

**硕士学位论文**

基于机器学习的酒店价格预测分析与系统实现

**专业名称**： **计算机技术**

**研究生姓名： 邓 磊**

**导师姓名**：  **吴巍炜**

The analysis and implementation of hotel price prediction based on machine learning

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

Deng Lei

Supervised by

Prof. Weiwei Wu

April 2017

**东南大学学位论文独创性声明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名： 日期：

**东南大学学位论文使用授权声明**

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括以电子信息形式刊登）论文的全部内容或中、英文摘要等部分内容。论文的公布（包括以电子信息形式刊登）授权东南大学研究生院办理。

研究生签名： 导师签名： 日期：

摘要

随着我国信息技术的持续高速发展，宽带网络和高速移动互联网得到快速普及应用，人们越来越倾向于在线选择旅游产品。我国的旅游产业因此发展迅猛，据国家旅游局数据显示，2015年中国国内旅游、出境旅游人次和国内旅游消费、境外旅游消费均列世界第一。在这种形势下，由于当前大部分国内消费者都属于对价格一场敏感的群体，因此为用户提供某个酒店在当前时间预定和未来入住时间之间的一个价格预测势必会提升用户体验。同时，预测出来的价格也可以为预定平台提供价格调整指导信息。

酒店价格是一个多方博弈后的结果，需要综合多方的因素进行考量。现有的类似研究工作大都只是针对影响酒店价格的特征进行分析，而不是给出对酒店价格的预测。本文针对酒店的历史价格信息和酒店的固有特征对未来某天的价格进行预测，从价格历史变动规律这一角度来辅助OTA(Online travel agency)对酒店定价，主要进行了以下四 个方面的研究：

1. 通过携程的英文酒店预订平台，编写爬虫获取最为热门的18个城市的酒店历史价格数据。包括酒店的当前预定价格至未来第28天的价格数据，酒店的基本信息，如开业时间，位置信息，酒店设施信息等。通过获取到的大量数据，分析得到酒店价格变动的基本规律，以及不同类型的酒店的价格区别等信息。
2. 使用传统的机器学习算法对酒店价格进行分别预测，获得影响影响酒店价格预测结果的各项因素，并确定每个影响因子的作用大小。
3. 通过集成学习算法对酒店价格进行预测。基于改进的Stacking集成学习算法，提出一个分层的(Hierachrical Ensemble Learning Model, HELM ) 集成学习模型来对酒店价格进行预测。
4. 设计并实现了一个以HELM为核心的的酒店价格预测系统。

**关键词：**酒店价格预测，二分K-均值，集成学习，Restful

Abstract

目录

[摘要 I](#_Toc444193534)

[Abstract II](#_Toc444193535)

[目录 III](#_Toc444193536)

[图目录 V](#_Toc444193537)

[表目录 VI](#_Toc444193538)

[缩略词表 VII](#_Toc444193539)

[第一章 绪论 1](#_Toc444193540)

[1.1 研究背景 1](#_Toc444193541)

[1.2 研究现状 2](#_Toc444193542)

[1.3 研究内容 3](#_Toc444193543)

[1.4 论文组织结构 3](#_Toc444193544)

[第二章 背景知识 5](#_Toc444193545)

[2.1 机器学习 5](#_Toc444193546)

[2.2 二分k均值聚类 5](#_Toc444193547)

[2.2.1 5](#_Toc444193548)

[2.2.2 6](#_Toc444193549)

[2.3 随机森林、梯度提升决策树、k近邻、XGBOOST、 7](#_Toc444193550)

[2.3.1 8](#_Toc444193551)

[2.3.2 8](#_Toc444193552)

[2.4 集成学习 9](#_Toc444193553)

[2.5 本章小结 10](#_Toc444193554)

[第三章 数据采集与预处理分析 11](#_Toc444193555)

[3.1 数据的获取 11](#_Toc444193556)

[3.1.1 11](#_Toc444193557)

[3.1.2 12](#_Toc444193558)

[3.2 数据的预处理 13](#_Toc444193559)

[3.2.1 13](#_Toc444193560)

[3.2.2 14](#_Toc444193561)

[3.2.3 15](#_Toc444193562)

3.3 价格特征分析-------------------------------------------------------------------------------------------------16

[3.3 本章小结 15](#_Toc444193563)

[第四章 基于传统机器学习算法的酒店价格预测 17](#_Toc444193564)

4.1 酒店价格预测问题定义………………………………………………………………………..18

[4.1 随机森林 17](#_Toc444193565)

[4.2 k近邻 17](#_Toc444193566)

[4均方根误差RMSE（Root Mean Square Error） 19](#_Toc444193568)

[4.3 梯度提升决策树 19](#_Toc444193569)

[4.3.1 XGBOOST 20](#_Toc444193570)

[4.4 本章小结 24](#_Toc444193573)

[第五章 基于集成学习的酒店价格预测 25](#_Toc444193574)

[5.1 二层Stacking预测 25](#_Toc444193575)

[5.2 基于二分k-均值聚类的二层Stacking预测 25](#_Toc444193576)

[5.3 实验结果及其分析 30](#_Toc444193580)

[5.4 本章小结 35](#_Toc444193583)

[第六章 预测系统实现 36](#_Toc444193584)

[6.1 系统功能 36](#_Toc444193585)

[6.2 原型实现 36](#_Toc444193586)

6.3 本章小结………………………………………………………………………………………….37

第七章 总结与展望………………………………………………………………………………………..39

7.1 工作总结………………………………………………………………………………………….41

7.2 研究展望………………………………………………………………….………………………42

[致谢 38](#_Toc444193587)

[参考文献 39](#_Toc444193588)

[作者简介 42](#_Toc444193589)

# 图目录

# 表目录

# 缩略词表

|  |  |  |
| --- | --- | --- |
| 缩略词 | 英文全称 | 中文全称 |
| GBDT | Gradient Boosting Decision Tree | 梯度提升决策树 |
| RMSE | Root Mean Square Error | 均方根误差 |

1. 绪论

本章首先介绍酒店价格预测的研究背景，然后对当前的研究现状进行了分类陈述，其次介绍了本文的主要工作，最后介绍本文的组织结构。

## 研究背景

随着我国信息技术的持续高速发展，宽带网络和高速移动互联网得到快速普及应用，不仅给人们的生活带来了极大的便利，也快速的改变着我们的消费习惯。根据中国互联网络信息中心(CNNIC)在2016年1月发布的第37次中国互联网络发展状况统计报告显示，截至2015年12月，我国网民规模达到6.88亿，互联网普及率为50.3%，较2014年年底提升了2.4个百分点；与此同时，中国手机网民规模达到6.20亿，网民中使用手机上网人群占比由2014年底的85.85%提升至90.1%。截至2015年12月，在网上预订过旅行产品的网民规模达到2.60亿，较2014年底增长3782万人，增长率为17.1%，其中在网上预订酒店的网民占14.7%。同时，通过手机预订酒店产品的网民规模达到了2.10亿，较2014年12月增长了7596万人，增长率为56.4%。从上面的统计数据可以看出，我国网民使用旅行预订的网民体量巨大，且增速明显，这一点在手机网民中显得尤为突出。

此外，报告也指出，航空公司的直营业务发展势头逐年向好，将逐步OTA的利润市场。然而在酒店行业，除了一些大型连锁酒店和部分高端酒店，绝大多数酒店都没有实力去运营一个自己的预订网站。此外，我国的旅游产业发展迅猛，据国家旅游局数据显示，2015年中国国内旅游、出境旅游人次和国内旅游消费、境外旅游消费均列世界第一。在这种形势下，为用户提供某个酒店在当前时间和入住时间之间的一个价格预测，由于大部分国内消费者都属于对价格异常敏感的群体[1]，这势必会提升用户的使用体验，因而能够提高携程这种有大型直采业务的OTA的订单成交量。

当前各大OTA都还没有提供关于某一酒店价格的预测，只提供（如携程网）了各大网站的价格对比。而若像kayak那样为机票价格提供未来的预测，并提供该预测的准确度，依据这些丰富的预测信息，用户会更愿意停留在当前预订平台，最终达成订单的概率也会增大。另外，酒店也可以依据用户输入的信息，如酒店所在城市、时间、价格范围、酒店所在范围以及星级等相关信息，并结合预测出的价格，动态调整酒店的价格。通过本文的工作，消费者可以更加容易预订到令自己满意的酒店，在线预订平台也能获得各竞争酒店价格变化的信息，以酒店也能提高入住率，这能提高消费者-预订平台-酒店三方的满意度。

## 相关研究现状

目前有不少针对酒店价格问题的研究，但是大多都是从盈利的角度单单研究影响酒店价格的因素，或者是从酒店客房数量需求的预测，很少有通过历史价格数据以及酒店自身资源拥有情况来对酒店未来某一时间的价格预测的研究。下面将从影响酒店价格的因素和对酒店客房数量需求预测这两个方面来进行介绍。

### 影响酒店价格的因素

对产品价格以及影响产品价格的因子进行分析，这早已在金融和房地产等行业得到了广泛应用[2]。在该文中作者使用组成产品的不同特征来论证了价格的性质，在他提出的特征价格模型（hedonic pricing model）中把产品的**各项特征**当作**独立变量**，而价格当作一个**依赖变量。**通过找到有相同特征的酒店，就可能比较相似酒店的客房价格[3]。用特征价格理论得到的结果显示影响酒店客房价格的因素多种多样，以至于在不同的实验结果中有可能得到相反的结论[4]。出现这样的问题主要是因为：采用的实证方法(empirical methods，线性和非线性回归估计，有参和无参算法)，用到的数据质量及完备性，应用区域，以及模型中用到的特征不一致。因此，在前期的数据准备阶段的工作尤为重要。

文献[5]在**特征价格模型**中使用了五个不同的独立变量（酒店的星级，建筑年龄，餐厅有无，距离市中心的距离，是否临河），然而建筑年龄和是否临河最终因为在建立的模型中影响不明显而被排除掉。该模型显示离市中心的**距离**是影响客房价格的最大因素，其次是有无餐厅和酒店的星级。文献[6]等使用**特征价格模型和回归分析**研究了影响酒店价格和影响用户下单的特征，结果显示**位置信息**是影响下单成功的决定因素。其他的因素包括房间的大小，有无浴缸，有无独立卫生间。而游泳池，空调，壁橱，礼品券等信息对房价的影响不大。文献[7]使用**线性回归模型**使用以色列的215家酒店共计30000个房间来研究客房数量，星级情况，是否是连锁酒店对价格的影响。结果显示，酒店的**星级情况**是酒店价格不同的决定因素。然而，一个酒店是否是连锁酒店在以色列的不同地区对客房价格的影响却完全相反。文献[8]同样使用**特征价格模型**来研究了Norway的74家酒店数据，结果却显示**小冰箱和吹风机**是决定酒店房价的关键因素。对于单人间，同等情况下连锁酒店的价格更高，且提供客房服务的房间价格更低；对于双人间，是不是连锁酒店不影响酒店的价格，与市中心的距离才是影响双人间价格的重要因素。而游泳池和餐厅都对酒店价格不构成影响。文献[9]等人使用了**分位数回归分析**，结果显示酒店的**建筑时间**与酒店的价格成正相关关系。同样也显示连锁情况与距离市中心的距离对客房的价格没有影响。文献[10]同样使用**特征价格模型**表明酒店价格与是否**临近机场或者中央商务区**有关。并且，连锁酒店及提供免费停车的酒店的价格相对更高。但是，提供免费早餐对酒店价格没有影响。文献[13]提出了从**用户的评论中**提取产品特征和评价观点的方法，该研究提出的OPINE在特征提取工作上实现了降低3%的召回率的情况下，提高22%的准确度。

综上可以看出，对于研究的酒店所处的国家或地区，以及使用的不同方法，得出影响价格的因素有很大不同。其中，有的研究工作使用的数据集过小，不够全面，有的研究工作使用的特征过少，只能反映局部的特性。最重要的是，大部分的研究工作都只是针对影响酒店价格的特征进行分析，而不是着重给出对酒店未来价格的预测。

### 酒店客房需求预测

已有大量针对旅游需求预测的研究，这些研究采用了不同的方法和不同的数据。文献[14][15][16]采用的因变量（即需要预测的变量）包括到达的游客数量，游客消费，游客在目的地停留的天数。其中到达的游客数量是除游客消费之外最常采用的因变量，而停留天数通常被当作旅游需求的替代品。这些文献采用的数据来自于旅游者自身，酒店的登记记录，抽样调查等。使用这些数据忽视了那些住宿在亲戚朋友家的情况。此外，预测模型中的自变量包括酒店所在地的人口数量，人均收入，酒店及其竞争者的房价，汇率，消费者品味，营销费用以及其他可量化变量，如大型营销活动或者大型体育赛事。旅游需求预测中的定性研究方法采用了多种方法，包括线性、非线性模型，时间序列化技术，计量经济学模型和人工智能方法。

随着人们越来越习惯在线选择旅游产品，产生了大量的在线行为数据，通过这些数据可以作为作为游客需求的指示。谷歌趋势是有谷歌公司提供的一款公开工具，通过该工具用户可以获得规格化的在谷歌搜索上产生的查询数据。这些查询数据不仅是实时且高频的（取代了原来的以周、季度或者年度为单位的数据），同时也对用户的细小行为变化敏感[17]。但是根据文献[18]，目前较少有研究在旅游业中使用在线数据来做预测。

## 研究内容

当前网络上有众多的酒店预订平台，其中携程作为国内最大的旅游集团能提供最全的酒店信息。这些数据是用来做本文预测的元数据。这些元数据中隐含着哪些地区的酒店最为热门，哪些区域的酒店价格均价最高，哪些区域的酒店评价最高，哪些酒店的交通最为方便等信息。本文就是利用携程网上爬取到的酒店实时价格信息，存入到本地文件中，在挖掘出来的特征基础上建立起价格预测模型。主要研究内容包括：

1. 分析携程网酒店价格数据的主要特征计。首先通过编写爬虫获取到携程网上大量的酒店实时价格数据，及酒店的基本信息，如开业时间，位置信息，酒店设施信息等。通过获取到的大量数据，分析得到不同评分酒店的平均价格，酒店的星级和评分分布状况，酒店的星级与评论数量对比，当前预约价格与入住当天预约在不同星级下的价格对比，预约时间点与预约价格对比。通过这些分析，表明获取到的数据符合人们的日常认知，且具有典型性，可以作为进一步的研究对象。
2. 使用传统的机器学习算法对酒店价格进行分别预测，包括随机森林，k近邻，梯度提升决策树，XGBOOST(eXtreme GradientBoosting)。获得影响影响酒店价格预测结果的各项因素，并确定每个影响因子的作用大小。设计相关实验验证本文提出的算法和测度的可行性和有效性，并对其效果进行客观的对比和分析。
3. 通过集成学习算法对酒店价格进行预测。基于传统的机器学习算法，提出一个分层的(Hierachrical Ensemble Learning Model, HELM ) 集成学习模型来对酒店价格进行预测，该模型使用二分k均值算法作为预处理，并使用第二部分的传统机器学习算法结果作为新的训练集，以此来得到最终的价格预测结果。
4. 实现了一个基于Flask的Restful架构以展示最后的预测结果。通过显示酒店的分布，当前预定价格与入住当天预定价格的对比。

## 本文结构安排

本文共分为七章，组织安排如下：

第一章：绪论。介绍论文的研究背景、介绍论文的研究背景、研究现状、研究内容及论文组织结构。

第二章：背景知识。首先简要的介绍了机器学习机器发展历史；其次介绍了本文采用的聚类方法二分k均值算法；随后介绍了随机森林，梯度提升决策树，k近邻和XGBOOST算法；最后介绍集成学习。

第三章：数据采集与预处理。通过爬取携程网的酒店搜索，获取大量的价格数据，介绍了数据的结构和预处理工作，最后分析了酒店的价格与各项特征的基本联系。

第四章：基于传统机器学习算法的酒店价格预测。使用随机森林，k近邻，梯度提升决策树，XGBOOST(eXtreme GradientBoosting)对酒店价格进行预测，并得到影响预测效果的特征重要性。

第五章：基于集成学习的酒店价格预测。基于二分k均值聚类算法和前一章的传统机器学习算法对酒店价格进行预测。

第六章：系统实现。主要介绍介绍了整个实验结果的展示的原型系统。

第七章：总结全文的工作，并对后续的研究方向进行了展望。

1. 背景知识

本章介绍酒店价格预测的背景知识。我们首先简要介绍了机器学习的概念和发展历程，然后介绍二分K均值聚类，随后介绍本文用到的四个机器学习算法，最后介绍集成学习的概念。

## 机器学习

机器学习是一门专注于研究通过进行科学计算，利用得到的经验来提升系统自身性能的一门学科。在计算机领域，“经验”存在于“数据”之中。因此，机器学习中的主要研究内容就是从给定的数据中得到“模型”的算法，即“学习算法”。通过学习算法，给其输入经验数据那么就可以从这些数据中得出相应的模型，在面对新的数据的时候，模型能够提供相应的预测结果。换而言之，计算机科学专门研究“算法”，而机器学习是专门研究“学习算法”的分支[19]。上面提到的经验数据又称之为训练数据（训练集），“新的数据”又称为测试数据（测试集）。

根据训练集是否拥有标签，机器学习一般可以分成有监督学习（supervised learning）和无监督学习（unsupervised learning）。标签是指训练集中的每一个样本不仅包含了自变量，还有自变量在某种取值情况下对应的因变量的取值或者类型。数学表达式如下：

(2-1)

上式中的X和Y都是向量，函数映射F()表示了向量X与Y的映射关系。有监督学习中的每一个样本是X和Y成对给出，无监督学习中的样本只有向量X。在本文的预测研究中，每一个样本都是一条带标签的数据，X表示某一酒店在某一时刻的当天至未来第七天同一时刻的预定价格和酒店的固有信息，标签Y是距当前爬取时间8天后同一时间点的酒店真实价格。

监督学习以分类和回归作为代表，本文用到的随机森林、梯度提升决策树、k近邻和XGBOOST都是常见的监督学习算法，这些算法对原始数据集的依赖性较为敏感。在监督学习中，由于自变量和因变量都是成对给出的，所以当给予特定的自变量，我们可以知道输出的因变量的大致范围。这是和无监督学习的显著区别。

无监督学习以聚类为代表。与有监督学习算法不同，聚类方法仅仅是依赖样本的某些特定属性将相似的归为同类，模型并不了解划分出来的类别的具体意义。尽管如此，无监督学习在当前迅猛发展的深度学习中体现出越来越强大的功能，其中卷积神经网络就是杰出的代表。

## 二分k均值聚类

二分K-均值（Bisecting k-means）作为经典聚类算法k-means的一个变体，是典型的基于原型的聚类（Prototype-based clustering）。此类算法假设聚类结构能够通过一组原型来刻画[11]，通常情况下，算法先对原型进行初始化，然后对原型进行迭代更新求解。相对于k-均值算法，二分k-均值算法改进了随机选择初始质心的随机性造成聚类结果不确定性的问题。

在酒店预测问题中，各类型的酒店的位置、价格以及其他基本信息（比如星级，评分等）相差较大。具有相似属性的酒店的价格变动规律会有一定的联系，比如，地理位置相近的酒店调整价格的幅度和时间可能更接近，相同星级的酒店可能具有更驱动同的价格变化趋势。因此，本文首先考虑对酒店进行聚类，以对不同聚类中的酒店分别进行训练，再在对应的测试集上进行预测。基于该预测问题的实际情况，本文以酒店的经度、纬度、评分和星级四个特征对酒店进行聚类，采用这四个特征进行聚类。

误差平方和（Sum of the Squared Error, SSE）。用来在每一轮执行完成聚类分析后，对每个簇所属的非质心点计算其与最近质心点之间的距离，将所有这些距离值()相加求和，作为SSE来评估当前的聚类后的质量。聚类的目标即是使得SSE最小化。SSE的具体计算公式如下：

**算法1：二分K-均值算法**

**输入:** 所有酒店的经纬度坐标点及酒店数据集D，需要生成的簇集数K

**输出:** 聚类结果clusters

1. 初始化聚类中心

，C=[C0]

初始化聚类结果

1. 初始化lowestSSE =   
    (a) for i = 1,2…,|C| do:

i\* = i

clusters.add(

3.

二分K-均值算法使用的参数注解如下：

：聚类个数

：初始数据集

：属于聚类Ci的酒店集合

：聚类中心集合

：聚类误差平方和

：属于第i个聚类的酒店集合

：聚类过程中生成的2个聚类中心

：的误差平方和

：除外当前的所有聚类中心  
：对聚类之前其他聚类总误差平方和  
:第i家酒店  
:的经纬度  
:酒店所属的聚类中心  
:类簇集合

:除去数据集的其他所有簇集  
:传统k-均值聚类算法

二分k-均值算法的具体步骤如算法1所示。算法在第1步和第2步分别对聚类中心以及该聚类中心的聚类结果进行初始化。从第3步开始，当聚类簇集数小于K时尝试寻找其中一个合适的类簇进行二分聚类，直到得到K个聚类簇。步骤(a)对当前所有的簇集依次利用传统的k均值算法聚成2个类别，若当前聚类后的SSE小于原来的聚类情况，则更新聚类中心集合。第4步输出最后的聚类结果。

## 传统机器学习算法

### 随机森林

随机森林（Random Forest）是机器学习中的典型性方法之一，随机森林是在以决策树为基学习器的基础上构建的Bagging方法的变体。因此，此处首先介绍决策树和Bagging方法。

决策树（Decision Tree, DT）是机器学习中用于分类和回归的基本方法。决策树的决策过程呈树形。以分类问题为例，决策树表示利用各项特征对样本进行分类的过程，可以看作是一个条件函数，也可以看成是在特征空间与类空间上的条件概率分布[20]。决策树的生成思想是递归地自顶向下二分特征空间，是一种基于数据的归纳学习算法。常见的决策树算法有Quinlan于1986年提出的ID3[21]算法和1984年提出的C4.5[22]算法，以及Breiman等人于1984年提出的分类与回归树算法（Classification and regression tree,CART）[23]。

决策树模型中的所有信息都通过其树形结构表现出来，学习过程过程包含三个步骤：特征选择，树的生成与树的剪枝。决策树中有根结点、内部结点和叶结点。其中，每个内部结点表示一个特征或者一个属性。决策树模型示意图如图2.1。决策树的生成遵循奥卡姆剃刀原则，每一次都是选择最优的特征来对特征空间进行分割，这样使得每次结点分裂都是向着具有最好的分类的目标前进。如果这些训练数据的子集都能够被大致正确分类或者没有了可供分裂的特征，那么就新建一个叶结点，将该子集分到此叶节点中。如果子集中的数据还有没被正确分类或者还有待分割的特征，那么就递归地进行上述分裂。最后，当所有子集都被正确分类或者达到设定的某一阈值时（如树的深度），决策树的生成结束。如何确定最优的结点分裂是决策树学习的核心部分，同时也是ID3、C4.5和CART的最大区别。一下介绍三种分裂方法。

第一种，信息增益(Information divergence)

在信息增益计算过程中用到了信息论中的熵（entropy），熵是随机变量的不确定性的度量。假设随机变量X是一个离散且取值有限的随机变量。那么X的熵表示为：

2-1

从上式中可以看出，随机变量的熵与随机变量的取值无关，只与随机变量的分布有关，当熵越大，随机变量的不确定性也就越大。信息增益是内部结点按照某一特征进行分割时所减少的熵，即不确定性减少的程度。计算方式如下：

2-2

C(A)：特征A对应的所有取值集合

D：样本集合

Dc: 样本集合D中特征A取值为c的集合

H(D|A)：样本集合D在特征A下的经验条件熵

根节点

内部结点

叶结点

图 2-1 决策树模型示意图

第二种，信息增益比。

C4.5算法采用信息增益比来选择特征，与容易产生过拟合的ID3算法不同的是，C4.5不会偏向于取值较多的特征。信息增益比的计算公式为：

2-3

2-4

HA(D)：样本集合D关于特征A的所有取值的熵

第三种，基尼指数。

CART中的分类树使用基尼指数来分裂节点。样本集合D的基尼指数计算公式为：

2-5

Ck：表示样本集合D中属于第K类的样本

K：样本集合D中的类别数

那么在选取特征的时候，若特征A将样本D分割成两部分D1和D2，则样本集合D的基尼指数有如下定义：

2-6

基尼指数同熵的定义相似，表示集合D的不确定性，基尼指数Gini(D,A)表示样本集合D经过特征A分割后的不确定性。基尼指数的值越大，那么样本的不确定性也就越大，所以在分割节点的时候选取使样本确定性增大的能得到最小基尼系数的那个特征，并且能够同时确定最优切分点，即能够确定通过特征A的某个取值来分割。

Bagging是作为集成学习中并行式计算最著名的代表，不经修改就能用于二分类、多分类和回归任务中。其流程分为三个步骤：首先，通过重复采样，生成T个采样集，每个采样集中有m个样本，这m个样本都是有放回训练集的随机采样；其次，通过对每个采样样本训练出一个学习器；最后，将T个学习器的预测结果结合作为最后的输出，对于分类任务，使用简单投票法作为最后结果，对于回归任务，则是采用简单平均法作为最后的输出结果。

随机森林在决策树和Bagging的基础上进一步引入了“随机”性质。“随机”性体现在两个方面：在随机森林中，对结点的分裂过程中使用到了随机的特性，不像传统决策树在节点分裂过程中，总是从当前的特征集中选取最优的特征，随机森林在待分裂节点中随机选择包含k个特征的子集，再从此子集中选取一个最优特征进行分裂。简而言之，“随机”体现在随机选取子集，子集中的特征也是随机的。随机森林的生成过程同Bagging。

### K近邻

K近邻算法（k-nearest neighbor,k-NN）也是一种基本的分类与回归方法，由最近邻算法扩展而来。K近邻算法的基本流程是：在给定的训练数据集T上，对新输入的样本x，在T上找出与x最相近的k个样本。对于分类问题，k个样本多数属于某个类，那么这个样本也属于某个类；对于回归问题，将k个样本的均值作为此样本的预测值。

K近邻算法利用训练集对特征向量空间来进行划分，并将其作为生成的模型，在该模型中，k值的选择、距离的度量以及决策规则是三个基本要素[20]。由于在使用K近邻算法时需要加载训练集中的全部数据，并需要计算所有样本点与待预测样本点之间的距离度量，因此在求解大规模预测问题的时候如果用简单的线性扫描算法来计算会有很高的复杂度，通常是通过构造kd树来产生特殊的存储训练数据，以减少计算的复杂度。

### 梯度提升决策树

梯度提升决策树GBDT（Gradient Boosting Decision Tree）模型是一种用于解决回归问题的树模型。当设定阈值时也可以用于二分类任务，不太适合做多分类问题。模型的输入是带标签的样本集合，输出是n棵树组成的回归树，每棵树都有若干个叶子结点，每个叶子结点都带有一个权值，该权值是本叶子结点上的样本的标签的平均值。具体算法步骤如算法1所示[12]。算法在第1步处初始化，得到使得损失函数极小化的一个常数，此时生成只有一个根结点的树。第2(a)步计算损失函数的负梯度在当前模型的值，将其作为残差的估计（对平方损失函数，它就是通常所说的残差，对于一般损失函数，它就是残差的近似值）。第2(b)步估计回归树叶结点区域，以拟合残差的近似值。第2(c)步更新回归树。第3步输出得到的最终模型。GBDT用到的参数如下所示：

L: 损失函数

y: 样本x的真实值

: 当前的回归树模型

M: 决策树棵树

: L负梯度在第m棵决策树第i个样本取值

: 第m棵树的叶结点区域

: 第m棵树第j个叶子结点的权值

I: 指示函数

**算法1：GBDT梯度提升决策树算法**

**输入**: 训练集，损失函数

**输出:** 回归树

1. 初始化
2. 对m=1,2,…,M  
    (a) 对i=1,2,…N, 计算损失函数的负梯度在当前模型的值

(b)对rmi拟合一个回归树，得到第m棵树的叶结点区域Rmj，j=1,2,…J

(c)对j=1,2,…J,计算

(d)更新

1. 得到回归树

### XGBOOST

XGBOOST（eXtreme Gradient Boost）由陈天奇在2015年提出[24]，该算法是在传统的GBDT上改进而来，是GBDT的一种快速实现。它能够充分利用计算资源进行并行计算，因此速度快；同时由于改进了算法的整体框架逻辑，因此可跨平台进行运算且容错性能更好。还能提高算法的精度

XGBOOST算法在GBDT上改进的主要有两点。一是在损失函数中做了二阶泰勒展开，二是在目标函数之外加入正则项，用来权衡目标函数的下降和模型的复杂度，以避免过拟合。

## 集成学习

集成学习（ensemble learning）是一种通过生成多个学习器来完成学习任务，它也被称之为多分类器系统（multi-classifier system），基于委员会的学习（committee-based learning）。

集成学习的学习过程可以分为两个部分：首先生成一组个体学习器（individual learner），然后通过结合策略将这些个体学习器产生的结果结合起来（如对个体学习器产生的结果简单取平均）产生最后的输出。

### 两类典型的集成学习方法

个体学习器包括同质（homogeneous）的学习器，如神经网络集成的个体学习器全是神经网络，梯度提升决策树中的个体学习器全是决策树，此时的个体学习器又称为基学习器（base learnner）。个体学习器也可以是不同类型的，如同时包括神经网络和决策树，此时的集成学习是异质的（heterogenous）。根据个体学习器生成的不同方式，集成学习可以分为两大类，一类是个体学习器以串行方式生成，这以Boosting为代表；一类是个体学习器之间以并行方式生成，以Bagging为代表。

Boosting的训练过程呈现阶梯状，基学习器按次序逐次生成（在具体的实现上可以达到并行生成，如XGBOOST），基学习器的训练集在每次训练的过程中都会按照某种策略重新生成，通过对所有基学习器的预测结果进行综合而产生最后的预测结果。其中，Boosting中的基学习器都是弱学习器，所谓弱学习器是指泛化性能略好于随机预测的学习器，如在二分类问题中的预测精度略高于50%的学习器。因而，弱学习器具有偏差高，方差小的特点，即泛化性能弱，防止过拟合能力强的特点。

提升树模型如图2-1所示。

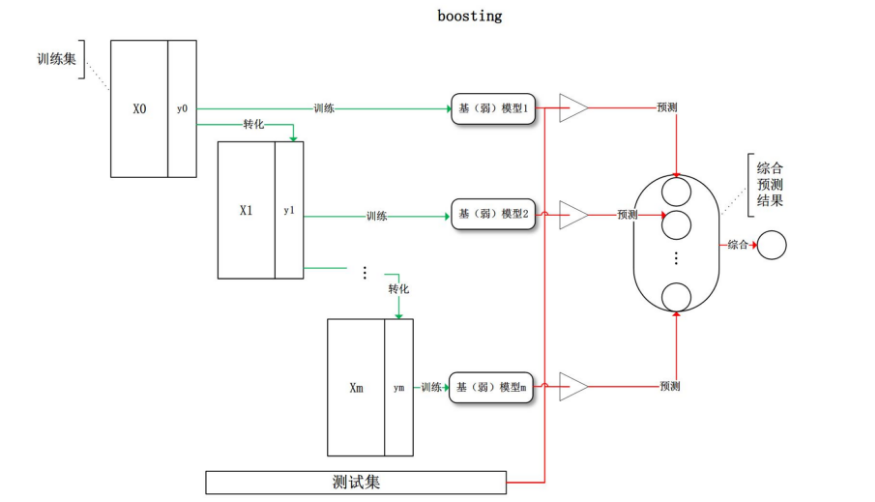


图2-1 Boosting模型

Bagging在训练过程中，通过对训练集有放回的重复采样，采样出T个含m个样本的采样集合，对这T个采样集合分别训练，得到T个基学习器。对于分类问题，Bagging通常采用简单投票法；对于回归问题，使用简单平均法来得到最后的预测结果。Bagging中的基学习器通常都是强学习器，具有偏差低而方差高的特点，即泛化性能强，防止过拟合能力弱的特点。Bagging模型如图2-2所示。

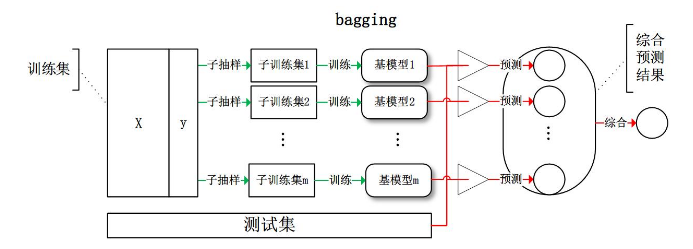


图2-3 Bagging模型

### 集成学习中的结合策略

假定某集成学习过程中包含T个基学习器，基学习器在样本x上的输出为.常见的结合策略有如下三种[19]。

### 2.4.2.1 平均法

当基学习器的输出连续数值型结果时，常用的一种结合策略是直接使用简单平均法：

另外一种策略是对基学习器的结果进行加权平均，其中是相应个体学习器的权重，研究发现[26]只有当权重非负时，才会有集成性能优于单一个体学习器的学习结果。

在研究中发现[27]，加权平均法在某些情况下可能不会比简单平均法性能更好。通常，当个体学习器性能相差比较大的时候不适宜采用简单平均法，而采用加权平均法。当它们的性能接近时，才适宜采用简单平均法。

### 2.4.2.1 投票法

在分类任务中，常常对学习器的预测标记采取投票的方式得到最后的结果。学习器对样本x的预测可以记作N维向量，其中是学习器在类别上的输出。

* 绝对多数投票法：若某个标记的得票数超过一半，那么就预测该样本为此类，否则拒绝预测。
* 相对多数投票法：即预测为得票最多的标记，如果同时又多个标记活得最高票，那么随机选取一个即可。
* 加权投票法：类似于加权平均法，对所有学习器的输出加权求和。

### 2.4.2.1 学习法

当在集成学习中，若训练数据相对较多，那么可以采用Stacking集成学习算法[27]来进行结合。Stacking集成学习算法的工作流程如下：

* 首先从原数据集中训练出多个学习器
* 通过训练出的学习器产生各自的输出，组合成新的训练集
* 新的训练集的标签还是原来训练集中的标签，在新的数据集上再次利用机器学习算法进行预测。

本文采用个体学习器为同质决策树的梯度提升决策树作为本文提出的模型-分层集成模型（Hierachrical Ensemble Learning Model, HELM）模型的核心，并在二分k-均值的基础上使用Stacking集成学习算法对酒店价格进行预测。本文的价格预测模型定义如下：

用表示获取到的所有酒店价格样本集合。每一个样本包括两个部分，表示某酒店在某时刻的各项特征（酒店ID,数据获取时间，酒店各项基本信息，当前显示的价格至未来第七天的价格）,表示样本对应的标签，即未来第八天的真实价格。本文的研究目标即是需要通过对训练集进行训练而得到最优的回归树，即得到对酒店价格的预测模型。

价格预测结果以平均绝对百分误差(Mean absolute percent error, MAPE)作为评价标准.其中At表示样本的真实标签，Ft表示样本的预测值。其计算公式如下所示：

2-7

## 本章小结

本章介绍了与本文研究相关的背景知识。首先介绍了机器学习的概念和类别，随后重点介绍了聚类算法中的二分k均值算法，本文用到的四个机器学习算法，最后介绍了集成学习的概念。

1. 数据采集与预处理分析

本章将主要介绍酒店及其价格的基本特征，价格变动规律。我们依次从数据的获取、数据预处理和数据分析来进行介绍。

## 数据的获取

我们基于携程的酒店搜索接口编写了多线程爬虫程序来抓取酒店的价格数据。由于我们要获取带有时间因素的价格数据，因此程序固定在每日的0点、4点、8点、12点、16点和20点启动，每次并行爬取18个城市的价格数据。对于每一个城市，爬取最热门的400家酒店的当前预定当日至未来第八天的价格。由于携程网的酒店展示界面每页只有15家酒店，因此在每个时间点，需要进行次访问搜索接口。整个程序从2016年3月18日一直运行至2016年5月16日。爬虫架构如图3-1所示。

携程酒店搜索

酒店动态价格爬取模块

（爬取酒店的动态价格）

酒店静态信息爬取模块

（爬取酒店的静态信息，如经纬度，星级，评论数）

待

抓 取

的

酒

店

序

列

酒店动态价格数据库

酒店动态价格数据库

图 3-1 爬虫架构图

由于是根据酒店静态信息爬取模块爬取到的酒店来进行酒店价格爬取，因此在静态信息爬取这一步就已经根据酒店的星级和评论量筛选出了400家最热门的酒店，这主要是为了排除静态信息过少会导致动态价格过少这种情况。

## 数据的预处理

爬取得到的原始数据分成两类：

**酒店静态信息** 包括酒店ID，酒店星级，酒店评分，酒店评论量，酒店设施，酒店位置，酒店所处区域，酒店经纬度，酒店最新预定时间等信息。

**酒店动态价格信息** 包括酒店ID，爬取日期，爬取时间，入住时间，当晚入住至未来第八天每晚入住的价格等信息。

由于爬取到的数据杂乱且冗余现象严重，对数据的处理需要进行复杂的分析。所有的数据分成两个部分：酒店静态信息表，酒店价格信息表。因此，首先进行的是依据酒店ID将酒店静态信息分别加入到酒店动态信息数据库中去；其次是整理数据中的缺失数据，缺失数据包括空值和取值异常（比如价格数据应为数值型，获取得到的可能为“-”等）两种。然后删除掉缺失价格数据超过一半的样本，并以均值填充价格缺失值，并将所有数据转换成数值型；最后是为每个样本找出其对应的标签，即未来第八天入住的真实价格。通过上面的处理，得到一个初步可用的数据集。

## 价格特征分析

通过上面的数据预处理步骤，我们可以得到如下的一些数据特征。以下图示的价格都是指的平均价格。

图3-2位不同时间点对应的提前八天预定价格和第八天当日预定价格对比图。通过该图可以看出提前预定比入住当天预定价格会低大约5%，这符合实际生活中越早预定价格越低的特点。其中，提前预定的价格随着时间的推移而逐渐增加，到晚上8点的时候，提前八天预定的价格达到最高，并且与第八天当日预定价格相当。当日预定价格在总体上也呈现上涨趋势，在每日的16点此价格达到最高，随后逐渐降低。这一点也侧面反映了下午是客流的高峰期，时间越晚，客流量

越低。

图3-2 爬取时间与均价对比图

图3-3显示了酒店在不同评分情况下的提前八天预定价格与第八天当日预定价格对比。这里的评分是入住过的用户给予的平均打分。该图显示了两个趋势：第一，提前预定的价格普遍要低于入住当日再预定的价格；第二，随着评分的提升，酒店的均价也在不断上涨，其中所获得的酒店评分大都集中在3.6到4.8这个区间，而评分4.9的酒店只有5家，评分5.0的只有一家。因此，出现小部分的“异常”也是在实际的经验之中。评分4.9和5.0的酒店均价不到200，这可以显示出不少人对酒店的价格比较敏感，只要提供了具有性价比的服务，都能够收到用户良好的评价。

图3-3 酒店评分与均价对比图

图3-4为不同“星级”下的提前八天预定价格与第八天当日预定价格对比。这里的“星级”有两种，一是政府评定的星级，最高到5星；另一种是携程用户给予的评定，最高是5.5星。在Grades低于2.5时，由于价格较低，两种价格变动不明显；在星级高于2.5时，即当前预定未来第八天的酒店比第八天再预定更划算，这个差价也随着星级的增加而不断加大。

图3-4 酒店评分与均价对比图

图3-5为在不同星期条件下酒店的均价对比图。这里的星期值指样本获得的日期，比如“星期四”表示的是当前预定下周五的均价和下周五入住时再预定的均价。从图上可以看出客房均价从星期三开始逐渐上升，到星期六的时候达到最高值，星期天开始逐步回落。其中，星期五星期六的两个均价差值最大，且星期六均价较最低的星期二高了约10%。

图3-5 星期与均价对比图

## 本章小结

本章分别从数据获取，数据预处理，酒店价格特征处理这三部分进行了介绍。从酒店价格特征分析处可以看出，我们爬取的价格数据符合正常的市场规律，并能够为后序研究提供部分统计信息。

在第四章中，将主要介绍基于传统机器学习算法的酒店价格预测模型。

1. 基于传统机器学习算法的酒店价格预测

本文的第二章中对将要用到的四个机器学习算法进行了详细的介绍，同时根据主要讨论基于传统机器学习算法的酒店价格预测过程。首先介绍酒店价格预测的定义，然后详述特征工程和各个机器学习算法的预测流程。

## 酒店价格预测问题定义

针对特定的酒店价格数据集，预测酒店的价格变化情况是一个复杂的问题。当前类似研究都是针对特定城市的数据集，并且到目前为止还没有针对国内城市，使用酒店的静态信息和酒店的动态历史价格来建模以对未来的价格进行预测。为了提升用户与预定平台之间的信息鸿沟，酒店价格预测是一个值得研究的问题。

本文不像旅游管理方向研究者使用特征价格模型（hedonic pricing model）来对酒店价格进行评估，而是采用多种机器学习算法来通过历史价格进行对未来价格的预测。要提升预测的准确度，需要解决多方面的问题。其间有两个问题最为重要，这是本文需要解决的主要难点：

1. 爬取的数据会存在缺失、异常值等情况（上文的数据预处理只进行了价格缺失的处理），如何将其可应用于机器学习算法是首先需要解决的问题。对数据的处理，需要进行特征工程，因为数据和特征往往决定了机器学习能够达到的最好效果，而各种模型和算法只是逐渐的朝着这个上界逼近。
2. 如何在处理好的数据上选用预测效果好的算法。由于不同的算法对数据的要求不一样，比如逻辑回归通常需要将数据归一化，以便模型训练能够快速训练完成，否则可能造成不收敛。然而，XGBOOST可以不用归一化，也可以不用处理缺失值就能够直接进行模型训练。因此，针对不同算法的调参将是本章的重点。

## 特征工程

特征工程是使用与数据相关的领域知识来创造出特征以使得机器学习算法得以工作。特征工程是机器学习的基石，同时它也是困难且耗时的。特征工程的目的是最大程度地从原始数据中获得特征以供机器学习算法使用。在实际的工程中，特征工程一般包括如下三个部分：

（1）特征使用方案。要完成既定的预测目标，我们需要基于对预测问题的理解，尽可能多地找出对于预测目标有影响的特征。对于欲获取的特征，还要进行可用性评估，包括获取难度，获取的准确率，特征的覆盖率。

（2）特征的获取方案。在这一部分工作中，需要考虑两个方面，一是如何获取特征，二是对获取到的特征以何种方式存储。

（3）特征分析。这是特征工程中最重要也最耗时的部分，该部分的工作包括两个大类：一是特征清洗，二是特征预处理。

1）针对特征清洗，我们要进行异常样本检测，比如某酒店的价格大部分时候都不超过500元，然而突然某一天的价格提升到了3000元，那么这就是一个异常样本，我们需要根据实际情况来简单丢弃此样本或者是用均值代替异常值；此外，若原始数据集不均衡（正负样例差别太大，样本权重差别大），那么还需要进一步的进行采样工作。

2）特征预处理工作需要对两种情况的特征进行处理：

针对单个特征，若某特征的变化范围过大，大么就要进行归一化操作；对于类别特征，需要进行离散化操作；对于某些取值有限的特征（如酒店所在区域），需要进行离散化操作，将其转换为不同的类别。

针对多个特征要考虑降维和特征选择，如果特征过多，需要进行进行降维操作。常见的方法有主成成分分析算法（Principal Component Analysis,PCA）和线性判别式分析（Linear Discriminant Analysis,LDA）。PCA通过线性投影，把高维的数据映射到低维的空间中表示，目标是使数据的方差在所投影的维度上达到最大，以使得较为完整的保留原始数据的特性的同时减少数据的维度。LDA通过将高维的样本映射到最佳判别矢量空间，其目标是使得同一类别的数据点尽量接近，不同类别的数据点尽量分开。对于特征选择，有多种方法，常用到的有三种方法：第一是种通过方差选择、相关系数、卡方检验等方式来获取自变量和目标变量之间的关联，以及自变量之间的相互关系；第二是通过目标函数来决定是否加入一个新的特征；第三是通过我们的机器学习算法生成的模型来判断特征的重要性，实现特征的自动选择。

本文中采用了python中sklearn模块中的VarianceThreshold方法来首先对各个特征的方差进行计算，根据设置相应的阈值，排除掉方差小于阈值的特征，方差选择应用于数据集准备阶段。此外，对于本文用到的树模型，本文使用了feature\_importances\_这一方法来对重要特征进行选择，并在重新选择重要特征后进行重复训练并预测。

以下将对本文的特征工程按照如上所述做出详细介绍。

### 特征使用与获取方案

在本阶段的目的是尽可能多地找出对预测目标有影响的特征，并进行评估特征的可用性及获取与存储方式。

本文用到的数据集是南京市在2016年3月18日至5月16日的酒店价格数据。原始数据中包括389家酒店共计82536个样本，每个样本有74维特征:

酒店名，酒店ID，酒店所在区域，用户/政府评级，星级，酒店评分，评论量，可否预定，现金，最新预定时间，wifi，抓取时间，餐厅，接机服务，泳池，健身房，停车服务，平均得分，位置得分，清洁得分，服务得分，开业时间，客房数量，与市中心距离，酒店链接，酒店小图链接，酒店大图链接，价格是否隐藏，定金，定金消费，经度，纬度，有无促销，有无团购，团购类型，会员指南，会员折扣，团购价格，hasAverageRate，averageRate，taReviewsCount，taReviewUrl，landMarkDistance，有无促销，促销文本，促销指南，当日入住至未来第二十八日入住的价格。

从3.3节中的特征价格分析可以看出，酒店的自身因素（如星级，评分）和时间因素（如爬取时间，是否周末）都会对预测目标有所影响。因此，本文主要是从这两个方面进行特征的选取，选取的特征多数是酒店自身因素与在时间变量下的变化情况。包括如下29个特征：

根据评分和星级对酒店进行聚类，将每家酒店分到对应的簇中。

同一评分酒店数量；同一星级酒店数量；同一簇中酒店数量。

样本前7天的平均价格；7天中价格是否有变化；7天中价格变动的次数；

同一评分中价格变动的酒店数量；同一评分中价格变动占比；

同一星级中价格变动的酒店数量；同一星级中价格变动占比；

同一簇中价格变动的酒店数量；同一簇中价格变动占比；

同一开业时间中价格变动的酒店数量，同一开业时间中价格变动占比；

入住当日是否周五或周六（这两天的平均价格最高）；入住当日是本月的第几天；

评论数量相对上一个时间点是否有变化；评论数量相对上个时间点的变化数；同一个簇中评论数量变动的酒店数量；同一星级中评论数量变动的酒店数量；同一评分中评论数量变动的酒店数量；同一开业时间中评论数量变动的酒店数量；

最新预定时间相对上一个时间点是否有变化，最新预定时间相对上个时间点的变化数，同一个簇中最新预定时间相对上一个时间点变化的酒店数，同一星级中最新预定时间变动的酒店数量；同一评分中最新预定时间数量变动的酒店数量；同一开业时间中最新预定时间变动的酒店数量。

由于上述特征都可以从原数据集中组合生成，且样本容量并不大，直接存放于本地即可。

### 特征分析与特征监控

本文在特征分析这一阶段主要做的工作是异常值检测、缺失值填充、归一化、离散化操作和在算法中判断特征的重要性以实现特征的自动选择。

本文面临的异常值主要是价格出现异常增高的情况，出现的概率较小，以均值代替。对不同特征值缺失采取不同的方式，如对开业年份或者酒店评分这类缺失较少的特征采取手动补全，即通过领域知识，找出这类特征的真实值来填充；而对价格这类特征，若一个样本中价格缺失值超过一半，那么简单的丢弃该样本。对取值变动范围较大的特征，我们往往要做归一化操作，比如酒店的评论数量。对酒店的类别特征往往进行了离散化操作，如星期。

在实验的算法中，一直保持对特征重要性的监控，排除小于阈值的特征。

## 实验设置

### 硬件配置

本实验所用到的计算机硬件配置如表4-1所示

表4-1 实验所用计算机硬件配置

|  |  |
| --- | --- |
| 硬件 | 配置 |
| CPU | Intel(R) Xeon® CPU E5-1620 V3 @ 3.50GHz |
| 内存 | 16GB |
| 硬盘 | 1TB |

### 软件环境

1. 操作系统  
   本文的实验采用的操作系统为Ubuntu14.04
2. 程序设计语言

本实验中的所有算法均使用python编写，版本为2.7.4。集成开发环境为Spyder2.0。

### 实验数据描述

本章使用到的数据4.2.1中描述的爬取自携程网的实时客房价格数据，经过初步处理后数据的描述如表4-2所示。

表4-2 酒店客房价格数据集

|  |  |
| --- | --- |
| 描述 属性类别 | 属性个数 样本个数 |
| 训练集 离散、连续 | 74 82536 |
| 测试集 离散、连续 | 74 10208 |
| 酒店数 --- | --- 389 |

### 模型验证

模型验证是为了验证模型性能的一种统计分析方法，同时也是为了避免过拟合。技术分为外部验证和内部验证两种。模型外部验证是指把训练好的模型应用于来自相同分布的全新的数据集上面，这种验证方法适用于数据量较大的情况。然而，在实际应用中，数据往往比较匮乏，因此，我们往往使用相同的数据来对模型进行评估，这就是内部验证法。在内部验证法中，常用的有交叉验证法。交叉验证的思路是首先把数据集分成两部分，一部分用于构建模型，另外一部分用于验证模型。交叉验证的方法有多种，常见的有如下三种：

1. 简单交叉验证法（Hold-Out Cross Validation）

该方法随机地将原始数据分成两组，一组用作训练集，一组用作验证集。在训练集上面训练出分类器，再在验证集上面对模型进行验证，得到最后预测的准确率作为本模型的性能指标。简单交叉验证法的好处是计算量少，缺点是由于对数据集是随机分成两组，因此最后在验证集上面的预测准确率与原始分组方式有很大关系，通过这种方式获得的结果在实际应用中并不具有可行性。

1. 留一交叉验证法（Leave-One-Out Cross Validation， LOO-CV）

假如初始数据含有n个样本，那么留一交叉验证法将n-1个样本作为训练集，剩下的1个样本作为测试集，因此会得到N个模型，将这些模型的预测平均值作为此模型的性能指标。该种验证方法中用到的训练数据最多，得到的结果比较可靠，其次在训练过程中没有引入随机因素，因此预测结果是可重复的。但是，该验证方法也有比较明显的缺点，当初始数据集过大时，计算成本将急剧增加，因此适用于样本较小的模型验证。

1. K折交叉验证法(K-fold Cross Validation)

K折交叉验证法首先将初始数据分成K组，将K-1组数据作为训练集，剩下的一组作为验证集，依次执行K次，得到K个模型。将这K个模型在验证集上面得到的平均预测准确率作为该模型的性能指标。通过K折交叉验证法即克服了留一验证法计算量大的缺点，有能够有效的避免过拟合与欠拟合。

本文的实验采用K折交叉验证法来对模型进行验证，K的取值为5、8或者10，算法采用scikit-learn中的GridSearchCV方法来实现。

## K近邻预测

在K近邻学习中，我们对给定的某个测试样本，根据距离度量找出样本集中与其最为靠近的k个样本，对于分类问题，可以通过投票法来进行最终预测，对于回归问题，可以使用平均法对k个样本的实际值作平均来当作样本的最终输出。因此，K值、决策算法将是本文调参的主要考量点。所得到的预测结果如表所示：

表4-3 K近邻预测平均绝对百分误差

|  |  |
| --- | --- |
| C K alg | MAPE |
| 5 2 ball\_tree | 0.1336 |
| 5 2 kd\_tree | 0.1335 |
| 10 2 ball\_tree | 0.1336 |
| 10 2 kd\_tree | 0.1336 |
| 5 4 ball\_tree | 0.1263 |
| 5 4 kd\_tree | 0.1263 |
| 10 4 ball\_tree | 0.1263 |
| 10 4 kd\_tree | 0.1263 |
| 5 6 ball\_tree | 0.1248 |
| 5 6 kd\_tree | 0.1248 |
| 10 6 ball\_tree | 0.1248 |
| 10 6 kd\_tree | 0.1248 |
| 5 8 ball\_tree | 0.1243 |
| 5 8 kd\_tree | 0.1243 |
| 10 8 ball\_tree | 0.1243 |
| 10 8 kd\_tree | 0.1243 |

从上面的实验可以看出，在本实验中，预测效果随着K值的增加而变得更好，并且与交叉验证次数C，决策算法都没有明显的联系，因此考虑到算法的效率选取K等于8，C等于10，alg为kd\_tree作为算法的最优参数，得到的最优MAPE为0.1243。

## 随机森林预测

由于随机森林是在以决策树为基学习器的基础上构建的Bagging方法的变体，因此随机森林的参数主要分为两个部分，一部分是Bagging框架相关参数，一部分是CART决策树的参数。

随机森林中的Bagging框架参数包括如下三种：

n\_estimators作为弱学习器的最大迭代次数，也称之为弱学习器的个数。当该参数太小的时候，由于学习到的信息太少而导致欠拟合的发生，当n\_estimators过大又会因为过学习而造成过拟合的发生，该值默认是10。

oob\_score是用来表征是否用袋外样本来对模型进行评估，由于袋外分数反应了训练好的模型的泛化能力，可以将默认为False的oob\_score设置为True。

criterion是用来评估分类和回归树在分类时的误差，在回归问题中默认使用均方误差（mean squared error, mse），这和分类模型的默认参数不一致，分类模型中的默认参数是基尼系数gini和可选的参数信息增益。

随机森林中的决策树参数主要包括如下7种：

max\_features是用于在分裂结点的时候考虑使用的特征数目，默认为auto，即使用所有的特征来对结点进行分裂。在实际的操作中，若预测问题中的特征数少于50，使用默认参数通常就能得到比较好的效果。当特征数目较多的时候，就需要多考虑使用多种参数来进行测试。

max\_depth是决策树的最大深度，默认为None，若采用默认参数，那么结点将会一直扩展直到叶子节点所包含样本小于参数min\_samples\_split\_samples或者所有叶子节点都不能再分裂为止。在数据或者特征较少的情况下，该值使用默认参数即可。当样本多或者特征较多的情况，需要对此参数进行限制，以避免消耗过多的计算资源。该参数默认为None，常用取值范围在10-100之间。

min\_samples\_leaf叶子节点中最少的样本数，通过该参数来限制结点继续分裂，并用在优化决策树，当叶子节点中的样本数少于此参数，那么该叶子节点将会和兄弟结点一起被剪枝。此参数默认为1，在样本数量不够大时，用默认值即可，否则需要增大该参数。

min\_samples\_split是内部结点进行分裂时所需要的最小样本数。同样，此参数限制了结点的继续分裂，此参数默认是2。与min\_sampes\_leaf类似，其取值随样本容量的变化而变化。

min\_weight\_fraction\_leaf是叶子节点最小的样本权重和，默认为0，意即对于叶子节点的权重不做考虑，当训练样本中存在较多缺失值，或者样本不均衡，此时将考虑对此参数做出修改操作。

max\_leaf\_nodes是最大叶子节点数，默认为None，意即不限制最大叶子节点数。当特征较多的时候，才考虑修改此参数，最优的值可以交叉验证得到。

min\_impurity\_split是一个用来限制决策树分裂的参数，如果某结点的不纯度（在回归问题中用均方差来计算，分类问题中用基尼系数来计算）小于此参数，那么节点将停止分裂，成为叶子节点。

由于随机森林的基学习器具有偏差低的特点，即基学习器都不为弱模型，故sklearn中随机森林中设置决策树最大深度的max\_depth为None。随机森林在训练过程中主要目的是为了降低方差，而引入越多的基学习器，方差就会越大，因此在sklearn的基学习器个数参数n\_estimators默认为10。根据数据集的实际情况，本文在对随机森林调参过程中，主要考虑交叉验证的K值、基学习器个数、决策树最大深度这三个参数对最终预测结果的影响，所得模型的平均绝对百分误差如表4-3所示。

表4-3 随机森林平均绝对百分误差（10棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1158 |
| 5 6 | 0.1174 |
| 5 7 | 0.1148 |
| 5 8 | 0.1141 |
| 10 5 | 0.1163 |
| 10 6 | 0.1165 |
| 10 7 | 0.1149 |
| 10 8 | 0.1161 |

表4-3 随机森林平均绝对百分误差（20棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1159 |
| 5 6 | 0.1156 |
| 5 7 | 0.1156 |
| 5 8 | 0.1153 |
| 10 5 | 0.1160 |
| 10 6 | 0.1165 |
| 10 7 | 0.1147 |
| 10 8 | 0.1147 |

表4-3 随机森林平均绝对百分误差（50棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1165 |
| 5 6 | 0.1165 |
| 5 7 | 0.1149 |
| 5 8 | 0.1144 |
| 10 5 | 0.1154 |
| 10 6 | 0.1158 |
| 10 7 | 0.1147 |
| 10 8 | 0.1149 |

表4-3 随机森林平均绝对百分误差（100棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1162 |
| 5 6 | 0.1156 |
| 5 7 | 0.1150 |
| 5 8 | 0.1147 |
| 10 5 | 0.1160 |
| 10 6 | 0.1157 |
| 10 7 | 0.1147 |
| 10 8 | 0.1150 |

从上面的实验中可以看出，随机森林的总体预测比较稳定，不仅体现在树的棵树变动时得到的预测结果也相对稳定，在使用不同的交叉验证折数也有同样的效果。并且，在不同的参数情况下都没有出现过拟合的现象，这也体现了随机森林的不会出现过拟合[25]的优点。同时，可以看到在树的深度max\_depth为7和8的时候，不管树的棵树是多少，得到的MAPE总是小于相同条件下max\_depth为5和6的情况。本次实验的最佳结果在n\_estimators等于10，k等于5，max\_depth等于8的条件下取得，得到的最优MAPE为0.1141。得到的各特征对最终结果影响的比重如图4-1所示，其中前七天的平均价格、当前是一个月中的第几天，同一评分中酒店在7天中有变化的数量这三个特征对最后的预测起了最大的作用，其中影响最大的特征中大部分都是经由特征工程重新生成的组合特征，这可以看出特征工程在解决问题中的重要性。

图4-2 随机森林中特征重要性对比

## 梯度提升决策树预测

在GBDT回归模型中，参数也可以分为两类，一类是Boosting框架的参数，一类是弱学习器CART回归树的参数。此处以本文中取得最佳预测结果的GBDT参数选取作为示例。

Boosting框架参数主要包括如下三种:N\_estimators，弱学习器的最大迭代次数，即最大的弱学习器的个数，默认是100。learning\_rate，即每个弱学习器具的权重缩减系数，也称之为步长，默认是1。Loss即损失函数，由于本实验是一个数据噪音不多的回归模型，因此选择均方差ls即可。

GBDT类库弱学习器参数主要包括如下五种。划分是考虑的最大特征数max\_features，可以使用多种类型的值，默认为“None”，由于本实验中的维度为104维，特征数并不大，使用默认的即可。决策树的最大深度max\_depth，默认可以不输入，这样决策树在建立字数的时候不会限制子树的深度。叶子结点最少样本数min\_samples\_leaf，用来限制一个叶子结点中包含的最少样本数，默认是1，如果某叶子结点数目中小于该样本数，那么该叶子结点将会和兄弟结点一起被剪枝。叶子结点最小的样本权重和min\_weight\_fraction\_leaf，用于限制叶子结点所有样本权重和的最小值，默认是0，如果小于该值，则会和兄弟节点一起被剪枝。最大叶子节点数max\_leaf\_nodes，通过限制该参数，可以防止过拟合，默认为“None”。

与基于Bagging的随机森林不同的是，基于Boosting的梯度提升决策树的基学习器都为弱学习器，因此其最大深度max\_depth默认为3；弱学习器具有较低的方差，但是偏差较高，因此在模型的整个训练过程中需要逐步降低偏差，这需要设置更多的弱学习器来实现，所以其弱学习器个数参数n\_estimators默认设置为100。在这一部分的调参过程中，同样考虑了交叉验证的K值、基学习器个数、决策树最大深度这三个参数对最终预测结果的影响，所得模型的平均绝对百分误差MAPE如表4-4至表4-7所示(MAPE下的取值从左到右是在学习率取[0.1,0.15,0.2,0.25]下得到)。

表4-3 GBDT平均绝对百分误差（20棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1345 0.1222 0.1224 0.1219 |
| 5 6 | 0.1365 0.1219 0.1213 0.1247 |
| 5 7 | 0.1346 0.1189 0.1167 0.1185 |
| 5 8 | 0.1342 0.1184 0.1178 0.1181 |
| 10 5 | 0.1346 0.1222 0.1225 0.1217 |
| 10 6 | 0.1365 0.1220 0.1211 0.1216 |
| 10 7 | 0.1347 0.1187 0.1169 0.1181 |
| 10 8 | 0.1345 0.1183 0.1179 0.1192 |

表4-3 GBDT平均绝对百分误差（50棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1243 0.1269 0.1272 0.1281 |
| 5 6 | 0.1178 0.1212 0.1259 0.1287 |
| 5 7 | 0.1202 0.1199 0.1215 0.1308 |
| 5 8 | 0.1162 0.1179 0.1208 0.1223 |
| 10 5 | 0.1232 0.1269 0.1272 0.1283 |
| 10 6 | 0.1179 0.1207 0.1251 0.1283 |
| 10 7 | 0.1176 0.1191 0.1213 0.1245 |
| 10 8 | 0.1159 0.1184 0.1210 0.1259 |

表4-3 GBDT平均绝对百分误差（100棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1255 0.1282 0.1281 0.1346 |
| 5 6 | 0.1210 0.1239 0.1276 0.1315 |
| 5 7 | 0.1195 0.1220 0.1251 0.1315 |
| 5 8 | 0.1177 0.1208 0.1244 0.1273 |
| 10 5 | 0.1378 0.1281 0.1284 0.1345 |
| 10 6 | 0.1199 0.1239 0.1272 0.1316 |
| 10 7 | 0.1206 0.1205 0.1251 0.1403 |
| 10 8 | 0.1174 0.1190 0.1255 0.1267 |

表4-3 GBDT平均绝对百分误差（200棵树）

|  |  |
| --- | --- |
| KFold max\_depth | MAPE |
| 5 5 | 0.1319 0.1289 0.1310 0.1326 |
| 5 6 | 0.1219 0.1316 0.1294 0.1329 |
| 5 7 | 0.1213 0.1259 0.1289 0.1341 |
| 5 8 | 0.1204 0.1242 0.1241 0.1314 |
| 10 5 | 0.1376 0.1286 0.1319 0.1387 |
| 10 6 | 0.1215 0.1314 0.1305 0.1344 |
| 10 7 | 0.1214 0.1257 0.1260 0.1331 |
| 10 8 | 0.1205 0.1223 0.1257 0.1292 |

从上面的实验可以看出，随着树的棵树逐渐增加到50，MAPE变得最佳，随后预测效果逐步变差。同时，当树的深度增加时，同样条件下的MAPE会逐渐降低。在本次实验中取得最好效果的参数为n\_estimators等于50，KFold等于10，max\_depth等于8，learning\_rate等于0.1，在此参数下得到的最优MAPE为0.1159。

最后，通过GBDT得到的特征重要性如图4-所示：

图4- GBDT中特征重要性对比

从上图可以看出，GBDT得出的特种重要性与随机森林所得大致相同，同时大部分排名靠前的都是通过特征工程生成的组合特征，这再一次说明特征工程的重要性。

## XGBOOST预测

XGBOOST与GBDT相比，拥有多种优势，比如对缺失值的处理和交叉验证。XGBOOST内置了对缺失值处理的规则，XGBOOST在不同结点处处理缺失值采用不同的方法，并会自动学习未来对缺失值的处理方法。同样，XGBOOST内置交叉验证方法，XGBOOST在每一轮的boosting迭代中使用到交叉验证，因此，可以容易地获取boosting迭代次数。

XGBOOST的参数共分为三类：第一，通用参数，用来实现函数的宏观控制，包括使用哪种booster以及在boosting过程中需要用到的最大线程数；第二，booster的具体参数，用来控制每一轮的boosting，第三，是学习目标参数，该参数用来控制学习目标的表现，这里面的有效数据的度量方法eval\_metric对于回归问题，默认是均方根误差，对于分类问题，默认是二分类错误率，由于默认的典型值里面没有我们的评价指标MAPE，因此可以自定义MAPE并传入。

XGBOOST预测过程中，一般使用如下的步骤来进行参数调节：

1. 选定一个较高的学习速率，即booster参数中的eta，默认值为0.3。通过XGBOOST中的“cv”函数在每一次迭代过程中使用交叉验证，从而得到在此学习速率下的最佳决策树数量。

2. 在上面确定的学习速率和对应的最佳决策树数量的条件下，进行决策树参数调优。本实验对三种参数进行了调优：最小叶子节点样本权重和min\_child\_wieght，树的最大深度max\_depth，指定节点分裂需要的最小损失函数减小值gamma；min\_child\_weight参数用于避免过拟合，当该值取较大时能够避免模型过度学习局部特殊样本，当该值过大时又可能导致欠拟合，因此需要进行参数调优。max\_depth也是用来避免产生过拟合，当max\_depth的取值越大，就能更容易学习到局部的样本，该参数的一般取值在3-10之间；由于只有当节点分裂后损失函数的值降低才会导致节点分裂，gamma用来确定节点分裂需要的最小损失函数下降值，默认为0。

3. 同样进行模型复杂度的调优，以降低过拟合风险。即调节booster参数中的lambda参数和alpha参数。

4. 减小学习速率，重新找出相关参数。

如下个表展示学习速率eta为[0.30，0.25，0.20]的情况下，min\_child\_weight取值[3,4,5,6]，max\_depth取值[6,7,8]，gamma取值为[0.2，0.3，0.4].

表4-3 xgboost预测平均绝对百分误差（max\_depth=6）

|  |  |
| --- | --- |
| eta min\_child\_weight | MAPE |
| 0.3 3 | 0.1110 |
| 0.3 4 | 0.1122 |
| 0.3 5 | 0.1147 |
| 0.25 3 | 0.1104 |
| 0.25 4 | 0.1296 |
| 0.25 5 | 0.1292 |
| 0.2 3 | 0.1369 |
| 0.2 4 | 0.1386 |
| 0.2 5 | 0.1361 |

表4-3 xgboost预测平均绝对百分误差（max\_depth=7）

|  |  |
| --- | --- |
| eta min\_child\_weight | MAPE |
| 0.3 3 | 0.1128 |
| 0.3 4 | 0.1117 |
| 0.3 5 | 0.1094 |
| 0.25 3 | 0.1203 |
| 0.25 4 | 0.1111 |
| 0.25 5 | 0.1105 |
| 0.2 3 | 0.1717 |
| 0.2 4 | 0.1412 |
| 0.2 5 | 0.1385 |

表4-3 xgboost预测平均绝对百分误差（max\_depth=8）

|  |  |
| --- | --- |
| eta min\_child\_weight | MAPE |
| 0.3 3 | 0.1066 |
| 0.3 4 | 0.1043 |
| 0.3 5 | 0.1036 |
| 0.25 3 | 0.1173 |
| 0.25 4 | 0.1150 |
| 0.25 5 | 0.1126 |
| 0.2 3 | 0.1409 |
| 0.2 4 | 0.1414 |
| 0.2 5 | 0.1397 |

由于gamma在设定的取值范围内对同等条件下的MAPE没有影响，故此处不做展示。从上面的结果可以看出，针对本数据集，当学习速率逐渐降低的时候，评测指标MAPE总体逐渐逐渐增大，并且当max\_depth取8，eta取0.3，min\_child\_weight取5时，我们将得到最优的MAPE为0.1066，这个结果是当前所有模型结果中最优。

## 本章小结

本章首先对酒店价格预测问题给予了定义，然后对特征工程和实验设置进行了详细的描述，最后对本文用到的四个预测算法参数给予了详细介绍，并介绍了如何通过这些算法来获取到最佳的平均绝对百分误差，保存下每个模型的最优参数以及生成出来的预测结果，该预测结果将会作为第五章中的Stacking框架的输入。

下一章将介绍二层stacking预测和本文提出的HELM模型。

。

1. 基于集成学习的酒店价格预测

在本章中，我们将在上一章的预测基础上利用Stacking集成学习算法和本文提出的HELM模型对酒店价格进行预测，并分析了影响模型预测的特征因素，最后对所有的预测结果进行了对比分析。

## 实验设置

### 硬件配置

本实验所用到的计算机硬件配置如表5-1所示

表4-1 实验所用计算机硬件配置

|  |  |
| --- | --- |
| 硬件 | 配置 |
| CPU | Intel(R) Xeon® CPU E5-1620 V3 @ 3.50GHz |
| 内存 | 16GB |
| 硬盘 | 1TB |

### 软件环境

1. 操作系统  
   本文的实验采用的操作系统为Ubuntu14.04
2. 程序设计语言

本实验中的所有算法均使用python编写，版本为2.7.4。集成开发环境为Spyder2.0。

### 实验数据描述

本章使用到的数据4.2.1中描述的爬取自携程网的实时客房价格数据，以及第四章中各个算法预测出最好效果MAPE的算法参数。

表4-2 酒店客房价格数据集

|  |  |
| --- | --- |
| 描述 属性类别 | 属性个数 样本个数 |
| 训练集1 离散、连续 | 74 82536 |
| 测试集1 离散、连续 | 74 10208 |
| 酒店数 --- | --- 389 |

## 二层Stacking预测

Stacking本身是一种著名的集成学习方法，很多集成学习算法都可以看作是Stacking的变体，同时Stacking也可以被看作是一种结合策略[19]。由于次级训练集是通过初级学习器预测产生而来，当直接使用初级学习器预测出来的结果生成次级学习器的训练集，则有产生过拟合的风险。因此，在本文的Stacking实验中，我们使用5折交叉来从初始训练集中随机划分出5个大小相同的样本集合，在每一折的训练过程中，针对此时测试集中的每一个样本，记录每一个初级学习器在该样本上的预测值，将每一个样本在所有的初级预测算法中得到的预测值和其原有的标签作为新的样本生成次级训练集。本节使用这种经过改进的Stacking算法来进行酒店价格预测，具体算法如算法4所示。

**算法4：一种改进的Stacking算法CStacking**

**输入**: 训练集

初级学习算法:K-nearestNeighborKnn(KNN), RandomForest(RF)，GradientBoostingDecisionTree (GDBT), XGBOOST(XGB)

次级学习算法：XGBOOST(xgb)

**输出:** 次级学习器

1. 初始化

for h in [KNN,RF,GDBT,XGB]:

= KFold(D)

end for

2.

3. for i =1,2,…,m do

for t = 1,2,3,4 do

end for

end for

4.

5.

我们使用第四章中得到的各模型最优参数来训练初级学习器，并从每一折的测试集中产生预测结果，将之作为次级学习器的输入。各模型的主要参数设置如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 算法 | 参数1 | 参数2 | 参数3 | 参数4 | MAPE |
| KNN | K=8 | Kfold=10 | alg=kd\_tree | - | 0.1243 |
| RF | n\_estimators=10 | Kfold=5 | max\_depth=8 | - | 0.1141 |
| GBDT | n\_estimators=10 | Kfold=5 | max\_depth=8 | learning\_rate=0.1 | 0.1174 |
| XGB | max\_depth=8 | min\_child\_weight=5 | eta=0.3 | - | 0.1066 |

以上的数据显示了，在我们的酒店价格数据集上取得最优效果的是XGBOOST，最差的是KNN，其次，通过上一张的实验发现，XGBOOST的训练速度最快，其次是GBDT，再次是随机森林，最差的是KNN。这是因为XGBOOST在GBDT的基础上进行了改进，支持并行处理，KNN的效率最差是因为KNN没有进行显示的训练，该算法在训练阶段仅仅把样本存储起来，没有进行训练，当接收到测试样本后再进行处理。

通过本节的改进Stacking集成学习算法预测，得到次级中GBDT的最优参数同其第初级学习器的最优参数，得到的最优MAPE为0.0938。各模型预测效果对比如图5-1所示。

图5-1 各算法预测效果对比图

## 分层集成学习模型HELM

本文在前面的实验基础上进一步地提出了改进算法来实现更为精确的酒店价格预测。本文选取的酒店价格数据采集自南京市，通过对酒店的经纬度分析（如图5-2），我们可以看出酒店在地理位置上有明显的类别特征。因此，本文提出了一个分层的集成学习模型HELM，其具体流程如算法4所示：



图5-2 南京市酒店分布图

**算法3** 基于GBDT和二分k-均值的酒店价格预测算法HELM。  
输入：原始训练集、K值集合

输出：类簇个数K与提升决策树

步骤：

1. 初始化
   1. 读入训练集Train
   2. 设置最大聚类个数集合K
2. for k from 2 to K：
   1. clusters\_k = bikmeans(Train,k)
   2. 对clusters\_k中每个类簇中的训练数据训练并设置不同的Booster参数和GBDT类库参数构建梯度提升树。
   3. 计算并存储在当前聚类数k下的MAPE。
3. 从以上K-1个聚类结果中选取MAPE最小的k值作为选取的聚类数，并以该参数下训练的GBDT类库参数构建梯度提升树。
4. 初始化聚类结果
5. 初始化lowestSSE =   
   (a) for i = 1,2…,|C| do:

i\* = i

添加到类簇集合*clusters*4.

本文提出的算法基本思想是计算每个is-a关系补全方法在部分抽样数据上的补全效果，剩余部分数据效用基于矩阵分解的算法进行预测，最终得到每个方法在整个数据上的补全的效果。因此本文设计实验由两个部分组成，共有八组，如表5-1所示。第一个部分是将该方法应用于Instance-of关系补全评估的实验，实验的目的分别有以下五个：

1. 验证该方法计算得出的准确率和召回率基本与真实情况符合。
2. 对比该方法与均值方法的差别。
3. 对比在不同矩阵密度上该方法和平均值方法的效果的。
4. 验证该方法提出的评估测度、和是否能真实反映每个方法的评估效果。
5. 验证该方法能否为知识工作者选择合适的is-a补全方法。

实验流程如下：

1. 加载每个方法及数据集。
2. 用每个方法补全数据集中的缺失的instance-of关系或者subclass-of关系。
3. 计算每个方法对所有概念的补全的准确率和召回率。
4. 分别得到完整的准确率矩阵和召回率矩阵，代表每个方法对每个概念补全is-a关系的准确率和召回率。
5. 为每个方法在步骤4生成的矩阵中，随机挑选部分数据构建稀疏矩阵。
6. 对步骤5形成的稀疏矩阵，利用矩阵分解算法SVD预测缺失的准确率和召回率，得到和，最终形成完整的矩阵准确率矩阵和召回率矩阵
7. 根据和计算评估测度加权准确率，加权召回率和加权F值。
8. 根据不同每个实验的目的，对矩阵、、和的数据进行分析。

本文提出的评估方法不仅能为instance-of关系补全选择一个适合的方法，同样也能为不同特点的数据选择一个效果最好的subclass-of补全方法。因此，我们设计了第二部分实验，将该方法应用于subclass-of关系补全评估。这部分实验目的是主要为验证该方法能为不同特点的数据选择合适的is-a关系补全方法。

表5-1 实验设计概括表

|  |  |  |  |
| --- | --- | --- | --- |
| 系列 | 编号 | 类别 | 目的 |
| Instance-of | 1 | 验证 | 验证该方法计算出来的准确率、召回率与真实情况基本符合。 |
| 2 | 对比 | 比较该方法计算出来的准确率、召回率比用平均值方法评估的结果。 |
| 3 | 对比 | 比较在不同密度下，利用SVD计算的误差和平均值得到的效果。 |
| 4 | 验证 | 验证该方法是否能为上下位关系完整，属性关系稀疏的数据选择最合适的方法。 |
| 5 | 验证 | 验证该方法是否能为上下位关系稀疏，属性关系完整的数据选择最合适的方法。 |
| Subclass-of | 6 | 验证 | 验证该方法是否能为instance-of关系完整，属性关系稀疏的数据选择最合适的方法。 |
| 7 | 验证 | 验证该方法是否能为instance-of关系稀疏，属性关系完整的数据选择最合适的方法。 |
| 8 | 验证 | 验证该方法是否能为属性关系部分缺失，instance-of关系完整的数据选择最合适的方法。 |

### 实验数据

本文提出的基于矩阵分解的is-a关系补全方法评估算法与具体的is-a关系补全方法无关，因此本文没有针对性地分别为instance-of关系和subclass-of关系选择了三个补全算法进行评估，其方法名和特点如表5-2所示，每个方法详细介绍请见第三章。

表5-2 待评估方法及其特征表

|  |  |  |
| --- | --- | --- |
| is-a关系类别 | 方法名 | 特点 |
| Instance-of |  | 基于随机森林的机器学习方法，特征是上下位关系，等价类等。 |
|  | 基于SVM的机器学习方法，特征是上下位关系，等价类等。 |
|  | 基于属性的概率统计计算。 |
| Subclass-of |  | 关联规则挖掘算法，基于instance-of关系的概率统计。 |
|  | 基于instance-of关系的概率统计，改良的支持度计算。 |
|  | 先用instance-of关系补全算法补全数据，在此基础上进行关联规则挖掘算法补全subclass-of关系。 |

本实验的数据来自于DBpedia3.9，DBpedia项目是将维基百科中包含的内容转变为结构化的知识，以RDF数据的形式保存下来，供人们以及各种Web应用的访问。实验利用DBpedia3.9构建了6个本体，其数据特点如表5-1所示。其中黄金标准数据集是Gold-standard（GS），它是Dbpedia3.9的数据，该数据集的作用是验证is-a关系补全的正确性和完整性。是待评估数据集，其中由部分Dbpedia3.9数据组成，我们通过jena[51]的API删除该数据上1/2实例的instance-of关系。是在的基础上，对每个实例删除1/2的属性关系，因此相对于来说就是属性关系稀疏，上下文关系完整的本体。是在的基础上去除1/4概念的上下位关系，相比于和的数据特点是上下位关系稀疏，属性关系完整。由部分Dbpedia3.9数据组成，对每个概念删除了1/2的继承关系。是在的基础上对每个实例删除1/2的属性关系，因此相比于，的数据特点是上下位关系完整，属性关系稀疏。是在的基础上去除1/2实例的instance-of关系，相比于，的数据特点是部分instance-of关系缺失，属性关系完整。

上述本体构建我们使用的编程工具是jena，主要用到了该项目的三个模块的，其中RDF API部分提供了核心接口去构建和读取RDF数据。ARQ部分提供了API供用户在RDF中构建 SPARQL查询。TDB提供高性能的RDF存储，使得项目对RDF数据读写都非常高效。

表5-3 实验数据集及其特征

|  |  |
| --- | --- |
| 数据集名称 | 特点 |
| Gold-standard（GS） | Dbpedia3.9数据，schema完整，实例上下位关系和属性关系完整。 |
|  | 部分DBpedia3.9数据，但部分缺失instance-of关系。 |
|  | 上下位关系完整，属性关系稀疏，部分缺失instance-of关系的DBpedia数据。 |
|  | 上下位关系稀疏，属性关系完整，部分缺失instance-of关系的DBpedia数据。 |
|  | 部分DBpedia3.9数据，但部分缺乏subclass-of关系。 |
|  | 每个实体instance-of关系完整，属性关系部分缺失。 |
|  | 每个实体部分instance-of关系部分缺失，属性关系完整。 |

### Instance-of关系补全评估实验详细设计

本节将从实验目的、输入、输出和预期实验结果几个方面来详细描述Instance-of关系补全评估实验的设计。

因为本文提出用SVD方法预测和来节省计算时间，提高计算效率。所以本文第一个实验设计是验证预测得到和与真实值的保持的在一定阈值内，我们利用RMSE来描述、与、之间的误差。RMSE的定义见公式4.6。如果该误差在一定阈值内，则认为该方法确实能在较大程度提高效率的情况下，较准确的估算每个方法的准确率和召回率。如果误差超过一定阈值，则认为该方法不能准确估算每个方法的准确率和召回率，为了保证实验的客观性，我们使用具有不同特点的3个本体数据集作为待评估数据集Gold-standard（GS）作为黄金本体，实验的详细设计请见表5-4。

实验2的目的是对比用SVD方法预测值得方法和平均值代替空缺值方法的效果，相应设计参数请见表5-5。因为在大量实际应用中，许多属性值缺失或者由于某些原因不可使用，如果属

性是离散值属性，则使用属性的平均值来填充缺失值[46]。因此本文使用均值方法作为基准，与

表5-4 Instance-of关系补全准确率和召回率误差计算实验设计表

|  |  |
| --- | --- |
|  | 实验1 |
| 目的 | 验证实验，验证该方法计算出来的准确率和召回率与真实情况基本符合。 |
| 输入 | Gold-standard（GS）+ ，Gold-standard（GS）+ ，Gold-standard（GS）+ ，所有方法集合，矩阵密度k。 |
| 输出 | 通过SVD预测得到的、及其RMSE。 |
| 实验预期结果 | SVD的误差（RMSE）在一个阈值范围之内。 |

本文提出的方法进行对比，如果实验结果是利用SVD计算的RMSE相比平均值方法得到RMSE值要小，则说明本文提出的方法准确性比均值方法更高。实验详细设计如表5-5所示。

表5-5 基于SVD预测的方法与均值方法效果对比实验设计表

|  |  |
| --- | --- |
|  | 实验2 |
| 目的 | 对比实验，验证该方法计算出来的precision、recall和F1比用简单的平均值方法评估的结果要更加精准。 |
| 输入 | Gold-standard（GS）+ ，Gold-standard（GS）+，Gold-standard（GS）+， Method集合，矩阵密度0.6。 |
| 输出 | 通过SVD预测得到的、及其RMSE。  通过均值方法计算得到， 及其RMSE。 |
| 实验预期结果 | 利用SVD计算的RMSE相比平均值方法得到RMSE值要小。 |
| 预估结论 | 本文提出的方法比均值方法准确度更高。 |

方法对本体O中概念的is-a关系补全的准确率均值和召回率均值的计算如公式5.1，5.2所示：

(5.1)

= (5.2)

其中n表示本体O中类别的总数，x表示对本体O进行了补全的类别个数。

由于SVD的预测效果受矩阵的密度影响较大，本文设计了实验3，详情请见表5-6。

表5-6 不同密度下算法效果对比实验设计表

|  |  |
| --- | --- |
|  | 实验3 |
| 目的 | 对比实验，在不同密度下，利用SVD计算的误差和平均值得到的效果的对比，以便用户一个设置一个合理密度。 |
| 输入 | Gold-standard（GS）+， Method集合，矩阵密度向量k。 |
| 输出 | 通过SVD预测得到的、及其RMSE。  通过均值方法计算得到， 及其RMSE。 |
| 实验预期结果 | 随着矩阵密度的增加，利用SVD和平均值预测真实值的误差逐渐降低。 |

构建了若干个不同密度值，将SVD计算效果与平均值得到的效果的对比，来客观地反应该方法的优劣。矩阵密度的计算如公式4.7所示，我们设置的密度向量K为[0.3, 0.4, 0.5, 0.6, 0.7, 0.8]。

通过以上实验基本能较全面的分析基于SVD预测instance-of关系补全的准确率和召回率的效果。实验4和5的目的是为了验证通过本文提出的评估计算方法和评估测度，，是否能准确的评估每个方法在整个本体上进行补全的效果。为了较为客观反映各个方法的在不同数据上的效果，我们分别构建了两组对比实验，实验4使用的两个数据的区别是属性关系是否稀疏，实验5使用的两个数据的区别是上下位关系是否稀疏。具体方案如表5-7所示。预期的实验结果该方法是可以自动考虑数据特点，为属性关系稀

疏的数据选择合适它的数据Method1，为上下位关系稀疏的数据选择合适它的数据Method3。从而可以说明该方法确实可以为不同特点的数据选择最合适的方法。

表5-7 instance-of关系下算法结果正确性实验设计表

|  |  |  |
| --- | --- | --- |
|  | 实验4 | 实验5 |
| 目的 | 对比实验，验证该方法是否能为具有不同特点的数据选择最合适的方法。 | |
| 输入 | Gold-standard（GS）+ ，Method集合。 | Gold-standard（GS）**+**  Method集合。 |
| 输出 | 最佳方法序号，及每个方法。、和。 | 同左。 |
| 数据特点 | 上下位关系完整，属性关系稀疏，部分缺乏instance-of关系。 | 上下位关系稀疏，属性关系完整，部分缺乏instance-of关系。 |
| 实验预期结果 | 。 | 。 |
| 原因 | 是基于上下位关系的机器学习方法。 | Method3是基于属性关系统计的算法。 |
| 预估结论 | 说明该方法的确能为具有不同特点的数据选择最合适的方法。 | |

通过以上5个实验，我们较合理且全面的考察了本文提出的方法的准确性、与均值方法对比的效果、不同密度情况下的表现以及是否能不同特点的数据集选择合适的方法。

### Subclass-of关系补全评估实验详细设计

本文提出的评估算法是与被评估方法无关的，不仅适用于instance-of关系补全方法的评估，也适用于subclass-of关系补全评估。本节将从实验目的、输入、输出和预期实验结果几个方面来详细描述Subclass-of关系补全评估实验的设计。

实验6用被设计的目的是验证该方法计算得到的subclass-of关系补全的和与真实值的保持的在一定阈值内，实验详细介绍见表5-8所示。

本文利用RMSE来描述、与、之间的误差。如果该误差在一定阈值内，则认为该方法确实能在较大程度提高效率的情况下，较准确的估算每个方法的准确率和召回率。

为了较为客观反映各个方法的在不同数据上的效果，我们分别构建了两组实验，实验7使用的数据的特点是属性关系是缺失，实验8使用的数据特点的区别是上下位关系稀疏。具体方案如表5-9所示。预计的实验结果该方法是可以自动考虑数据特点，为上下位关系稀疏的数据

表5-8 Subclass-of补全准确率和召回率误差计算实验设计表

|  |  |
| --- | --- |
|  | 实验6 |
| 目的 | 验证实验，验证该方法计算出来的准确率和召回率与真实情况基本符合。 |
| 输入 | Gold-standard（GS）+ ，Gold-standard（GS）+ ，Gold-standard（GS）+ ， 所有方法集合，矩阵密度0.6。 |
| 输出 | 通过SVD预测得到的、及其RMSE。 |
| 实验预期结果 | SVD的误差（RMSE）在一个阈值范围之内。 |
| 预估结论 | 说明该方法确实能在较大程度提高效率的情况下，较准确的估算每个方法的准确率和召回率。 |

选择合适它的数据。从而可以说明该方法确实可以为不同特点的数据选择最合适的方法。

表5-9 Subclass-of关系下算法结果正确性实验设计表

|  |  |  |
| --- | --- | --- |
|  | 实验7 | 实验8 |
| 目的 | 对比实验，验证该方法是否能为具有不同特点的数据选择最合适的方法。 | |
| 输入 | Gold-standard（GS）+ 。  Subclass-of关系补全Method集合。 | Gold-standard（GS）+ 。  Subclass-of关系补全Method集合。 |
| 输出 | 最佳方法序号，及每个方法  、和。 | 同左。 |
| 数据特点 | instance-of关系完整，属性关系部分缺失。 | instance-of关系部分缺失，属性关系完整。 |
| 实验预期结果 | 三个方法表现差异不明显 |  |
| 原因 | 在这种情况，三个方法的基本思想都是利用instance-of关系进行关联规则挖掘，从而补全subclass-of关系。 | 的基本思想是能在instance-of关系部分缺失的情况下，利用属性关系补全instance-of关系，然后再进行关联规则挖掘，补全subclass-of关系。 |
| 预估结论 | 说明该方法的确能为具有不同特点的数据选择最合适的方法。 | |

## 实验结果及其分析

本节将分别对instance-of关系补全评估实验和subclass-of关系补全评估实验进行结果展示和分析。

### Instance-of关系补全评估实验结果及其分析

Instance-of关系补全评估实验共有5个，首先分析实验1。实验1的目的是为了验证本文提出的方法计算出来的准确率和召回率与真实情况是否符合。为了保证实验的客观性，实验选择了三个数据集，，，分别计算每个方法对所有类别的补全的准确率和召回率的真实值 和。然后利用SVD预测和，最后计算预测值和真实值之间RMSE。表5-10为密度为0.6时利用SVD补全后的均方根误差（RMSE）表。三个本体的准确率RMSE 的平均值是0.060760946，召回率的RMSE 是0.130730657。而预测召回率的RMSE比准确率的大是由于召回率本身的数据波动较大，而在这种情况其RMSE小于0.2，更加说明该方法预测缺失的准确率和召回率的误差是较小的。实验结果表明，利用该方法能在提高效率的情况下，较准确的预测每个方法的准确率和召回率。

表5-10 密度一定下利用SVD补全instance-of关系的RMSE表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | Precision | Recall | |
|  | 0.057156234 | | 0.154428243 |
|  | 0.053251897 | | 0.134432199 |
|  | 0.072570489 | | 0.114840008 |
| Average（，，） | 0.060760946 | | 0.130730657 |

实验1说明了本文提出的用SVD预测instance-of关系的准确率和召回率的误差在一定范围，因此该方法预测得到的准确率和召回率是基本接近真实值的。由于平均值填充缺失值是一种简单、常见的处理缺失数据的方式[47]。特别是数据变化不大的情况下是非常简单有效的。但目前本体的构建工作都是基于人工或自动抽取，很难保证数据分布均匀，较容易出现数据偏移，在这种情况，SVD预测效果应该要优于简单的平均值填充。因此我们设计实验2来对比平均值填充与用SVD预测在准确率缺失值，从而验证用SVD计算出来的准确率、召回率比用简单的平均值方法评估的结果要更加精准。实验结果如图所示，SVD预测值的RMSE均小于用平均值的RMSE。因此可以得出结论，SVD预测方法结果与均值方法相比更加精准。

图5-1 SVD与均值法在三个本体上的RMSE图

矩阵的密度对SVD预测的效果影响较大[52]，为了描述不同密度下SVD的预测效果，我们设计了实验3，同时均值在不同密度下的表现情况也加入该实验进行对比。如下图所示，x轴表示矩阵密度，y轴表示RMSE。设置的密度分别为[0.3，0.4，0.5，0.6，0.7，0.8]。如图5-2中黑色虚线所示，随着矩阵密度的增加，利用SVD和平均值预测真实值的误差基本的趋势是逐渐降低。其中密度在0.5以下是RMSE较大，且变化趋势趋于平缓。在密度为0.5时存在一个异常数据点，根据分析实验中间结果发现，是由于该缺失的数据真实值较为异常。因此可以得出结论，随着矩阵密度的增加，误差逐渐降低。

图5-2 SVD与均值法在不同矩阵密度下的RMSE图

通过实验3得到结论：随着矩阵密度的增加误差逐渐降低。但对于本体构建者来说，提高矩阵密度虽然提高预测精准度，但计算工作量会增加。因此，本体构建者需要根据自己实际情况选择一个合适的密度。如果需要较高的精准度，则提高矩阵密度。如果需要以较少的工作量得到一个一般的精确度即可，则可以选择一个较低的密度。

实验4和5的目的在于验证不同的数据特点情况下，该方法是否能为具有每个数据选择一个最合适的方法。下面主要分析实验4的实验结果，实验4的待评估数据集的特点是instance-of关系不完备，上下位关系完整，属性关系稀疏，其稀疏程度是实验5数据的0.5倍。其实验结果如图5-3所示，每个方法的准确率，召回率和F1值都用不同颜色表示。根据实验结果，我们观察到 和的准确率、召回率和F1值明显高于。其原因是由于， 和是基于上下位关系，即父子类关系和等价类关系作为特征学习得到的模型，不需要使用属性关系。而method3是通过统计每个属性谓词的主语和宾语属于不同类别的概率作为实例与某个类别是否有instance-of关系的依据，如果属性缺失，统计结果将只能反映部分instance-of关系，导致其结果的召回率较低。在属性缺失的情况下，受噪音数据干扰较大，其准确率也受到影响，明显低于到 和的准确率。

图5-3 基于SVD计算不同方法评估的准确率、召回率和F1值图

实验5中待评估数据集的特点是instance-of关系不完备，属性关系完整，上下位关系稀疏，其稀疏程度是Gold standard数据的0.5倍。其实验结果图5-4所示，每个方法的准确率，召回率和F1值都用不同颜色表示。根据实验结果，我们观察到method3的召回率要明显高于method1和和method2，同时对比图5-3，method1的准确率基本没有变化，而召回率明显下降，这是由于method1的算法特征中的上下位关系能够较为精准的决定一个实例与一个概念是否具有instance-of关系，如“Jone”是“Scientist”，如果有一条上下位关系“Scientist是Person的子类”，那么能推出“Jone”与“Person”具有“instance-of关系”。上下位关系的部分缺失，只会使得召回降低，因为召回率反映的正例被分为正例的覆盖率。即如果没有“Scientist是Person的子类”关系，就会漏掉“Jone”属于“Person”这个正例，从而是召回下降。上下位关系的缺失并不会使得method3的效果下降，因为method3的的算法基本思想是利用属性关系计算实例属于概念的概率。因此在这种情况下，从理论上来说method3最适合。而综合考虑准确率和召回率及F1值，的确是method3最合适。

图5-4基于SVD评估不同方法补全的加权准确率、加权召回率和加权F1值图

通过本文提出的方法，能真实、准确地反映不同方法在实际数据上补全instance-of关系的效果，从而为本体构建者选择instance-of关系补全方法提供依据。不仅节约了时间和人力成本，能在很大程度上提高本体丰富工程的准确性和效率。

### Subclass-of关系补全评估实验结果及其分析

Subclass-of关系补全评估实验共有3个，首先分析实验6。实验6的目的是验证本文提出的方法得到的准确率和召回率与真实情况基本相符。为了保证实验的客观性，实验选择了三个数据集，，，首先分别计算每个方法对所有类别的补全的准确率和召回率的真实值和。然后利用SVD预测和，最后计算预测值和真实值之间RMSE。表5-11为密度为0.6时利用SVD补全后的均方根误差（RMSE）表。三个本体的准确率RMSE的平均值是0.106365014，召回率的RMSE平均值是0.169481078。召回率预测的RMSE比准确率的大是由于召回率本身的数据波动较大，而在这种情况其RMSE小于0.2，说明了该方法预测缺失的准确率和召回率的误差是较小的。因此可以得出结论，利用该方法能提高效率的情况下，较准确的评估每个方法的准确率和召回率。

表5-11 密度一定下利用SVD补全subclass-of关系的RMSE表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | Precision | Recall | |
|  | 0.094093549 | | 0.173270759 |
|  | 0.101718674 | | 0.152965409 |
|  | 0.123282818 | | 0.182207066 |
| Average（，，） | 0.106365014 | | 0.169481078 |

实验7和8的目的在于验证不同的数据特点情况下，该方法是否能为具有每个数据选择一个最合适的方法。下面主要分析实验7的实验结果，实验7的待评估数据集的特点是每个实体instance-of关系完整，属性关系部分缺失，对比实验8中的数据属性缺失一半。其实验结果如如5-5所示，每个方法的加权准确率，加权召回率和加权F值都用不同颜色表示。根据实验结果，我们观察到，，的加权准确率，加权召回率和加权F1值相差不大。其原因是由于这三个方法都是基于type关系做关联规则挖掘，的效果需要在数据集中instance关系缺失，且属性关系丰富的情况下通过补全instance-of关系，然后，而该数据集中属性关系部分缺失，对instance-of关系的补全基本没有效果，所以在这种情况下三者的补全效果并无明显差别，其实验结果也确实没有特别明显的差别，说明本文提出的方法确实能较客观、准确地评估每个方法在特定数据上的表现结果。

图5-5 基于SVD评估不同方法补全的加权准确率、加权召回率和加权F1值图

实验8中待评估数据集的特点是instance-of关系不完备，属性关系完整，其中instance-of关系对比实验7的数据缺失了一半。其实验结果如图5-6所示，每个方法的加权准确率，加权召回率和加权F值都用不同颜色表示。根据实验结果，我们观察到的召回率和准确率要明显高于和，这是由于在缺失instance-of关系的情况下，根据SDType算法和属性数据对instance-of进行了补全，从而提高了结果的加权准确率、加权召回率和加权F1值。而和基于instance-of关系的subclass-of补全方法，在instanc-of关系的大量缺失的情况下错误率和召回率都比较低，而且由于instance-of关系稀少，补全效果容易被噪音数据影响，导致准确率较低。

图5-6 基于SVD评估不同方法补全的加权准确率、加权召回率和加权F1值图

## 本章小结

本章针对第四章提出的基于矩阵分解的RDF数据的is-a关系补全方法的评估算法，设计了一系列对比和验证实验。实验结果表明以下几点：首先该方法计算得出的准确率和召回率基本与真实值误差较小，同时该方法与均值方法相比结果更加精确。并且在不同密度情况下该方法也要优于简单平均值方法的效果。最后的四个实验表明了该方法确实能为不同数据选择效果最好的方法，不仅提高了评估流程的效率，而且具备较高精度。

第六章将对本文的工作进行总结，并展望将来的研究工作。

1. 总结与展望

本章对本文工作进行概括与总结，并对未来研究工作进行总结与展望。

## 工作总结

近年来，RDF数据应用于很多领域，如数据集成、药物信息、生物医疗等领域，特别是在语义web中有广泛的实际应用。RDF数据中的is-a关系是这些应用中一些关键技术的基础，如：推理、一致性检测等。但由于各种原因，如导致大规模开放式连接知识库is-a关系不完整情况普遍存在，同时还存在噪音和不一致性数据。如部分知识库基本只有概念层信息组成，相反部分知识库很大程度基本只由实例层数据组成，缺乏清晰的结构。为了解决RDF数据中is-a关系的不完整问题，越来越多的研究者开始关注RDF中is-a关系的丰富补全问工作。

通过丰富概念层的概念与概念之间的隶属关系、实例和概念之间的属于关系，为更加强大的推理，一致性检查和改进查询结果提供了有力的支持。众多学者中提出了一些is-a关系补全方法，得到了不错的处理效果。但是由于各个方法各有其使用的场景，工作者不仅需要耗费大量的人力和时间分析各个方法的使用场景，还需分析其自身数据特点，才能挑选出一个最合适的其数据特点的方法。这个过程不仅工作量大、难以自动化、而且无法排除人为因素影响，做到精准选择。因此研究一个自动化的精确评估方法，帮助用户选择一个适合其数据的is-a关系补全方法，成为了一个比较重要的研究问题。

本文通过对RDF中is-a关系补全算法的分析，提出了一种基于矩阵分解的is-a关系补全方法评估算法，该算法一定程度上解决了上述问题。主要工作如下：

1. 提出了三个测度评估RDF中is-a关系评估：加权准确率，加权召回率和加权F值。加权准确率描述的是方法在整个本体补全的准确率，加权召回率描述的是方法在整个本体补全正例的覆盖程度，加权F值是对加权准确率和加权召回率的综合考虑。通过这三个测度方便用户从补全的准确度，覆盖率上综合评估方法的优劣，从而帮助用户和知识库构建者选择一个适合其数据特点、达到其应用指标的补全算法。
2. 提出了一个自动化、精准化的is-a关系补全方法评估算法，该方法通过采样的方式使不同的方法得到少量的结果，然后基于矩阵分解预测剩余部分的结果，进而得到每个方法在整个数据集下的完整表现，解决了RDF中数据量大带来的评估效率低下的问题。
3. 本文通过大量的实验对比评估了6个is-a关系补全方法在不同数据上的补全效果，验证了该方法确实能高效、精确、准确地评估is-a关系补全算法的效果，且该方法得到的结果与真实值的误差较小，与均值方法相对，效果更优。最后实验还分析不同矩阵密度下该方法的效果，实验结果表明，密度越大该方法的预测误差越小。

## 未来展望

本文在利用矩阵分解评估RDF数据中is-a关系算法补全效果，主要解决了多个is-a算法下如何高效精准的选择一个效果最佳算法的问题，进一步的工作可以在以下一个方面展开：

（1）本文仅分别评估了RDF中instance-of和subclass-of补全算法，在此基础上，可以完善RDF中其它关系补全方法评估。

（2）本文中的工作的矩阵分解算法使用的SVD，而矩阵分解算法还有基于概率的矩阵分解算法（Probabilistic Matrix Factorization，PMF）[53]等等，可以进一步对比不同矩阵分解算法对RDF数据的is-a关系补全评估效果。

（3）本文主要是将矩阵分解算法应用在RDF数据的is-a关系补全评估中，可以尝试将该方法应用到实体连接工作中。

# 致谢

时光荏苒如白驹过隙，东大已经陪伴我走过了两年半的美好时光，在研究生的学习过程中，最需要感谢的是我的导师漆桂林教授。您治学严谨，为人谦和。在苏州任课的时候开始便常常给予我们指导，即使人不在苏州也通过邮件跟我们交流，我们有时候晚上十点多发邮件，您还会回邮件指导我们，这里边是您对我们殷切的关心和为人师的强烈责任感。到再后来的我们回到南京的实验室学习和科研，更加感受到了您对学术严谨态度的和对学生的关怀。在您的交流中，学到不仅仅是知识，更多的是一种进取精神，感染我们不断努力，我想对漆老师致以最崇高的敬意和衷心的感谢！

另外还要感谢知识科学与工程实验室的全体同学，非常有爱的实验室，做项目积极认真，平日里相互逗乐关心，有时也会敞开胸怀地交流谈心聊前程与烦恼，这些美好的回忆，都深深地印在我的脑海里。一起走过得日子，历历在目，与你们相识，青春无悔。特别要感谢实验室的高桓博士，无论是在生活上还是学习中，他都给予我莫大的鼓励和帮助。

还要十分感谢东南大学-蒙纳什大学联合研究生院，尤其是所有为之诞生为努力的老师们。感谢Ben，Fred，Grace，Hsing等等从澳洲远道而来的老师为我们带来了大洋彼岸的教育理念和教学方法，你们严谨的态度和敬业的精神让我获益匪浅。

最崇高的敬意我要献给我的父母。正是你们无私地支持和帮助，我才能敢于面对学习和生活中的种种困难与挫折，才能昂首阔步地走到今天。你们既是我坚强的后盾，也是我前进的动力。

最后感谢评阅论文的各位专家,对你们付出的辛勤劳动表示诚挚的谢意！

千言万语道不尽自己的不舍和感激，谨以此文献给那些关心我学习、生活的每一位老师、同学和家人，并向他们表达我衷心的感谢。

# 参考文献

<http://www.wtoutiao.com/p/13dlBdu.html>

[2] Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Upsala Journal of Medical Sciences,82*(2), 134-137.

[3] Kisilevich, S., Keim, D., & Rokach, L. (2013). A gis-based decision support system for hotel room rate estimation and temporal price prediction: the hotel brokers' context. *Decision Support Systems,* *54*(2), 1119-1133.

[4] Sirmans, G. S., Macpherson, D. A., & Zietz, E. N. (2005). The composition of hedonic pricing

models. *Journal of Real Estate Literature,13*(1), 3-43.

[5] Bull, A. O. (1994). Pricing a motel’s location. *International Journal of Contemporary Hospitality Management,* *6*(6), 10-15.

[6] Monty, B., & Skidmore, M. (2003). Hedonic pricing and willingness to pay for bed and breakfast amenities in southeast wisconsin. *Journal of Travel Research,* *42*(2), 195-199.

[7] Israeli, A. A. (2002). Star rating and corporate affiliation: their influence on room price and performance of hotels in israel. *International Journal of Hospitality Management,* *21*(4), 405-424.

[8] Thrane, C. (2007). Examining the determinants of room rates for hotels in capital cities: the oslo experience. *Journal of Revenue & Pricing Management,* *5*(5), 315-323.

[9 Hung, W. T., Shang, J. K., & Wang, F. C. (2010). Pricing determinants in the hotel industry: quantile regression analysis. *International Journal of Hospitality Management,* *29*(3), 378-384.]

[10] Lee, S. K., & Jang, S. C. (2010). Room rates of u.s. airport hotels: examining the dual effects of proximities. *Journal of Travel Research,49*(3), 186-197.

[11] Etzioni, Oren, Tuchinda, Rattapoom, Knoblock, Craig A., & Yates, Alexander. (2003). To buy or not to buy: mining airfare data to minimize ticket purchase price. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, Dc, Usa, August*(pp.119--128).

[12] Groves, W., & Gini, M. (2013). Optimal Airline Ticket Purchasing Using Automated User-Guided Feature Selection. *IJCAI '13: Proceedings of the, International Joint Conference on Artificial Intelligence*.

[13] Popescu, A. M., & Etzioni, O. (2007). Extracting Product Features and Opinions from Reviews. Natural

Language Processing and Text Mining. Springer London.

[14] Frechtling, D. C. (2001). Forecasting tourism demand: methods and strategies.*Forecasting Tourism Demand Methods & Strategies*.

[15] Gang, L., Song, H. Y., & Witt, S. F. (2005). Recent developments in econometric modeling and forecasting. *Journal of Travel Research,44*(1), 82-99.

[16] Haiyan Song, & Gang Li. (2008). Tourism demand modelling and forecasting—a review of recent research. *Tourism Management,* *29*, 203–220.

[17] Bing, P., Chenguang [Wu, C. G. D. W., & Song, H. Y. (2012). Forecasting hotel room demand using search engine data. *Journal of Hospitality and Tourism Technology,* *3*(3), 196-210.

[18] Choi, H., & Varian, H. (2012). Predicting the present with google trends.*Economic Record,88*(Supplement), 2–9.

[19] 周志华. (2016). *机器学习 : = Machine learning*. 清华大学出版社.

[20] 李航. (2012). *统计学习方法*. 清华大学出版社.

[21] Quinlan, J. R. (1986). Induction of decision trees" machine learning. *in Data: Goals and General Description of the IN L.EN System." in*, 257--264.

[22] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

[23] Breiman, L. I., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees (cart). *,* *40*(3), 358.

[24] Chen, T., & Guestrin, C. (2016). Xgboost: a scalable tree boosting system. 785-794.

[25] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition*. 世界图书出版公司.

[26] Breiman, L. (1996). Stacked regressions. *Machine Learning,* *24*(1), 49-64.

[27] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks,* *5*(2), 241-259.

作者简介

何彬彬（1991—），女，汉族，湖南长沙人，现为东南大学知识知识科学与工程实验室硕士研究生，研究方向为语义web。

* 攻读硕士学位期间发表的论文

1. **何彬彬**. RDF中Is-a关系的质量评估[J]. 东南大学校庆论文集, 2015(5).