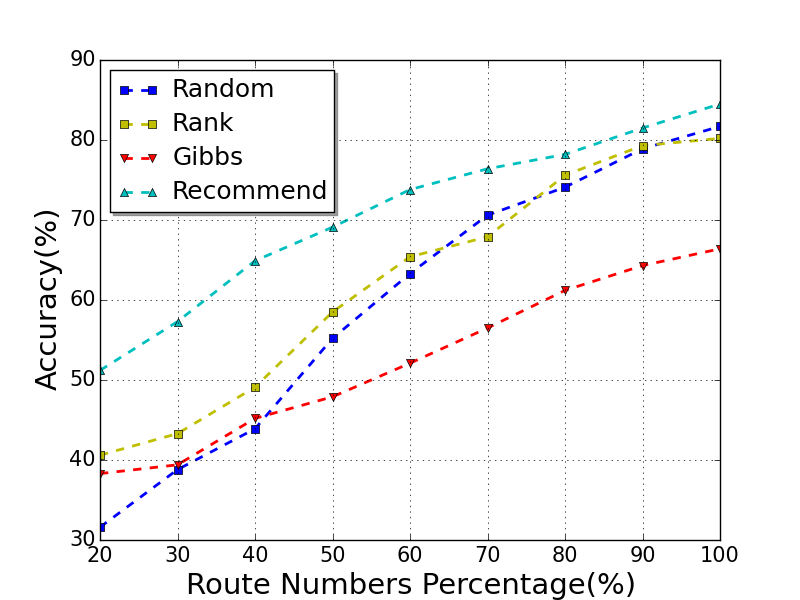


图6 – 3 路径缓存方法结果比较

Figure 6 - 3 Comparison of Route Cache Strategy

着缓存路径数量比例的增高，中转站选择正确的比例越高，而基于推荐程度的路径缓存策略即使在缓存路径数量较少时，依然有很高的比例可以选择到正确的中转站。

因此根据实验结果分析可知，在行程中对于每个中转城市缓存部分路径，可以在路径实时搜索前，有效的预判中转站满意度，并筛选满意度高的中转站对路径进行查询，减少路径搜索空间，在保证乘客满意度的情况下，提高搜索效率。

* 1. 路径缓存更新策略
     1. 路径缓存更新策略

在6.3小节中，我们假设乘客的查询是不间断的，但在实际应用中，乘客的查询是间断的，同时实时价格数据也在不断变化，所以如何更新缓存路径是保证服务满意度的关键。本小节中，首先定义行程路径信息的新鲜度函数，并以最大化平均行程路径信息的新鲜度为目标，提出路径缓存更新策略。

泊松分布中，事件到达的频率为，是下一次发生的时间间隔，的概率密度为：

|  |  |  |
| --- | --- | --- |
|  |  | (6-9) |

通过分析搜索数据如图6 - 4所示，图中横坐标表示查询时间间隔，纵坐标为对应时间间隔的查询比例，对于同一行程，所有乘客的查询时间间隔基本符合泊松分布。对行程查询频次统计后可知，基本符合Heaps定理[[[57]](#endnote-57)]，即虽然搜索总量大，但是对于单个行程的搜索量不会增长过快，大致搜索总量随时间呈线性增长。

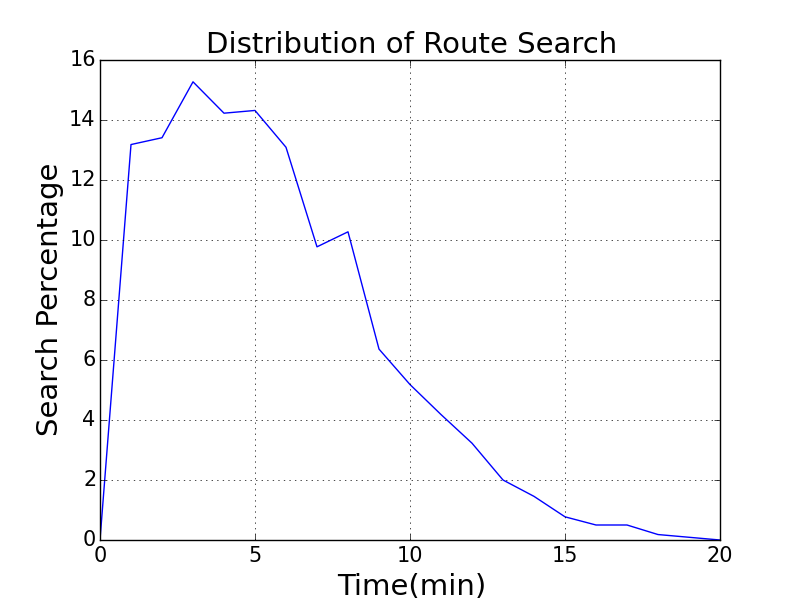


图6 – 4 行程搜索时间间隔分布

Figure 6 - 4 Time Interval Distribution of Route Search

在实际应用中，由于航班价格、高铁班次的实时性特征，缓存的路径会随着时间的变化为逐渐失效。假设系统共缓存条行程，其中行程的路径信息在刚更新缓存，此时新鲜度为1，定义在时间时新鲜度为，根据泊松分布，缓存在时间区间内更新的概率为：

|  |  |  |
| --- | --- | --- |
|  |  | (6-10) |

因此行程在时间，缓存路径的新鲜度期望为：

|  |  |  |
| --- | --- | --- |
|  |  | (6-11) |

定义向量代表每条行程在时间周期内的搜索频次，并归一化处理消除量纲影响，搜索频率可以记为。在6.2小节行程缓存中，我们定义了时间窗口，假设内行程路径更新次数为，对于行程更新时间间隔为，行程的平均新鲜度计算如下：

|  |  |  |
| --- | --- | --- |
|  |  | (6-12) |
|  |  | (6-13) |

向量也可以代表行程当前的搜索热度，定义表示在此时间内的平均新鲜度，而全部行程的平均新鲜度计算方法如下：

|  |  |  |
| --- | --- | --- |
|  |  | (6-14) |

每个时间窗口[0,]开始时，根据上个时间窗口的行程搜索频次更新向量，并重新计算调整行程路径的更新频次，优化目标是最大化平均新鲜度，使用拉格朗日乘数法对进行求解：

|  |  |  |
| --- | --- | --- |
|  |  | (6-15) |

其中结果即为行程更新缓存路径的间隔时间。

* + 1. 缓存路径更新策略结果分析

为验证缓存路径更新策略的有效性，使用真实的乘客搜索数据和实时机票、高铁价格数据，对各行程中飞机+高铁路径进行缓存更新策略的对比实验，使用连续24小时行程搜索数据，定义时间窗口为4小时，为了和第五章中的实验保持一致性，选择相同的三条行程进行实验，行程中每个中转站缓存20%的路径。

由于航班机票实时价格数据更新时间间隔为2小时，故根据当天机票价格最低、最高值，按照泊松分布随机产生行程上不同航班的价格数据改变时间，改变次数为3到6次。为验证缓存更新策略的有效性，时间窗口内共采用三种策略对缓存路径进

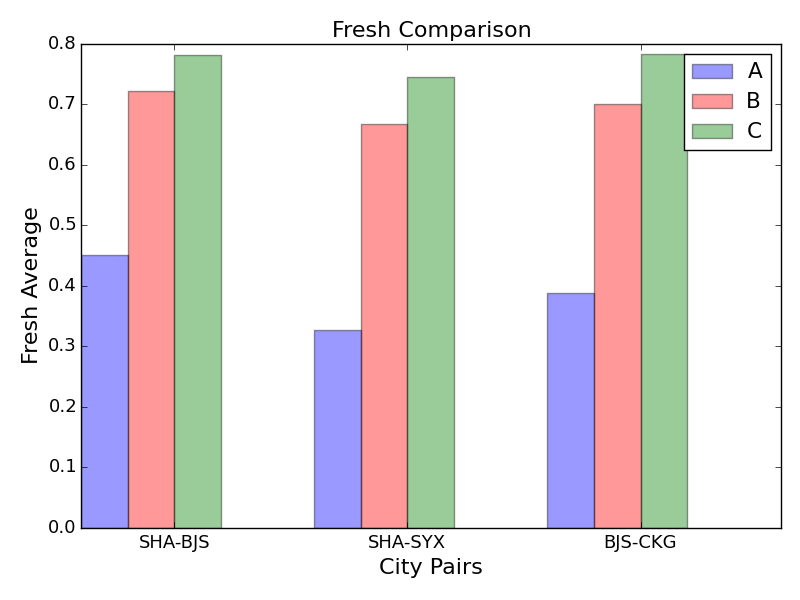


图6 – 5 缓存路径更新策略平均新鲜度结果比较

Figure 6 - 5 Fresh Comparison of Route Update Strategy

行更新：A算法所有行程的路径按照固定时间间隔对缓存路径进行更新；B算法以乘客搜索为驱动更新路径缓存，不主动对缓存路径进行更新；C算法是以平均新鲜度最大为目标，动态计算行程的缓存路径的更新频率。最终对比三个算法实验结果的行程路径平均新鲜度。

如图6 - 5所示，横坐标表示实验的三个出发-到达城市对，纵坐标为每个时间窗口内平均值，从图中可以看出使用固定时间间隔更新缓存策略时，路径的平均新鲜度最低，由于三条航线上的搜索量较大，因此算法B的平均新鲜度相比于算法C差距并不大，但是算法B是由乘客搜索驱动，导致会频繁度路径进行更新，而算法C在三条行程路径上平均新鲜度最好，说明根据上个窗口的行程历史搜索频次可以很好的反应搜索热度，每个窗口基于行程的热度，以最大化平均新鲜度为目标，调整时间窗口内缓存路径更新频率，保证缓存路径数据的实时准确性，从而可以保证乘客在实时路径搜索前，根据缓存路径有效的预判中转站满意度，并筛选满意度高的中转站对路径进行查询。

* 1. 本章小结

本章节共研究三个问题：

1. 针对出发-到达城市的搜索频率，定义各行程的流行度，并设计基于时间窗口的LFU缓存替换机制，从行程搜索命中率角度对缓存替换机制进行评估。
2. 针对出发-到达城市间存在多个换乘节点的情况，从路径的推荐程度和抽样角度设计路径缓存策略，乘客查询路径时，通过乘客偏好特征和缓存路径的相似度计算，筛选满意度高的换乘点进行路径查询，可以有效的节省路径搜索的时间且保证乘客满意度，最后使用历史机+铁订单评估路径缓存策略。
3. 针对不同行程的搜索时间分布规律，定义行程缓存路径信息的新鲜度计算方法，根据历史搜索频次，以最大化缓存路径的平均新鲜度为目标，制定缓存路径的更新策略，保证路径信息的实时有效性，并对实验结果进行评估。
4. 全文总结与展望
   1. 本文工作总结

本文中的主要研究工作体现在以下几个方面：

1. 本文基于城际间多种出行交通方式，首先介绍了面向乘客出行的城际个性化路径推荐模型，目标是乘客给定具体的出发日期、出发和到达城市后，在较短的时间内搜索并推荐该日期最符合乘客偏好的出行方案。整个模型主要共分为四个部分：系统与乘客的交互，路径的搜索、推荐和缓存部分，各部分内容之间相互联系。
2. 提出一种带约束的启发式路径搜索算法KSPG，使用有权有向图构建实时的换乘交通网络，每条边的权值为价格、时间等可叠加特征取值。在给定源点、终点和约束条件情况下，KSPG按照路径权值由小到大依次生成前条满足约束的最短路径，并作为推荐的备选路径，并在评估算法的搜索效率。
3. 提出一种基于信息熵的路径推荐算法，通过数据分析引入特征的信息熵定义，并针对乘客历史订单稀疏性问题，定义乘客相似度函数，基于top-n相似乘客的历史订单数据，计算连续、离散特征信息熵，计算乘客在不同特征上的偏好权重，并定义路径-路径相似度计算方法，通过计算相似度对条备选路径推荐排序，对路径实时搜索结果进行路径推荐，最后实验结果分，基于信息熵的特征权重可以很好的反应乘客偏好，高效快速的为乘客推荐与历史订单相似的出行路径。
4. 本文提出一种基于时间窗口的LFU缓存替换机制，在规定时间窗口内依据搜索频次更新缓存行程；并从Gibbs采样和路径推荐度角度，制定各中转站上路径的缓存策略，乘客查询时根据缓存路径对中转站进行筛选，提高路径搜索效率；最后针对搜索时间间隔分布，以最大化缓存平均新鲜度为目标，制定缓存路径的更新策略，保证信息的实时性，在不降低乘客服务满意度的情况下，提高搜索效率。
   1. 未来工作展望

本文中主要对于城际间个性化路径推荐问题进行了研究，通过对历史订单、实时价格数据的分析，在保证搜索效率和乘客满意度的前提下，提出路径的搜索、推荐和缓存算法，通过实验分析均取得较好的推荐结果。但时间所限，对于许多问题未能更加深入细致的研究和探索，因此，在现有工作基础上，对未来潜在的研究工作和方向做如下展望：

1. 推荐冷启动问题。基于信息熵的路径推荐算法中，根据乘客历史订单数据计算信息熵并定义不同特征的偏好权重，如果乘客没有订单，则完全无法计算乘客特征偏好和查找相似乘客，一般解决此类冷启动问题的可行方法是通过其他领域（例如酒店、旅游）购买历史、浏览数据，分析乘客特征偏好，设计跨领域的路径推荐算法。
2. 路径搜索权重的计算问题。在KSPG算法中，有向边的权重只能选择可以叠加计算的特征值，例如价格、时间等。如何在构建实时的换乘交通网络时，有效的将离散、连续特征偏好融入至有向边的权重中，将路径搜索与推荐的有效结合，路径搜索中加入乘客偏好，节省搜索时间，提高路径推荐效率。
3. 不同出发-到达城市的路径推荐问题。由于乘客每次出行目的地并不相同，在不同行程中乘客的偏好特征也不相同，而目的地可能会直接影响用户最终路径的选择。因此，如何衡量不同行程的相似度，根据目的地对乘客出行目的进行预测，在不同行程上如何计算路径相似度和推荐排序。
4. 缓存策略动态更新问题。路径缓存中，我们基于上个时间窗口的搜索频次，计算当前时间窗口缓存路径的更新缓存频率，虽然可以保证行程的平均新鲜度最优，但当搜索行程数量较多时，计算成本较高，同时时间增长，会出现初始更新频率与实际偏差较大的情况，因此需要根据实时搜索数据制定动态的行程缓存路径更新策略，进一步提升缓存效果，提高搜索效率并最大化乘客访问信息的新鲜度。

在未来的工作中，这些问题都有待进一步的探讨和研究。

参考文献

[]Ctrip. [cited 2016 Dec]; Available from: <https://www.ctrip.com/>.

[2]Qunar. [cited 2016 Dec]; Available from: <https://www.qunar.com/>.

[3]Triplehop’s TripMatcher.[cited 2016 Dec]; Available from: <https://www.ski-europe.com>.

[4]Resnick, P. and H.R. Varian, Recommender systems. Communications of the ACM, 1997. 40(3): p. 56-58.

[5]Ricci, F., L. Rokach, and B. Shapira, Introduction to recommender systems handbook. 2011: Springer.

[6]Linden, G., B. Smith, and J. York, Amazon. Com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 2003. 7(1): p. 76-80.

[7]Miller, Bradley N., et al. “MovieLens unplugged: experiences with an occasionally connected recommender system.” Proceedings of the 8th international conference on Intelligent user interfaces. ACM, 2003.

[8]Hoffman, Karla L., Manfred Padberg, and Giovanni Rinaldi. “Traveling salesman problem.” Encyclopedia of Operations Research and Management Science. Springer US, 2013. 1573-1578.

[9]Bielli, Maurizio, Azedine Boulmakoul, and Hicham Mouncif. “Object modeling and path computation for multimodal travel systems.” European Journal of Operational Research 175.3 (2006): 1705-1730.

[10]Miller-Hooks, Elise D., and Hani S. Mahmassani. “Least expected time paths in stochastic, time-varying transportation networks.” Transportation Science 34.2 (2000): 198-215.

[11]Nikolova, Evdokia, Matthew Brand, and David R. Karger. “Optimal Route Planning under Uncertainty.” ICAPS. Vol. 6. 2006.

[12]陈丽佳. “基于公交网络模型的最优路径算法研究与实现.” (2009).

[13]REDDY V R, KASILINGAM R G. Intermodal Transportation Considering Transfer Costs[C]// Proceedings of 1995 Global Trends Conference of the Academy of business Administration. Aruba: 1995.

[14]Delling, Daniel, Thomas Pajor, and Dorothea Wagner. “Accelerating multi-modal route planning by access-nodes.” European Symposium on Algorithms. Springer Berlin Heidelberg, 2009.

[15]刘舰, and 俞建宁. “多式联运运输方式选择的模型和算法.” 兰州交通大学学报(2010).

[16]J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In SIGSPATIAL, pages 99–108. ACM, 2010

[17]Dai, Jian, et al. “Personalized route recommendation using big trajectory data.” 2015 SIEEE 31st International Conference on Data Engineering. IEEE, 2015.

[18]HU, Ji-hua, et al. “Hierarchical Path Planning Method Based on Taxi Driver Experiences [J].” Journal of Transportation Systems Engineering and Information Technology 1 (2013): 029.

[19]Yuan, Nicholas Jing, et al. “T-finder: A recommender system for finding passengers and vacant taxis.” IEEE Transactions on Knowledge and Data Engineering 25.10 (2013): 2390-2403.

[20]Huang, Yuxia, and Ling Bian. “A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet.” Expert Systems with Applications 36.1 (2009): 933-943.

[21]Huagang Yin and et al., “Photo2trip: an interactive tripplanning system based on geo-tagged photos”, in MM10. ACM, 2010, pp. 1579–1582.

[22]Chang-Shing Lee and et al., “Ontological recommendation multi-agent for tainan city travel”, Expert Systems with Applications, vol. 36, no. 3, pp. 6740–6753, 2009.

[23]Laura Sebastia and et al., “e-tourism: a tourist recommendation and planning application”, IJAIT, vol. 18, no. 05, pp.717–738, 2009.

[24]Chieh-Yuan Tsai and Shang-Hsuan Chung, “A personalized route recommendation service for theme parks using rfid information and tourist behavior”, Decision Support Systems, vol. 52, no. 2, pp. 514–527, 2012.

[25]Okuma, Takashi, et al. “A pilot user study on 3-d museum guide with route recommendation using a sustainable positioning system.” Control, Automation and Systems, 2007. ICCAS’07. International Conference on. IEEE, 2007.

[26]Deng, Yong, et al. “Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment.” Applied Soft Computing 12.3 (2012): 1231-1237.

[27]Madduri, Kamesh, et al. “An experimental study of a parallel shortest path algorithm for solving large-scale graph instances.” Proceedings of the Meeting on Algorithm Engineering & Expermiments. Society for Industrial and Applied Mathematics, 2007.

[28]Androutsopoulos, Konstantinos N., and Konstantinos G. Zografos. “Solving the k-shortest path problem with time windows in a time varying network.” Operations Research Letters 36.6 (2008): 692-695.

[29]Wang, Zengping, Gang Li, and Jianwen Ren. “A new search algorithm for transmission section based on K shortest paths.” Diangong Jishu Xuebao(Transactions of China Electrotechnical Society) 27.4 (2012): 193-201.

[30]Cheng, Chunhsiang, et al. “A loop-free extended Bellman-Ford routing protocol without bouncing effect.” ACM SIGCOMM Computer Communication Review. Vol. 19. No. 4. ACM, 1989.

[31]Fanding, Duan. “A Faster Algorithm for Shortest-Ptath── SPFA [J].” Journal of Southwest Jiaotong University 2 (1994).

[32]Yen, Jin Y. “Finding the k shortest loopless paths in a network.” Management Science 17.11 (1971): 712-716.

[33]Martins, Ernesto QV, and Marta MB Pascoal. “A new implementation of Yen’s ranking loopless paths algorithm.” Quarterly Journal of the Belgian, French and Italian Operations Research Societies 1.2 (2003): 121-133.

[34]Hershberger, John, Matthew Maxel, and Subhash Suri. “Finding the k shortest simple paths: A new algorithm and its implementation.” ACM Transactions on Algorithms (TALG) 3.4 (2007): 45.

[35]王璇. 城际多方式出行路径规划方法研究. MS thesis. 北京交通大学, 2010.

[36]Kriegel, Hans-Peter, Matthias Renz, and Matthias Schubert. “Route skyline queries: A multi-preference path planning approach.” 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). IEEE, 2010.

[37]Resnick, P. and H.R. Varian, Recommender systems. Communications of the ACM, 1997. 40(3): p. 56-58.

[38]Kim, Jinyoung, Hyungjin Kim, and Jung-hee Ryu. “TripTip: a trip planning service with tag-based recommendation.” CHI’09 Extended Abstracts on Human Factors in Computing Systems. ACM, 2009.

[39]Huang, Yuxia, and Ling Bian. “A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet.” Expert Systems with Applications 36.1 (2009): 933-943.

[40]yle, Lorcan. Making Personalised Flight Recommendations using Implicit Feedback. Diss. Trinity College, 2004.

[41]Zhou, Tao, et al. “Bipartite network projection and personal recommendation.” Physical Review E 76.4 (2007): 046115.

[42]Huang, Zan, Hsinchun Chen, and Daniel Zeng. “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering.” ACM Transactions on Information Systems (TOIS) 22.1 (2004): 116-142.

[43]Claypool, Mark, et al. “Combining content-based and collaborative filters in an online newspaper.” Proceedings of ACM SIGIR workshop on recommender systems. Vol. 60. 1999.

[44]Pazzani, Michael J. “A framework for collaborative, content-based and demographic filtering.” Artificial Intelligence Review 13.5-6 (1999): 393-408.

[45]Karatzoglou, A., L. Baltrunas, and Y. Shi. Learning to rank for recommender systems. In Proceedings of the 7th ACM Conference on Recommender Systems. 2013. ACM.

[46]Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami. “Mining association rules between sets of items in large databases.” Acm sigmod record. Vol. 22. No. 2. ACM, 1993.

[47]Herlocker, Jonathan L., et al. “An algorithmic framework for performing collaborative filtering.” Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

[48]Herlocker, Jonathan L., et al. “An algorithmic framework for performing collaborative filtering.” Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

[49]Shardanand, Upendra, and Pattie Maes. “Social information filtering: algorithms for automating “word of mouth”.” Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., 1995.

[50]Zhou, Tao, et al. “Bipartite network projection and personal recommendation.” Physical Review E 76.4 (2007): 046115.

[51]Lü, Linyuan, and Weiping Liu. “Information filtering via preferential diffusion.” Physical Review E 83.6 (2011): 066119.

[52]Markatos, Evangelos P. “On caching search engine query results.” Computer Communications 24.2 (2001): 137-143.

[53]Yang, Ge. “A proxy caching algorithm based on popularity for streaming media.” 2013 Ninth International Conference on Natural Computation (ICNC). IEEE, 2013.

[54]Hassin, Refael. “Approximation schemes for the restricted shortest path problem.” Mathematics of Operations research 17.1 (1992): 36-42.

[55]Fisher, Marshall L. “The Lagrangian relaxation method for solving integer programming problems.” Management science 27.1 (1981): 1-18.

[56] Silverstein, Craig, et al. “Analysis of a very large web search engine query log.” ACM SIGIR Forum. Vol. 33. No. 1. ACM, 1999.

致谢

值此论文完成之际，谨向在两年半研究生学习阶段所有教导、关心、帮助过我的人表示最诚挚的感谢。

首先，感谢我的导师曹健教授。从研一到现在，无论科研还是生活，曹老师给予我很多指导和鼓励，在毕业论文的整个研究过程中，曹老师一直悉心教导、指点，当研究和实验陷入低谷时，总是给予帮助，每周思辨性的讨论为我提供开拓思维，学会如何从多方面考虑问题和解决问题，最后成功的化解瓶颈，让我受益匪浅。曹老师对科研的认真谨慎负责任的态度令我钦佩，生活中平易近人且不失风度，这些也是我今后在工作、生活和学习中应有的态度。

同时，也非常感谢已经毕业的陈誉文和杨芳洲学长。项目研究中，在他们的指导下逐渐理解科研方法，每周的交流使得科研目标更加明确，同时也要感谢赵亚峰同学，在同一项目组中共同成长、互帮互助，在科研实验中也给了我很大帮助，使得论文中实验才可以进展如此顺利。还要感谢姚艳博士、冯珊珊博士、张棪等同学，让我再SJTU-CIT的大家庭中度过最美好的研究生时光，是我莫大的荣幸。

最后，由衷的感谢一直陪伴的父母、家人和朋友，谢谢你们的支持和关心，在困难的时候给予我坚强的后盾，这也是我以后在工作中前进的最大动力。

附录一 符号与标记

|  |  |
| --- | --- |
|  | 极小值 |
|  | 极大值 |
|  | 平均值 |
|  | 连加运算符 |
|  | 积分运算符 |
|  | 集合A中元素个数 |
| # | 数量 |
|  | 以k为底，m的对数 |

附录二 英文缩略语表

|  |  |  |
| --- | --- | --- |
| OTA | Online Travel Agent | 在线旅游代理 |
| CF | Collaborative Filtering | 协同过滤 |
| KMCSP | K Multiple Constraints Shortest Paths | top-带约束最短路径问题 |
| KSPG | K Shortest Path Generator | 带约束的启发式路径搜索算法 |
| IERA | Information Entropy Recommendation Algorithm | 基于信息熵的路径推荐算法 |
| KNN | K-Nearest Neighbors | K最近邻模型 |
| FIFO | First In First Out | 最先进入最先换出 |
| LRU | Least Recently Used | 最近最少使用页面置换 |
| LFU | Least Frequently Used | 最近最不常用页面置换 |
| FBR | Frequency-Based Replacement | 基于访问频率页面置换 |
| MAE | Mean Absolute Error | 平均绝对误差 |
| MSE | Mean Square Error | 均方误差 |
| MAP | Mean Average Precision | 平均准确率均值 |
| BJS | Beijing | 北京城市代码 |
| SHA | Shanghai | 上海城市代码 |
| CAN | Guangzhou | 广州城市代码 |
| XMN | Xiamen | 厦门城市代码 |
| TSN | Tianjin | 天津城市代码 |
| SYX | SanYa | 三亚城市代码 |
| CKG | ChongQing | 重庆城市代码 |

攻读硕士学位期间已发表的论文

**已录用的论文：**

1. 华逸群 and 曹健. "基于模糊时间序列的机票价格预测." 小型微型计算机系统 (2015). （第一作者）

1. []Ctrip. [cited 2016 Dec]; Available from: <https://www.ctrip.com/>. [↑](#endnote-ref-2)
2. []Qunar. [cited 2016 Dec]; Available from: <https://www.qunar.com/>. [↑](#endnote-ref-3)
3. []Triplehop’s TripMatcher.[cited 2016 Dec]; Available from: <https://www.ski-europe.com>. [↑](#endnote-ref-4)
4. []Resnick, P. and H.R. Varian, Recommender systems. Communications of the ACM, 1997. 40(3): p. 56-58. [↑](#endnote-ref-5)
5. []Ricci, F., L. Rokach, and B. Shapira, Introduction to recommender systems handbook. 2011: Springer. [↑](#endnote-ref-6)
6. []Linden, G., B. Smith, and J. York, Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 2003. 7(1): p. 76-80. [↑](#endnote-ref-7)
7. []Miller, Bradley N., et al. "MovieLens unplugged: experiences with an occasionally connected recommender system." Proceedings of the 8th international conference on Intelligent user interfaces. ACM, 2003. [↑](#endnote-ref-8)
8. []Hoffman, Karla L., Manfred Padberg, and Giovanni Rinaldi. "Traveling salesman problem." Encyclopedia of Operations Research and Management Science. Springer US, 2013. 1573-1578. [↑](#endnote-ref-9)
9. []Bielli, Maurizio, Azedine Boulmakoul, and Hicham Mouncif. "Object modeling and path computation for multimodal travel systems." European Journal of Operational Research 175.3 (2006): 1705-1730. [↑](#endnote-ref-10)
10. []Miller-Hooks, Elise D., and Hani S. Mahmassani. "Least expected time paths in stochastic, time-varying transportation networks." Transportation Science 34.2 (2000): 198-215. [↑](#endnote-ref-11)
11. []Nikolova, Evdokia, Matthew Brand, and David R. Karger. "Optimal Route Planning under Uncertainty." ICAPS. Vol. 6. 2006. [↑](#endnote-ref-12)
12. []陈丽佳. "基于公交网络模型的最优路径算法研究与实现." (2009). [↑](#endnote-ref-13)
13. []REDDY V R, KASILINGAM R G. Intermodal Transportation Considering Transfer Costs[C]// Proceedings of 1995 Global Trends Conference of the Academy of business Administration. Aruba: 1995. [↑](#endnote-ref-14)
14. []Delling, Daniel, Thomas Pajor, and Dorothea Wagner. "Accelerating multi-modal route planning by access-nodes." European Symposium on Algorithms. Springer Berlin Heidelberg, 2009. [↑](#endnote-ref-15)
15. []刘舰, and 俞建宁. "多式联运运输方式选择的模型和算法." 兰州交通大学学报(2010). [↑](#endnote-ref-16)
16. []J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In SIGSPATIAL, pages 99–108. ACM, 2010 [↑](#endnote-ref-17)
17. []Dai, Jian, et al. "Personalized route recommendation using big trajectory data." 2015 SIEEE 31st International Conference on Data Engineering. IEEE, 2015. [↑](#endnote-ref-18)
18. []HU, Ji-hua, et al. "Hierarchical Path Planning Method Based on Taxi Driver Experiences [J]." Journal of Transportation Systems Engineering and Information Technology 1 (2013): 029. [↑](#endnote-ref-19)
19. []Yuan, Nicholas Jing, et al. "T-finder: A recommender system for finding passengers and vacant taxis." IEEE Transactions on Knowledge and Data Engineering 25.10 (2013): 2390-2403. [↑](#endnote-ref-20)
20. [] Huang, Yuxia, and Ling Bian. "A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet." Expert Systems with Applications 36.1 (2009): 933-943. [↑](#endnote-ref-21)
21. []Huagang Yin and et al., “Photo2trip: an interactive tripplanning system based on geo-tagged photos”, in MM10. ACM, 2010, pp. 1579–1582. [↑](#endnote-ref-22)
22. []Chang-Shing Lee and et al., “Ontological recommendation multi-agent for tainan city travel”, Expert Systems with Applications, vol. 36, no. 3, pp. 6740–6753, 2009. [↑](#endnote-ref-23)
23. []Laura Sebastia and et al., “e-tourism: a tourist recommendation and planning application”, IJAIT, vol. 18, no. 05, pp.717–738, 2009. [↑](#endnote-ref-24)
24. []Chieh-Yuan Tsai and Shang-Hsuan Chung, “A personalized route recommendation service for theme parks using rfid information and tourist behavior”, Decision Support Systems, vol. 52, no. 2, pp. 514–527, 2012. [↑](#endnote-ref-25)
25. []Okuma, Takashi, et al. "A pilot user study on 3-d museum guide with route recommendation using a sustainable positioning system." Control, Automation and Systems, 2007. ICCAS'07. International Conference on. IEEE, 2007. [↑](#endnote-ref-26)
26. []Deng, Yong, et al. "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment." Applied Soft Computing 12.3 (2012): 1231-1237. [↑](#endnote-ref-27)
27. []Madduri, Kamesh, et al. "An experimental study of a parallel shortest path algorithm for solving large-scale graph instances." Proceedings of the Meeting on Algorithm Engineering & Expermiments. Society for Industrial and Applied Mathematics, 2007. [↑](#endnote-ref-28)
28. []Androutsopoulos, Konstantinos N., and Konstantinos G. Zografos. "Solving the k-shortest path problem with time windows in a time varying network." Operations Research Letters 36.6 (2008): 692-695. [↑](#endnote-ref-29)
29. []Wang, Zengping, Gang Li, and Jianwen Ren. "A new search algorithm for transmission section based on K shortest paths." Diangong Jishu Xuebao(Transactions of China Electrotechnical Society) 27.4 (2012): 193-201. [↑](#endnote-ref-30)
30. []Cheng, Chunhsiang, et al. "A loop-free extended Bellman-Ford routing protocol without bouncing effect." ACM SIGCOMM Computer Communication Review. Vol. 19. No. 4. ACM, 1989. [↑](#endnote-ref-31)
31. []Fanding, Duan. "A Faster Algorithm for Shortest-Ptath── SPFA [J]." Journal of Southwest Jiaotong University 2 (1994). [↑](#endnote-ref-32)
32. []Yen, Jin Y. "Finding the k shortest loopless paths in a network." management Science 17.11 (1971): 712-716. [↑](#endnote-ref-33)
33. []Martins, Ernesto QV, and Marta MB Pascoal. "A new implementation of Yen’s ranking loopless paths algorithm." Quarterly Journal of the Belgian, French and Italian Operations Research Societies 1.2 (2003): 121-133. [↑](#endnote-ref-34)
34. []Hershberger, John, Matthew Maxel, and Subhash Suri. "Finding the k shortest simple paths: A new algorithm and its implementation." ACM Transactions on Algorithms (TALG) 3.4 (2007): 45. [↑](#endnote-ref-35)
35. []王璇. 城际多方式出行路径规划方法研究. MS thesis. 北京交通大学, 2010. [↑](#endnote-ref-36)
36. []Kriegel, Hans-Peter, Matthias Renz, and Matthias Schubert. "Route skyline queries: A multi-preference path planning approach." 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). IEEE, 2010. [↑](#endnote-ref-37)
37. []Resnick, P. and H.R. Varian, Recommender systems. Communications of the ACM, 1997. 40(3): p. 56-58. [↑](#endnote-ref-38)
38. []Kim, Jinyoung, Hyungjin Kim, and Jung-hee Ryu. "TripTip: a trip planning service with tag-based recommendation." CHI'09 Extended Abstracts on Human Factors in Computing Systems. ACM, 2009. [↑](#endnote-ref-39)
39. []Huang, Yuxia, and Ling Bian. "A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet." Expert Systems with Applications 36.1 (2009): 933-943. [↑](#endnote-ref-40)
40. []yle, Lorcan. Making Personalised Flight Recommendations using Implicit Feedback. Diss. Trinity College, 2004. [↑](#endnote-ref-41)
41. []Zhou, Tao, et al. "Bipartite network projection and personal recommendation." Physical Review E 76.4 (2007): 046115. [↑](#endnote-ref-42)
42. []Huang, Zan, Hsinchun Chen, and Daniel Zeng. "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering." ACM Transactions on Information Systems (TOIS) 22.1 (2004): 116-142. [↑](#endnote-ref-43)
43. []Claypool, Mark, et al. "Combining content-based and collaborative filters in an online newspaper." Proceedings of ACM SIGIR workshop on recommender systems. Vol. 60. 1999. [↑](#endnote-ref-44)
44. []Pazzani, Michael J. "A framework for collaborative, content-based and demographic filtering." Artificial Intelligence Review 13.5-6 (1999): 393-408. [↑](#endnote-ref-45)
45. []Karatzoglou, A., L. Baltrunas, and Y. Shi. Learning to rank for recommender systems. in Proceedings of the 7th ACM Conference on Recommender Systems. 2013. ACM. [↑](#endnote-ref-46)
46. []Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases." Acm sigmod record. Vol. 22. No. 2. ACM, 1993. [↑](#endnote-ref-47)
47. []Herlocker, Jonathan L., et al. "An algorithmic framework for performing collaborative filtering." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999. [↑](#endnote-ref-48)
48. []Herlocker, Jonathan L., et al. "An algorithmic framework for performing collaborative filtering." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999. [↑](#endnote-ref-49)
49. []Shardanand, Upendra, and Pattie Maes. "Social information filtering: algorithms for automating “word of mouth”." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., 1995. [↑](#endnote-ref-50)
50. []Zhou, Tao, et al. "Bipartite network projection and personal recommendation." Physical Review E 76.4 (2007): 046115. [↑](#endnote-ref-51)
51. []Lü, Linyuan, and Weiping Liu. "Information filtering via preferential diffusion." Physical Review E 83.6 (2011): 066119. [↑](#endnote-ref-52)
52. []Markatos, Evangelos P. "On caching search engine query results." Computer Communications 24.2 (2001): 137-143. [↑](#endnote-ref-53)
53. []Yang, Ge. "A proxy caching algorithm based on popularity for streaming media." 2013 Ninth International Conference on Natural Computation (ICNC). IEEE, 2013. [↑](#endnote-ref-54)
54. []Hassin, Refael. "Approximation schemes for the restricted shortest path problem." Mathematics of Operations research 17.1 (1992): 36-42. [↑](#endnote-ref-55)
55. []Fisher, Marshall L. "The Lagrangian relaxation method for solving integer programming problems." Management science 27.1 (1981): 1-18. [↑](#endnote-ref-56)
56. www.ctrip.com [↑](#footnote-ref-1)
57. [] Silverstein, Craig, et al. "Analysis of a very large web search engine query log." ACM SIGIR Forum. Vol. 33. No. 1. ACM, 1999. [↑](#endnote-ref-57)