题 目：基于城市数据的发展预测系统

大学生创新大赛

基于城市数据的发展预测系统

Development Prediction System Based on Urban Data

学　院：　软件学院

专　业：　软件工程

学　号：　20301138

指导教师：　车啸平

北京交通大学

中文摘要

摘要：

随着城市发展和物联网技术的普及，城市规划面临着前所未有的挑战和机遇。海量的数据为城市规划提供了丰富的信息来源，但同时也带来了数据处理和分析的困难。本文针对欠发达地区由于数据量较少难以利用大数据进行城市规划的问题，提出了一个基于跨城市数据的城市基础设施规划解决框架。

该框架包括五个主要步骤：网格化城市数据、界定城市边界、特征选择、比较城市相似度以及模型构建。在界定城市边界的步骤中，我们提出了基于 POI(Point of Interest)数量的城市网格判定方法。在特征选择步骤中，我们结合了 Fisher Score 和 MRMR(Minimum Redundancy Maximum Relevance)两种常用的特征选择算法。在计算城市相似度的步骤中，我们综合考量了 POI 数量特征以及 POI 分布结构，创新性地使用 SSIM(Structure Similarity Index Measure)比较城市间的 POI 分布结构。在模型构建的步骤中，我们提出了 MSC(MultiSource Collaborative) Tradaboost 模型，充分地结合多个源域数据对目标域上的任务进行处理。

通过实验，我们验证了该框架自身的有效性以及相较于其他方法的优越性。尽管本文取得了一定的成果，但仍存在一些问题和挑战，如城市边界的精准度量、多维度城市数据的融合以及基础设施自身特性的考虑。未来，我们希望对这些问题进行深入研究，为城市规划提供更全面的解决方案。

关键词：城市计算；POI；城市规划；迁移学习；

ABSTRACT

ABSTRACT：

As urbanization accelerates and the Internet of Things (IoT) technology becomes widespread, urban planning is facing unprecedented challenges and new opportunities. Massive data provides a wealth of information sources for urban planning, but also brings difficulties in data processing and analysis. This paper addresses the issue of insufficient data in underdeveloped areas, which hinders the utilization of big data for urban planning, by proposing a cross-city data-based urban infrastructure planning framework.

The framework consists of five main steps: gridding city data, defining city boundaries, feature selection, comparing city similarity, and model construction. In the step of defining city boundaries, we propose a city grid determination method based on the number of Points of Interest (POI). In the feature selection step, we combine the Fisher Score and Minimum Redundancy Maximum Relevance (MRMR) algorithms, two commonly used feature selection methods. In the step of calculating city similarity, we innovatively use the Structural Similarity Index Measure (SSIM) to compare the POI distribution structure between cities, taking into account both the quantity and distribution of POIs. In the model construction step, we propose the MultiSource Collaborative (MSC) Tradaboost model, which effectively combines data from multiple source domains to address tasks in the target domain.

Through experiments, we validate the effectiveness of the framework and its superiority compared to other methods. Despite the achievements of this study, some issues and challenges remain, such as the accurate measurement of city boundaries, the integration of multi-dimensional city data, and the consideration of the inherent characteristics of urban infrastructure. In the future, we hope to conduct in-depth research on these issues and provide a more comprehensive solution for urban planning.

KEYWORDS： Urban Computing; POI; Urban Planning; Transfer Learning;

目 录

1 引言

随着城市人口和互联网技术的普及，城市中产生的数据量也在以指数级的速度增长。通过合理分析城市数据，城市管理者不仅可以更方便地了解城市的运行状况、市民需求，以此来优化资源分配，推动智能城市发展[1]，还可以了解城市在各个维度的发展实况，并对城市未来趋势进行规划。

本章介绍了基于城市数据对城市发展进行预测的研究背景和意义，具体从传统层面、大数据层面进行分析，同时阐述了本文的主要研究内容和研究思路，最后介绍了本文的行文结构。

1.1 研究背景及意义

随着城市化的不断推进，我国提出发展建设智慧城市的目标，旨在通过数字化、信息化和智能化手段，提高城市的可持续性、宜居性和效率[2]。与此同时，一些新生城市正处于快速扩展的过程：人口大量涌入，城市面积的扩张。而随之而来的便是交通肿胀、紧急事件增多、资源消耗量增大、公共服务压力增加[3,4,5,6]等一系列问题。提前进行城市规划，合理利用城市地区的资源、空间等，可以尽量减轻这些问题，实现可持续的城市发展[7]。目前，随着大数据技术的发展，大数据技术以其强大的分析能力为城市规划者提供了工具、方法和参考依据[8]。大量研究和实践也证明，大数据技术是我国智慧城市的建设的关键核心。

Yu Zheng 提出了城市计算[9]这一概念，旨在研究如何利用大数据、物联网、人工智能等技术来解决城市问题，并提出了解决城市问题的一个基本框架，主要包括以下这四个步骤：

（1） 数据采集：城市计算的第一步是收集大量的城市数据。这些数据可以来自多种来源，如传感器、社交媒体、公共交通、卫星遥感等。这些数据反映了城市的各个方面，包括人口、交通、环境、能源、健康等。

（2） 数据处理：收集到的城市数据通常是异构的、不完整的和嘈杂的。因此，数据处理是城市计算的关键环节。数据处理包括数据清洗、数据融合和数据建模等步骤。数据清洗是去除数据中的噪声和异常值；数据融合是将来自不同来源的数据整合到一起；数据建模是将数据转换为可供分析和挖掘的形式。

（3） 数据分析：在数据处理之后，接下来是对城市数据进行分析。数据分析的

目的是挖掘数据中的隐藏规律和趋势，为城市管理者提供有价值的信息。数据分析可以采用多种技术和方法，如统计学、机器学习、数据挖掘等。数据分析可以分为描述性分析、预测性分析和推荐性分析。描述性分析是对过去的数据进行总结和概括；预测性分析是预测未来的需求、风险和机遇；推荐性分析是为决策者提供优化建议和调度方案。

（4） 服务与应用：城市计算的最终目标是为智慧城市提供服务和应用。基于城市数据分析的结果，可以开发各种创新的服务和应用，以提高市民的生活质量和满意度。例如，基于位置信息和用户兴趣的推荐系统可以帮助市民发现周边的餐厅、活动和景点[10]；智能家居系统可以实现家庭设备的远程监控和控制；基于数据分析的交通管理系统可以减少拥堵和提高道路使用效率。

以下是具体架构示意图。

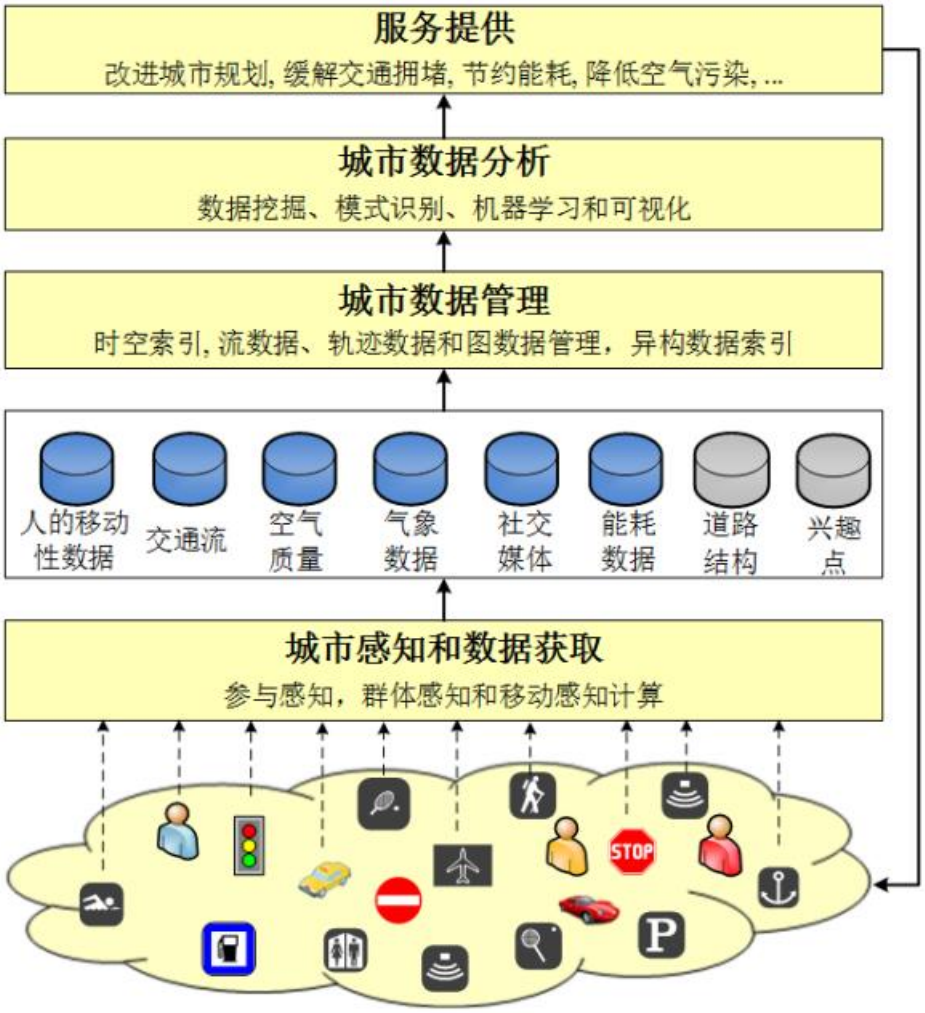

图 1-1 城市计算架构图

然而，各个城市之间的发展水平较不均衡。如北京、上海、广州、深圳等城市发展水平较高，其大量的人口以及智能设备的高普及率以及公共交通的发达程度足以为城市计算提供大量的数据支持。而大部分三四线城市，因其发展较晚，智能化水平较

为底下，产生的数据量较少，训练出的模型常常陷入欠拟合的局面[11]，不利于城市计算的开展，因此难以为城市管理者提供服务。但新生城市又正处于飞速发展阶段，提前进行城市规划以避免将来发生众多问题极为重要。本文将对如何在欠发达地区进行城市规划进行研究，以选址问题为例，提出了一套用于解决欠发达城市规划问题的解决框架。

1.2 国内外研究现状

本小节主要介绍了城市规划问题中各个粒度的研究现状，从传统层面的城市规划，再到结合大数据技术的城市规划再到如何使用大数据在欠发达城市进行城市规划。并研讨了跨城市数据分析技术的研究现状。

1.2.1 城市规划研究现状

城市规划涉及到多个方面包括城市发展战略、土地利用与交通规划、环境与生态保护等。Batty, M[12]探讨了城市的规模、尺度和形状对城市规划的影响，并指出，随着城市化的进程，越来越多的人口聚集在城市中，这使得城市规划面临巨大的挑战。Bettencourt, L. M. A., & West, G. B.等人[13]提出了一个统一的城市生活理论，旨在解释城市中的社会、经济和基础设施之间的相互关系。作者认为，城市是一个复杂的系统，需要在多个层次上进行研究。Hall, P.[14] 回顾了自 1880 年以来城市规划和设计的发展历程，并介绍了多种城市规划理论和实践，如花园城市、新城镇、城市更新等。这些理论和实践为当今城市规划提供了丰富的经验和启示。Wu, F.[15]专门研究了中国的城市和区域规划。分析了中国城市规划的特点和挑战，如快速城市化、土地供应与房地产市场、社会空间不平等等。然而，这些研究都只是在传统层面研究城市规划，如果结合大数据技术，可以提高城市规划的效果和效率。

Batty, M., Axhausen, K. W.,[7]等人讨论了智能城市的未来，重点关注了大数据技术如何用于城市规划。作者提出了一种基于大数据的城市建模方法，可以帮助城市规划者更好地理解城市运行机制，为城市发展提供有益指导。论文[16]提出了一种基于机器学习的方法，用于大规模评估城市环境质量。作者利用大数据技术收集和分析城市环境数据，为城市规划提供有益的参考信息。论文[9]系统地介绍了城市计算的概念、方法和应用。城市计算是一种将大数据技术应用于城市规划的方法，涉及交通管理、环境保护、社会服务等多个方面。文章详细介绍了如何利用大数据技术支持城市规划的决策过程。MNI Sarker, Y. Peng.等人[17]探讨了如何利用大数据技术提高城市的危机应对能力和抗灾能力。作者揭示了一些可以在灾害管理的不同阶段和提高韧性方面轻松使

用的重要大数据技术，如遥感图像、社交媒体数据、众包数据、地理信息系统（GIS）和移动元数据，可以帮助城市规划者更有效地应对自然灾害和人为灾害。这些文章系统地研究了如何结合大数据技术来解决城市规划、城市管理。但大多数研究都是利用“本地”数据解决“本地”问题。在一些城市化较晚、数据较少的城市，这样的解决方法的框架可能存在一些问题，如某一维度的数据因传感器未安装而难以训练模型；数据量太小导致训练的模型欠拟合等。

目前，为解决上述问题，主要有以下几个策略。

1.利用跨城市的数据：尽管某个城市的数据可能较少，但可以通过整合多个城市的数据来提高数据量。这样做的一个前提是这些城市之间存在某种程度的相似性，从而使得跨城市的数据能够为研究提供有价值的信息。

2. 利用时间序列数据[18]：尽管某个时间点的数据可能较少，但可以通过收集一段时间内的数据来提高数据量。这样做的一个前提是这段时间内的数据能够反映出城市发展的趋势和特征。

3.使用其他类型的数据[19]：除了传统的城市数据（如人口、交通、土地利用等），还可以尝试利用新兴的数据来源（如社交媒体、移动设备、遥感等），以提高数据量和数据质量。

而相比于其他方法，利用跨城市的数据进行分析有以下几点优点：

1.增加数据量：整合多个城市的数据可以显著提高数据量，从而使得分析结果更为可靠和稳定。这对于基于数据驱动的城市规划方法尤为重要。

2.发现普遍规律：通过比较不同城市之间的数据，研究者可以发现一些普遍适用的城市发展规律。这有助于理解城市之间的相似性和差异性，为制定适用于多个城市的政策提供依据。

3.借鉴他城经验：通过对比多个城市的数据，可以发现一些在其他城市中取得成功的城市规划实践，从而为本地城市的规划提供借鉴和参考。

4.促进跨地区合作：整合跨城市的数据可以促进城市间的合作和交流，为城市规划的研究和实践创造更多的合作机会。

5.提高模型的泛化能力：通过在多个城市的数据上训练和验证模型，可以提高模型的泛化能力，使其在新的城市和场景下具有更好的预测和分析性能。

6.丰富研究视角：利用跨城市的数据可以为研究者提供多元化的研究视角，有助于发现城市规划问题中的新颖和有趣的现象。

因此，我们认为：可以结合跨城市数据，将发达城市的海量数据，用于解决欠发达地区城市规划问题，从而有效减轻城市快速发展带来的一系列问题，推进智能城市的建设。

1.2.2 跨城市数据分析研究现状

跨城市数据用于城市规划主要涉及城市发展、交通规划、土地利用、公共服务设施等方面。Bettencourt, L. M., Lobo, J.[13]等人研究了城市间的数据关系，揭示了城市规模、创新和生活节奏之间的一般规律。Barthelemy, M.[20] 总结了空间网络分析的基本方法和应用，为城市规划问题提供了一种适应性的分析框架。Bettencourt, L. M.[21] 分析了城市公共服务设施规模的起源，并揭示了城市规模和公共服务设施之间的关系。通过这些论文，我们可以看到城市规划方面的跨城市数据研究已经取得了一定的成果。这些研究为城市规划的决策制定提供了有力的支持，同时也为未来城市规划的研究提供了新的方向和思路。

而跨城市数据在研究过程中也采用了多项技术，大部分都包括在城市计算所采用的技术中，涉及数据挖掘、空间分析、网络分析等。Zhang, J., Zheng, Y., & Qi, D.[22] 采用了深度学习技术来预测城市范围内的人流，提高了跨城市数据分析的精度和效率。Barthelemy, M.[20]这篇论文介绍了空间网络分析的基本方法和应用，为城市间的交通、基础设施等问题提供了分析工具。Yuan, Y., Raubal, M., & Liu, Y.[18]通过分析一段时间内的手机使用数据来研究城市居民的出行行为，运用时空数据挖掘技术揭示出行模式。Li, X., Peng, L.等人 [23]采用迁移学习方法将一个城市的空气质量预测模型应用于其他城市，提高了空气质量预测的准确性。通过这些技术，跨城市数据分析得以在不同领域如交通、环境、社会经济等方面取得显著的成果。本文也借鉴了上述的部分技术对跨城市规划问题进行分析，同时就基础设施分布问题提出了一套有效的解决架构。

1.3 论文主要内容

通过上述研究背景我们可以看出，利用跨城市数据可以解决欠发达地区数据量较少的问题，帮助城市管理者进行城市规划，从而减轻城市快速发展将带来的一系列问题，实现可持续发展。同时，如何更好地将其他城市数据迁移至目标城市再进行分析也引起了学术界的持续关注，具有研究价值。本节对研究内容以及研究思路进行了介绍。

1.3.1 研究内容

跨城市数据迁移是城市计算中的重要方法，指的是将一个城市的模型和算法应用于另一个城市以解决相似问题，从而使得欠发达城市克服数据稀缺问题。跨城市数据迁移的成功实现对于推动智慧城市的建设的作用可以体现在以下几个方面：

1.对于城市管理者来说，可以帮助他们更好地理解城市间的相似性和差异性，以及如何利用这些信息来改善城市规划和管理，并有助于形成更加全面和科学的城市发展

策略实现智慧城市的建设。

2.对于城市居民来说，可以获得更加精准的服务，提升了生活的便利性和舒适性。

3.对于城市的发展来说，欠发达城市解决了数据稀缺问题，进而以更加精确的模型指导城市规划，及时减轻甚至规避了发展中可能出现的问题。同时城市之间可以相互学习，共同解决城市发展过程中遇到的问题。

由此可以看出，研究跨城市数据迁移具有巨大的价值，结合上述 1.2 中论文，跨城市数据迁移主要存在以下几个问题。

1.来自城市的数据数量较大、质量参差不齐、结构高度分散，难以搭建模型进行分析。需要一种方法可以使分散的数据具有一定的结构，方便对其进行建模，以及需要对数据进行提纯，减少质量较低的数据，方便后续进行建模。

2.城市的边界难以界定，若未正确界定城市边界，在收集数据时可能收集市外数据，增加数据分析资源消耗，同时降低了模型对城市的拟合程度；亦或是未收集全部城市种的数据，导致模型欠拟合。

3.如何分析城市相似度。在进行数据跨城市迁移时如若考虑以高相似度的城市作为源城市，可以更有效地实现数据迁移，需要一种方式可以定量地分析两个城市之间的相似度。

4.上述论文大都考虑将一个城市的数据迁移至其他城市进行分析，如何才能结合多个城市的数据对目标城市进行分析，从而合理利用更多的数据量得到更加精确的结果。

为了解决上述问题，本文以基础设施的分布为例，研究了一个解决跨城市数据迁移的架构，提高了预测的精确度。

1.3.2 研究思路

针对上述提到的几个问题，本文提出以下几个具体的研究思路：

1.针对城市数据数量庞大，质量参差不齐，本文提出了一种结合 fisher score 和 MRMR(Minimum Redundancy Maximum Relevance)算法的特征选择方法来过滤掉无价值信息，同时保证基本信息不丢失，加快后续数据处理以及模型构建，也降低了模型的复杂性，减少了过拟合的风险。

2.针对城市数据结构高度分散，难以训练。本文借鉴文献[20]思想，将城市网格化按地域划分成若干个大小相同的正方形格子。后续便可以对每个网格进行分析和处理，将城市数据进行了格式化的储存。

3.针对城市边界难以界定问题，本文提出了一种基于 POI(Point Of Interest)数量变化的城市边界测定算法，通过观测 POI 数量变化速度来判断城市边界，从而减少因收集错误数据导致的模型欠拟合问题。

4.针对城市相似度问题，本文提出了一种结合宏观 POI 类型数量占比以及 POI 分布

结构的相似度算法，通过提取城市 POI 分布的深度特征计算两座城市的相似度。

5.针对如何利用多个城市的数据进行迁移问题，本文采用了基于实例的迁移学习方法，构建了 MSC Tradaboost(MultiSource-Collaborative Tradaboost)模型，并通过对比实验证明了它较其他模型有更好的泛化能力。

以下是本文研究思路示意图。

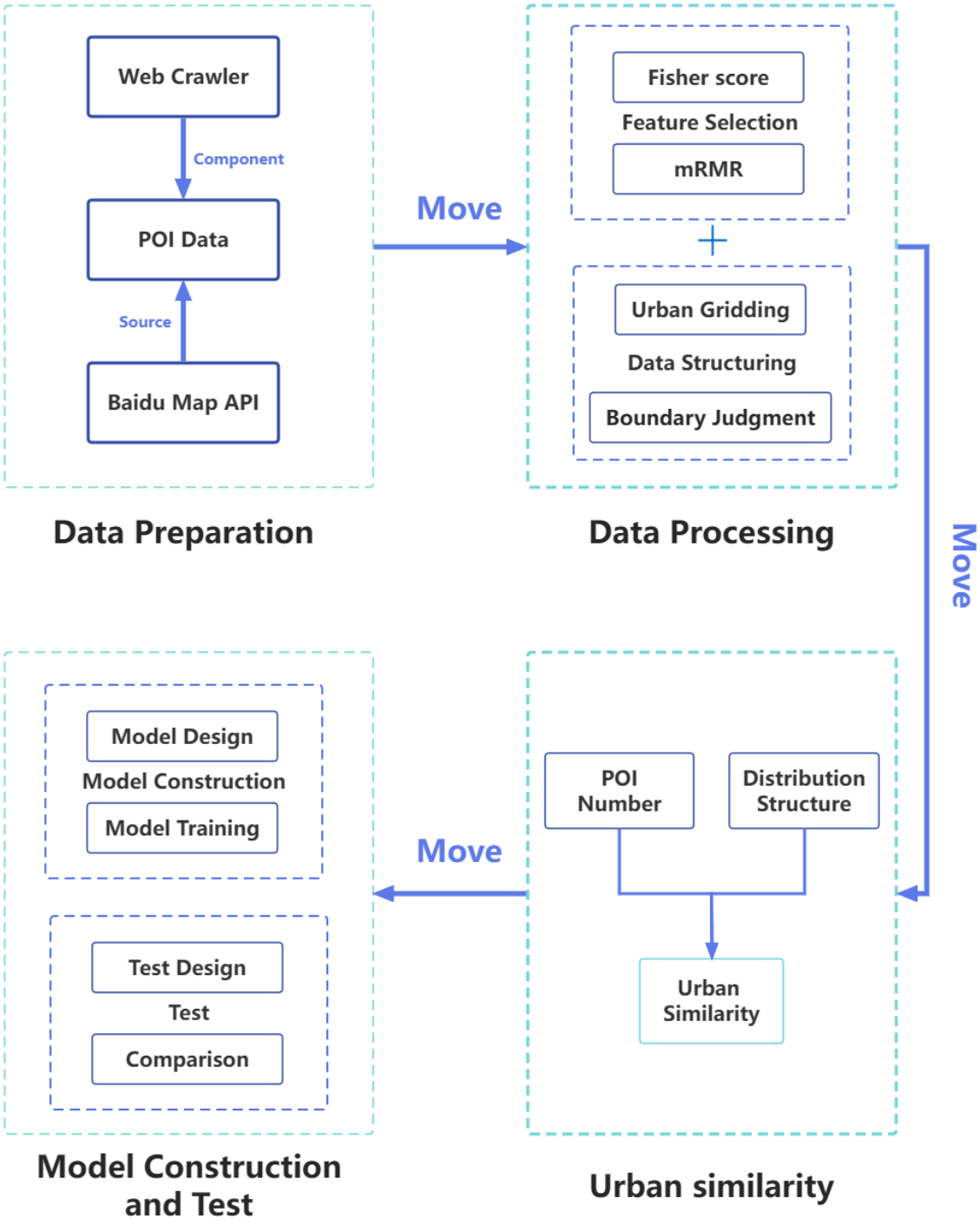

图 1-2 整体研究思路图

1.4 论文结构安排

本论文共分为六章，结构如下：

第一章：引言本章从传统层面和大数据层面介绍城市规划的研究背景，引出欠发达地区数据量较少的问题，这也是本文主要想要解决的问题。接着，介绍研究思路和研究内容。

第二章：相关技术按照解决框架的处理流程，本章依次介绍相关技术，为后续框架的介绍提供理论基础。

第三章：数据获取与处理本章介绍如何获取数据，并说明如何对海量的 POI 数据进行结构化和筛选，为后续的相似度计算和模型构建提供基础。

第四章：城市间相似度度量及数据迁移本章介绍如何度量城市间的相似度，以及如何在高相似度的城市之间进行数据迁移，使模型可以通过多个目标域的数据进行学习并综合解决目标域的问题。

第五章：实验设计与结果分析本章通过设计一个主实验和多个对比实验，详细说明利用解决框架解决问题的流程，并通过对实验结果的分析证明了它的优越性和有效性。

第六章：总结与展望通过总结前五章内容，本章提出论文工作中存在的一些问题以及未来展望。

这是本论文的结构安排，各章节之间紧密联系，层层递进，为解决欠发达地区城市规划问题提供了一个全面的解决方案。

1.5 本章小结

本章阐述了本文的研究背景及意义，并从传统层面、大数据层面介绍了城市规划的研究现状，并以欠发达城市数据稀缺这一现实问题出发，研讨了跨城市数据迁移的研究现状，最后通过分析跨城市数据研究中存在的一系列问题，初步介绍了本文的提出的一系列研究思路，并介绍了本文基本结构。

2 相关理论技术

本章就本文提出的解决框架中涉及到的技术以及理论支撑进行阐述。按照处理流程分别分为数据准备、数据处理、城市相似度计算、迁移学习的相关技术。

2.1 数据准备相关技术

2.1.1 城市兴趣点（POI）

城市兴趣点（POI，Point of Interest）是指在城市地理空间中具有特定功能或者吸引力的地理位置。它在地图上的表现形式为一个点或者是一片区域。它通常具有两层含义：第一层包括各种类型的设施，如商业、娱乐、交通、教育、医疗等场所，而第二层则是该类型底下的具体设施，如医疗这一类别中有专科医院、急救中心、综合医院等详细设施。POI 数据通常用于地图应用、导航系统、位置服务和城市规划等领域。通过分析各区域 POI 数据的数量分布的变化，于城市而言可以更好的了解城市空间的活动模式，而对于个人来说，可以了解城市服务分布情况，为出行导航、生活服务提供推荐[24]。以下是 POI 类型以及其细分分类表（基于百度地图 API 提供的 POI 数据）。

POI 类型	细分类型
美食	咖啡厅、小吃快餐店、酒吧、中餐厅、茶座、外国餐厅、蛋糕甜品店、其他
酒店	快捷酒店、其他、民宿、公寓式酒店、星级酒店、
购物	便利店、商铺、家居建材、超市、家电数码、市场、购物中心、其他、百货商场
生活服务	图文快印店、其他、公共厕所、殡葬服务、通讯营业厅、房产中介机构、公用事业、照相馆、物流公司、维修点、洗衣店、宠物服务、家政服务、彩票销售点、售票处、报刊亭、邮局、步骑行专用道驿站
丽人	美发、美容、美甲、美体、其他
旅游景点	其他、景点、游乐园、风景区、教堂、寺庙、文物古迹、公园、博物馆、植物园、动物园、水族馆、海滨浴场
休闲娱乐	洗浴按摩、KTV、其他、游戏场所、休闲广场、农家院、网吧、电影院、度假村、剧院、歌舞厅

运动健身	体育场馆、健身中心、其他、极限运动场所
教育培训	培训机构、幼儿园、小学、中学、其他、科研机构、高等院校、图书馆、亲子教育、留学中介机构、成人教育、科技馆、特殊教育学校
文化传媒	广播电视、艺术团体、展览馆、其他、美术馆、文化宫、新闻出版
医疗	其他、药店、诊所、疗养院、综合医院、新冠疫苗接种点、医疗器械、急救中心、医疗保健、体检机构、专科医院、发热门诊、疾控中心、核酸检测点、方舱医院
汽车服务	汽车美容、汽车配件、汽车销售、汽车租赁、汽车维修、其他、汽车检测场
交通设施	公交车站、停车场、其他、地铁站、充电站、普通停车位、收费站、路侧停车位、接送点、加油加气站、港口、火车站、长途汽车站、桥、服务区、飞机场、电动自行车充电站
金融	典当行、银行、投资理财、ATM、其他、信用社
房地产	内部楼栋、写字楼、住宅区、宿舍、其他
公司企业	公司、园区、农林园艺、厂矿、其他
政府机构	行政单位、党派团体、各级政府、社会团体、居民委员会、公检法机构、其他、政治教育机构、福利机构、中央机构、涉外机构、民主党派
出入口	停车场出入口、其他、门、车站出口、车站入口、高速公路出口、高速公路入口、机场入口、自行车高速出入口、机场出口、
自然地物	其他、水系、山峰、岛屿

道路	路口、其他、城市主干道、高速公路、县道、国道、城市次干道、乡道、城市快速路
门址	门址点、其他
绿地	高尔夫球场、绿地公园
行政区划	区县级、其他、省级、市级、智能区域
地铁线路	
公交线路	普通日行公交车
行政地标	村庄、乡镇、商圈、其他、区县、地级市、省、省级城市
商圈	
铁路	地铁/轻轨、铁路
其他线要素	隧道、疫情管控区
水系	湖沼

表 2-1 POI 分类表

2.2 数据处理相关技术

2.2.1 城市边界判断

城市边界指的是城市区域与非城市区域之间的分界线。这个边界有时候可以是很明确的，例如河流或其他自然界限，但有时候界定城市边界可能会比较模糊。城市边界的界定对于城市规划、土地使用、资源分配以及人口统计等方面具有重要意义，在本文中，城市边界的界定可以降低城市数据受非城市区域数据“污染”，城市数据缺失这两个方面的问题。城市边界判断主要考虑以下几个因素

1.自然地理特征：如河流、山脉或森林等自然地理特征可能会成为城市的自然边界。

2.人口密度：城市与非城市区域之间的人口密度差异可以作为城市边界的一个参考指标。一般来说，城市区域的人口密度较高，而非城市区域的人口密度较低。Rozenfeld, H. D., Rybski, D.等人[25] 研究了城市的人口密度与城市边界之间的关系，发现人口增长遵循一定的规律，可以用于确定城市边界。

3.建筑密度和土地利用：城市区域的建筑密度和土地利用方式与非城市区域存在明显差异。通过对比这些差异，可以帮助确定城市边界。Xu, Z., Gao, X.[26]基于 POI 与城市空间结构和城市要素空间分布的关联性，提出了一种新的通过 POI 密度分布来判别城市建成区边界的技术方法。

4.行政区划：行政区划边界，如市、县等行政单位，也可以作为城市边界的一种界定方式。然而，这种方式可能不是最准确的，因为行政边界可能不完全符合城市发展的实际情况。

2.2.2 特征选择

特征选择（Feature Selection）是机器学习中预处理数据的一个重要步骤。特征选择的目的是为了筛选出最有用、最相关的特征，以提高模型的性能、减少计算时间、提高模型的可解释性等。特征选择主要有三种方法：

1.过滤方法（Filter methods），直接根据特征与目标变量之间的关联性来对特征进行评分和排序。过滤方法通常较为简单，计算复杂度低，但可能会忽略特征间的相互关系。典型的过滤方法包括：相关系数、卡方检验、互信息等。

2.包裹方法（Wrapper methods），包裹方法通过使用某种机器学习算法的学习性能作为评价标准，通过不断搜索特征子集空间来寻找最优特征子集。包裹方法通常可以找到更好的特征子集，但计算复杂度较高，通常不做考虑。

3.嵌入方法（Embedded methods），在模型训练过程中进行特征选择。这类方法通常可以同时考虑特征之间的关系和特征与目标变量的关联性。

而文献[27]从待处理的数据的结构出发，论述总结了针对不同数据结构的特征选择算法，下图是详细分类：

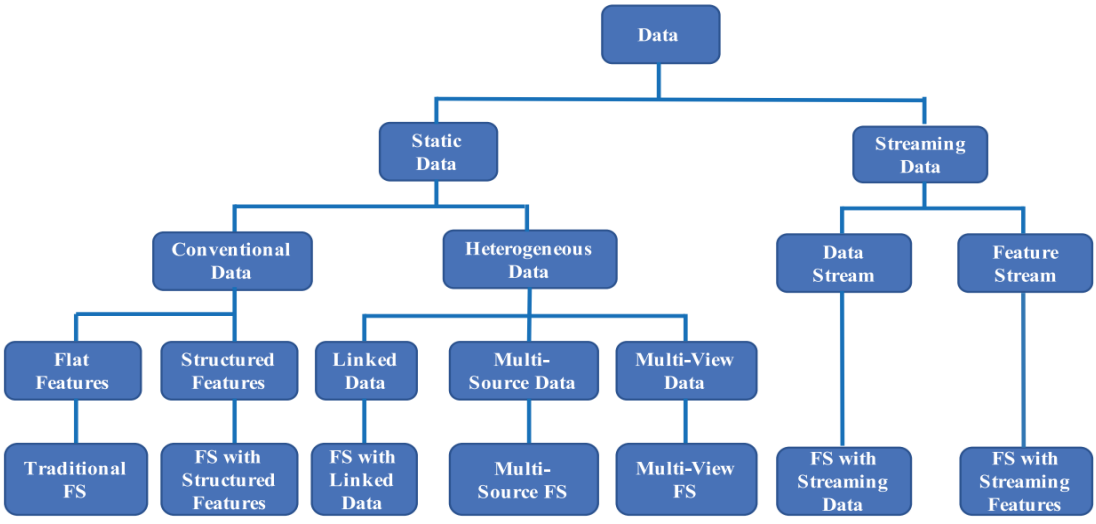

图 2-1 特征选择算法分类图

接下来就本文采用的两种特征选择算法进行介绍。

2.2.2.1 Fisher Score

Fisher Score 是一种特征选择方法。它的目标是在高维数据中找到最佳的特征子集，以便在低维空间中进行分类。

Fisher Score 的核心思想是最大化类间距离（between-class scatter）和最小化类内距离（within-class scatter）[28]。它希望找到那些使不同类别的样本尽可能分开，同时同一类别的样本尽可能靠近的特征。

Fisher 得分的计算公式如下：

$$FisherScore(j) = \left(\left(\mu_1(j) - \mu(j)\right)^2 + \left(\mu_2(j) - \mu(j)\right)^2\right) / \left(\sigma_1^2(j) + \sigma_2^2(j)\right) \quad (2\text{-}1)$$

其中，j 表示第 j 个特征，$\mu_1(j)$和 $\mu_2(j)$分别表示第一个和第二个类别在第 j 个特征上的均值，$\mu(j)$表示所有样本在第 j 个特征上的均值，$\sigma_1^2(j)$和 $\sigma_2^2(j)$分别表示第一个和第二个类别在第 j 个特征上的方差。

2.2.2.2 Minimum Redundancy Maximum Relevance

MRMR（Minimum Redundancy Maximum Relevance）是一种基于互信息（mutual information）的特征选择方法。它于 2005 年由 Peng 等人首次提出[29]，旨在找到一个特征子集，使得子集中的特征与目标变量具有最大的相关性，同时子集内特征之间具有最小的冗余。它的基本公式如下：

$$\varphi(j) = I(Xj; Y) - (1/|S|) \sum_{Xk \in S} I(Xj; Xk) \quad (2\text{-}2)$$

其中$\varphi(j)$是特征Xj的评分，$I(Xj; Y)$表示特征Xj与目标变量Y之间的互信息（即相关性），S是已经选择的特征子集，$I(Xj; Xk)$表示特征Xj与已经选择的特征Xk之间的互信息（即冗余性）。

在每次迭代中，我们选择评分$\varphi(j)$最大的特征Xj添加到已选择特征集合S中。通过这种方式，我们可以确保每次添加的特征与目标变量具有较高的相关性，同时与已选择特征具有较低的冗余性。重复这个过程，直到达到所需的特征数量或满足其他停止条件。

2.3 城市相似度计算相关技术

城市相似度的计算有助于比较城市之间的差异性以及相似性，在本文中，计算城市相似度方便后续源域数据的选择，降低了负迁移的可能性。目前，对于城市相似度的计算方法主要是通过计算城市各特征向量的相似度来进行的。如 Liu, H., & Long, Y.[30] 通过将城市划分为不同的功能区域，然后分析不同区域的功能特征，如住宅区、

商业区和工业区等。利用这些功能区的特征和属性，计算不同城市之间的功能区相似度。Wei, S.[11]通过网格化城市数据，比较宏观的各 POI 类型占比、微观的网格中 POI 的重要程度、网格中 POI 分布特征这三个维度的特征来比较各城市之间的相似度。Daniel, P., Justin, C.等人[31]将不同粒度的区域，如行政区域，整个城市区域表示为其场馆类别的集合，并采用不同的数据标准化方式计算各城市之间的相似度，这种方法可以提供一个有关城市特点和功能的全新视角。

依据其他类型数据进行城市相似度计算是另一种研究方向。Grant, M.[32]提出了一种基于共享微移动模式的方法来衡量城市之间的相似性。作者认为，共享微移动模式可以反映出城市的基础设施、人口密度、交通需求等方面的特征，从而为城市相似性提供一个新的度量标准。

2.3.1 SSIM(Structrue Similarity Index Measure)

SSIM 是一种度量两张图片相似度的方法，由 Wang, Z.等人[33]提出。它主要关注图像的亮度、对比度和结构相似性，能更好地模拟人类视觉对图像处理的过程。其主要计算过程如下所示：

设定局部窗口：首先，选择一个局部窗口（例如 8x8 或 16x16 像素）来逐步遍历整个图像。局部窗口的选择有助于克服图像失真的全局性质，并允许在空间域中进行局部质量评估。

计算亮度相似度：对于每个局部窗口，计算两幅图像的平均亮度（例如，窗口内所有像素值的均值），然后使用亮度相似度公式计算亮度分量。这个公式考虑了两幅图像的平均亮度之间的差异。

计算对比度相似度：接下来，计算两幅图像在局部窗口内的标准差（即对比度）。然后，使用对比度相似度公式计算对比度分量，该公式反映了两幅图像对比度之间的差异。

计算结构相似度：计算两幅图像在局部窗口内的亮度和对比度的相关系数。使用结构相似度公式计算结构分量，该公式反映了两幅图像结构之间的差异。

综合计算 SSIM：将亮度分量、对比度分量和结构分量按权重相乘，然后对整个图像求均值，得到最终的 SSIM 值。 SSIM 的值范围在 -1 到 1 之间，其中 1 表示完全相同，-1 表示完全不同。

本文拟利用 SSIM 可获得图像结构之间差异的特性，计算两个城市之间 POI 分布结构的相似度。

2.4 迁移学习相关技术

迁移学习是一种机器学习方法，它试图利用一个或多个源域（source domain）中的知识来改进在目标域（target domain）中学习任务的性能[34]。源域和目标域可能存在不同的分布或特征空间。在传统的机器学习方法中，训练数据和测试数据通常假设来自相同的分布。然而，在许多实际应用中，这个假设不成立，因为数据可能来自不同的环境或时间段。迁移学习试图解决这个问题，通过在不同的数据分布之间迁移知识，以便在目标任务上获得更好的性能。

2.4.1 迁移学习的分类

按照源领域数据和目标领域数据结构是否相同以及标注情况，迁移学习可以划分为：

1.同构迁移学习[35]，源域和目标域的特征空间相同，但数据分布可能不同。这类方法通常试图在源域和目标域之间找到一个共享的特征表示，使得在源域上学到的知识可以更容易地应用于目标域。

2.异构迁移学习[35]，源域和目标域的特征空间不同。这类方法需要找到一种映射，将源域和目标域的特征空间对齐，以便在不同的空间中迁移知识。

3.无监督迁移学习[36, 37]，在目标域或者源域缺乏标记数据。这类方法通常利用无监督学习技术，如聚类和降维，来发现源域和目标域之间的潜在结构，从而实现知识迁移。

4.有监督迁移学习[38]，在目标域有少量的标记数据。这类方法试图在源域和目标域之间找到一个共享的任务表示，使得在源域上学到的知识可以更容易地应用于目标域。

而按照迁移学习方法所采用的技术划分，又可以把迁移学习方法分为:基于特征选择的迁移学习算法研究;基于特征映射的迁移学习算法研究;基于权重的迁移[39]。

本文主要实现的是 POI 数据的跨城市迁移，即上述的同构迁移学习以及归纳式迁移学习。本文主要采用的迁移技术是基于权重的迁移。

2.4.2 基于权重的迁移

不是所有的源域以及目标域的样本都对最终分类有利，如何放大有利样本对模型的影响，减低不利样本的“污染”，是基于权重的迁移所侧重研究的。Dai, W.等人[40]将 Adaboost 算法思想运用于数据迁移，提出了 Tradaboost，在每次迭代时联合目标域源域

数据训练分类器，基于分类器对目标域的正确率、样本总数对样本进行权重更新，其更新策略是：源域的样本预测误差越大则权重越小，但是对于目标域样本而言，其预测误差越大，则权重越大。最终由多个弱学习器组合，构建出一个强学习器。具体公式如下：

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t \cdot |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t} \quad (2\text{-}3)$$

这是计算基分类器在目标域上的错误率的公式。其中 n 表示的是源域的样本数，m 表示的是目标域的样本数，ϵ_t表示第 t 次迭代时的错误率，w_i^t表示第 t 次迭代第 i 个样本的样本权重，$h_t(x_i)$表示第 t 次迭代训练的基分类器对样本x_i的预测值，$c(x_i)$表示样本x_i的标签值。

$$\beta_t = \epsilon_t/(1-\epsilon_t) \quad (2\text{-}4)$$

β_t表示第 t 次迭代在最终预测时所占的权重，同时也用于更新目标域样本的权重，而ϵ_t由公式 2-3 得出。

$$w_i^{t+1} = \begin{cases} w_i^t \cdot \beta^{|h_t(x_i)-c(x_i)|}, & 1 \le i \le n \\ w_i^t \cdot \beta_t^{-|h_t(x_i)-c(x_i)|}, & n+1 \le i \le n+m \end{cases} \quad (2\text{-}5)$$

这是权重更新的公式，其中β=$1/(1+\sqrt{2\ln n / N})$，N 为总迭代次数。

从上述公式可以看出，Tradaboost 只考虑了单个源域对于目标域的迁移，无法充分利用多个源域数据，且仍有产生负迁移的可能性。

在 Dai, W.等人的基础上，Yi Yao 等人[41]提出了 MultiSource-Tradaboost 以及 TaskTradaboost。其中，MultiSource-Tradaboost 的基本思想为：在每次迭代中，不同源域的数据分别与目标域数据联合训练分类器，横向比较各个分类器在目标域的正确率，并选择最优分类器作为本次迭代的基分类器，而其权重更新策略与 Tradaboost 类似。TaskTradaboost 分为两个阶段，在第一阶段，每个源域数据自身进行 Adaboost，选择出正确率大于阈值的基分类器组成集合进入第二阶段。在第二阶段的每次迭代中，依次得到基分类器集合中各个基分类器在目标域上的正确率，以正确率最高的分类器作为本次迭代的基分类器。其样本权重更新策略也与 Tradaboost 类似。

2.5 本章小结

本章按照解决跨城市数据迁移的流程依次介绍了使用的数据类型，数据处理时涉及到的相关技术，其中包括城市边界判断和特征选择，城市相似度算法以及迁移学习的相关技术，为后续的研究工作提供了理论基础。

3 数据获取和数据分析

本章基于上述理论研究，就如何获取城市 POI 数据、如何处理数据展开论述。在数据获取部分介绍了数据来源、获取工具以及数据获取结果。在数据处理部分介绍了基于 POI 数量的城市边界判断算法、基于 Fisher Score 和 MRMR 的混合特征选择方法。

3.1 数据获取

3.1.1 数据来源和爬取工具

百度地图 API(Application Programming Interface) 是一组由百度地图提供的应用程序接口，为开发者提供了地理信息库的访问途径。其地理数据具有类型齐全、规模庞大、实时更新等特点。据百度地图 API 官方数据：全球 POI 数据已覆盖 1.8 亿、具有超 3000 万地标类 POI。百度地图 API 同时为开发者提供了地图展示、地理编码与逆编码、地图检索、行政区划边界、位置服务等功能。

因为百度地图 API 的数据和功能充分满足本实验中对于完整的城市 POI 数据、地图构建的需求，因此本文采用百度 API 作为本次实验的数据来源。

网络爬虫（Web Crawler），又称为网页蜘蛛、网络机器人，是一种用于自动获取网页内容的程序。爬虫技术主要用于搜索引擎、数据挖掘、数据分析等领域。网络爬虫的主要过程有：

1.抓取（下载）：爬虫从 URL 队列中获取一个 URL，然后通过 HTTP 或 HTTPS 协议请求并下载网页内容。这一过程可能需要处理重定向、处理 cookie 和设置请求头等操作。

2.解析：下载的网页内容通常是 HTML、XML 或 JSON 格式，爬虫需要解析这些内容，提取有用的信息（如链接、文本、图片等）。解析网页内容通常使用正则表达式、XPath、CSS 选择器等方法。

3.存储：提取的有用信息需要存储起来，以便后续分析、处理或展示。存储方式可以是文件、数据库或其他存储系统，如 CSV、JSON、XML、MySQL、MongoDB 等。

4.去重：为避免重复抓取同一网页，爬虫需要检查已访问过的 URL。这可以通过哈希表、布隆过滤器等数据结构实现。

3.1.2 数据获取结果

基于百度地图 API、本文通过网络爬虫共获取全国 15 座城市共计 7548017 条不同的 POI 数据。其中每条记录的字段包括：名称、经度、纬度、所属城市、POI 类型、ID。其中经纬度采用 BD-09 坐标系，POI 类型如表 2-1 所示，ID 为各记录唯一标识。以下是各城市各 POI 类型数量表，数据获取时间：2023 年 3 月 20 日至 2023 年 4 月 1 日。

城市	北京	上海	深圳	广州	合肥
美食	76103	79792	78400	82070	35341
酒店	12779	11868	9937	12501	5397
购物	114376	133946	124903	155701	64956
生活服务	53869	57242	41118	49587	18820
丽人	21397	23997	22239	22444	9883
旅游景点	11658	7230	3920	7411	1949
休闲娱乐	21537	24426	13447	15601	7777
运动健身	10696	10335	6148	6643	2549
教育培训	28810	27570	19867	24133	11452
文化传媒	7364	4843	2718	3599	1298
医疗	14925	13193	13712	15195	6659
汽车服务	13960	17027	14672	20325	9873
交通设施	72211	41174	38748	51549	20531
金融	10179	5164	5587	5444	2623
房地产	96376	92055	67546	49297	27213
公司企业	104908	116386	123467	131361	48941
政府机构	57893	37639	19388	33104	13258
出入口	81064	69848	42925	44903	18026
自然地物	812	773	639	1367	553
道路	23447	16021	10716	23670	4684
门址	3178	7948	1475	597	69
绿地	127	88	74	42	7
行政区划	564	709	1018	868	266
地铁线路	21	25	13	13	2
公交线路	408	292	164	263	91

行政地标	9736	9418	2888	10765	16392
商圈	5	2	2	5	1
铁路	3	3	4	4	0
其他线要素	7	7	7	6	2
水系	0	0	1	0	0
总计	**848413**	**809021**	**665743**	**768468**	**328613**

城市	南京	宁德	厦门	武汉	长沙
美食	45729	11211	34497	52166	48065
酒店	6297	1749	4349	8709	10934
购物	67614	22675	46960	91652	91188
生活服务	24427	4421	14149	30850	23285
丽人	11441	3105	7649	13420	12690
旅游景点	4595	1697	2609	3282	2891
休闲娱乐	10536	1702	4018	12329	11556
运动健身	3450	418	2446	4261	3375
教育培训	13770	2253	8185	17116	14591
文化传媒	1964	294	1259	2164	1814
医疗	6528	2253	4356	11202	10257
汽车服务	8715	1742	5430	13185	12605
交通设施	35360	6114	13036	28954	24538
金融	3311	954	2006	4261	4058
房地产	52117	6430	23282	56126	32428
公司企业	57070	8580	44078	57979	48983
政府机构	17882	6413	7265	24088	16492
出入口	35221	2563	14287	36301	20316
自然地物	1002	1745	392	646	1240
道路	10544	1480	308	7011	4560
门址	952	33	179	1012	83
绿地	15	0	19	14	7
行政区划	436	46	174	392	263
地铁线路	23	0	0	17	6
公交线路	186	143	131	143	173

行政地标	11805	12785	2540	12040	24788
商圈	1	0	1	0	1
铁路	2	0	0	1	1
其他线要素	7	3	4	7	2
水系	0	0	0	0	0
总计	**431000**	**100809**	**246389**	**489328**	**421190**

城市	天津	杭州	西安	成都	菏泽
美食	42682	57121	51827	102797	24057
酒店	4748	10533	11434	16221	1822
购物	75099	97917	85399	155169	59621
生活服务	24974	30421	28591	51445	9290
丽人	11794	16234	13569	24512	7254
旅游景点	2876	6272	3957	5759	732
休闲娱乐	10258	15130	10341	23046	3389
运动健身	3949	4411	3504	6068	1021
教育培训	12639	16238	16501	22436	6544
文化传媒	1562	2987	1980	3620	683
医疗	9672	10438	11467	23246	6661
汽车服务	13186	12298	11749	21817	7151
交通设施	26913	50229	27240	53038	6936
金融	3873	4873	3652	6456	1560
房地产	60403	61842	36847	64233	6530
公司企业	57981	81917	51672	92467	20578
政府机构	23751	28716	15796	32714	9060
出入口	40709	41230	27512	53011	4579
自然地物	442	1957	655	1451	128
道路	5910	12923	6260	16697	1902
门址	323	2538	206	1750	21
绿地	35	11	17	26	0
行政区划	344	515	270	382	138
地铁线路	3	13	3	17	0
公交线路	44	202	56	290	24

行政地标	5345	16752	8357	18524	10035
商圈	5	2	4	3	1
铁路	2	1	1	0	1
其他线要素	1	16	2	1	0
水系	0	0	0	0	0
总计	**439523**	**583737**	**428869**	**797196**	**189718**

表 3-1 各城市 POI 数量表

3.2 数据处理

3.2.1 城市网格化

网格化城市数据，指的是在平面地图中，将连续的城市区域划分成有规律的、固定大小的正方形网格单元的处理方式。这种处理方法可以简化复杂的城市数据结构，便于进行空间分析和统计。在许多研究中，网格化处理已成为处理城市数据的常用方法。

以城市所有 POI 中纬度、经度各自的最大值、最小值为边界，将边界内区域平均划分成 1km*1km 的若干个网格。若两边界之间的距离无法被 1km 均分，则双边同时增加或裁减一定距离以满足网格化条件。

随后在各个网格中统计各 POI 类型的数量作为该网格的特征，从而形成高度结构化的城市 POI 分布矩阵。我们定义，由各个网格构成的、保留原本城市空间结构的城市矩阵为G_m，它的元素为单个网格。同时，为了更方便地分析各个网格的情况，我们将G_m中的每个网格定义为$g_m(i,\ j)$，其中 i 表示网格中心点的经度，j 表示网格中心点的纬度。

3.2.2 界定城市边界

城市是人类社会经济、政治、文化和科技活动的中心，具有高度集中的人口数量。从空间结构方面来讲，城市具有与非城市地区截然不同的建筑分布方式。反映在 POI 数据上，即城市地区的 POI 数量以及 POI 种类会有较大差别。本文从该思想出发，进行城市边界的判定。而为了方便后续实验中对城市网格化处理等操作，求得的城市范围为由南北两条纬度线、东西两条经度线包围而成的矩形范围。因此，求得的城市边界实则为上述四条边界线。

需要明确以下定义，方便后续阐述。

符号	定义
x	经度
y	纬度
L	坐标，形式为$(x,\ y)$
β	表示城市的集合，包括表 3-1 中的所有城市
ρ_m	m 城市中，判断城市边界时使用的数据结构栈
N	某一城市所有的 POI 数量
θ	阈值

表 3-2 符号定义表

第一步：需要根据各 POI 点的坐标值求出城市中 POI 数量最为密集的中心点。我们在网格化的城市数据上，只需找到所有网格中 POI 数量最多的网格，并将它的中心坐标作为城市中心点。我们首先定义$Num^{a}_{g_m(i,j)}$为网格$g_m(i,\ j)$中所有种类为 a 的 POI 数量，则城市中心点如下：

$$\overline{L}_m = (x,y) = argmax(\sum_a Num^{a}_{g_m(x,y)}), m \in \beta \qquad (3-1)$$

其中$\overline{L}_m$表示第 m 个城市中，POI 数量最多的网格的中心坐标，我们后续称该网格为中心网格，将中心网格入栈ρ_m，并且算作城市部分。

第二步：依次判断ρ_m栈顶网格东、南、西、北四个方向的网格中的 POI 数量，若大于阈值θ则入栈，并算作城市部分。

第三步：若ρ_m不为空，重复第二步，直至得到所有城市网格。将城市网格中的纬度、经度各自的最大值最小值依次作为北边界、南边界、东边界、西边界。我们定义x^{e}_{m}、x^{w}_{m}、y^{s}_{m}、x^{n}_{m}分别表示 m 城市四个边界的纬度值和经度值。

以北京为例、求出的边界如下图所示。

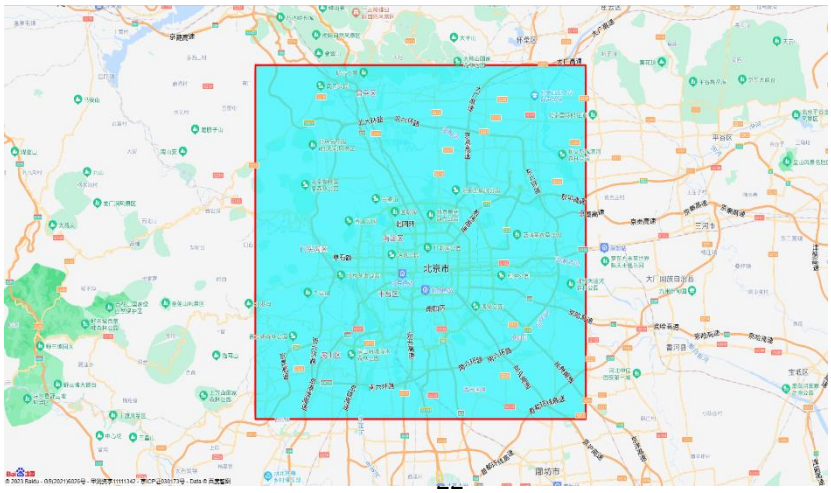

图 3-1 北京城市边界图

可以看到，该区域包括了北京市的主要城区部分，后续研究工作也将在边界范围内进行。

3.2.3 特征选择

Fisher Score 是一种属于 filter 方法的特征选择算法，它的基本思想是：最大化特征之间的类内方差，最小化类间方差，保证相同类别的样本在样本空间中的距离最小，不同类别的样本尽可能分开。越满足上述条件的特征，具有越高的分数。但是它并没有考虑特征之间的相关性，所选出来的特征子集可能具有冗余性。MRMR 算法选择出的特征子集具有与标签具有最大的相关性，同时其特征之间具有最小的冗余性，但 MRMR 算法采用的是贪婪策略更新其特征子集，初始特征的选择十分重要。因此，本文结合 Fisher Score 和 MRMR 算法进行特征选择，其主要步骤如下：

1.根据公式 2-1，依次求出各特征相对于标签的 Fisher Score，定义为 FS(j)。

2.设定阈值θ，将满足 FS(j) < θ的特征 j 删除，不作后续处理。

3.求 j = argmax(FS(j))，以 j 为初始特征加入特征子集。

4.在剩余未被删除的特征中，依次计算$I(Xj;Y)$，求特征与标签之间的相关性。

5.依次计算$(1/|S|)\sum_{Xk\in S} I(Xj;Xk)$，求特征与特征子集中的特征之间的冗余性。

6.选择特征 j 满足 j = argmax($I(Xj;Y)-(1/|S|)\sum_{Xk\in S} I(Xj;Xk)$)加入特征子集。

7.重复 4.5.6 步骤，直至选择出设定好的 n 个特征，输出特征子集。

3.4 本章小结

本章主要介绍了数据收集和数据预处理的过程。首先，在数据收集部分，我们详细讨论了数据来源——百度地图 API，以及用于获取数据的工具——网络爬虫。此外，我们还通过表格形式展示了收集到的数据。

其次，在数据预处理部分，我们按照以下顺序介绍了处理流程：

1.基于 POI 数量变化的城市边界判定算法：通过观察 POI 数量的变化，我们能够确定城市的边界，从而精确地对城市数据进行分析。

2.城市网格化处理：为了对城市数据进行更加细致的分析，我们采用了网格化处理方法，将城市划分为多个网格单元。这种方法有助于揭示城市数据的空间分布特征，并便于后续的分析和建模。

3.Fisher Score + MRMR 算法的特征选择方法：为了从大量的特征中筛选出最有代表性的特征，我们采用了结合 Fisher Score 和 MRMR 算法的特征选择方法。这种方法可

以有效地减少特征数量，同时保留了重要的信息。

总而言之，本章详细介绍了数据收集和预处理的过程，为后续的分析和建模奠定了基础。通过这些处理步骤，我们可以更好地理解和挖掘城市数据中的潜在规律。

4 城市相似度计算和模型搭建

在本章中，我们将详细探讨城市相似度计算和模型搭建的过程。首先，我们将介绍城市相似度计算的算法框架，重点关注特征提取和距离计算这两个关键环节。特征提取部分将解析如何从城市数据中提取有意义的特征，以反映城市的属性和结构。距离计算部分则将描述如何通过这些特征来衡量城市之间的相似性，以便在后续的模型搭建中充分利用这些信息。

接下来，我们将深入探讨模型搭建部分，包括模型的特点、结构、训练和优化方法。在此过程中，我们将阐述模型的核心思想以及它与现有方法的区别和创新之处。同时，我们将详细描述模型的结构，包括各个组件和层次，以便读者能够充分理解模型的构建和运作原理。

最后，我们将介绍模型训练和优化的方法，包括损失函数、优化器的选择，以及模型超参数的调整策略。通过本章的阅读，我们希望读者能够深入了解城市相似度计算和模型搭建的过程，为后续的实验和结果分析奠定坚实的基础。

4.1 城市相似度计算

4.1.1 城市相似度算法架构

城市相似度比较算法架构如图 4-1 所示。从图中可以看到，我们使用 POI 数据来进行分析。主要有数据处理、特征提取、距离计算三个部分，其中数据处理部分已于上文中介绍。以下是具体框架图。

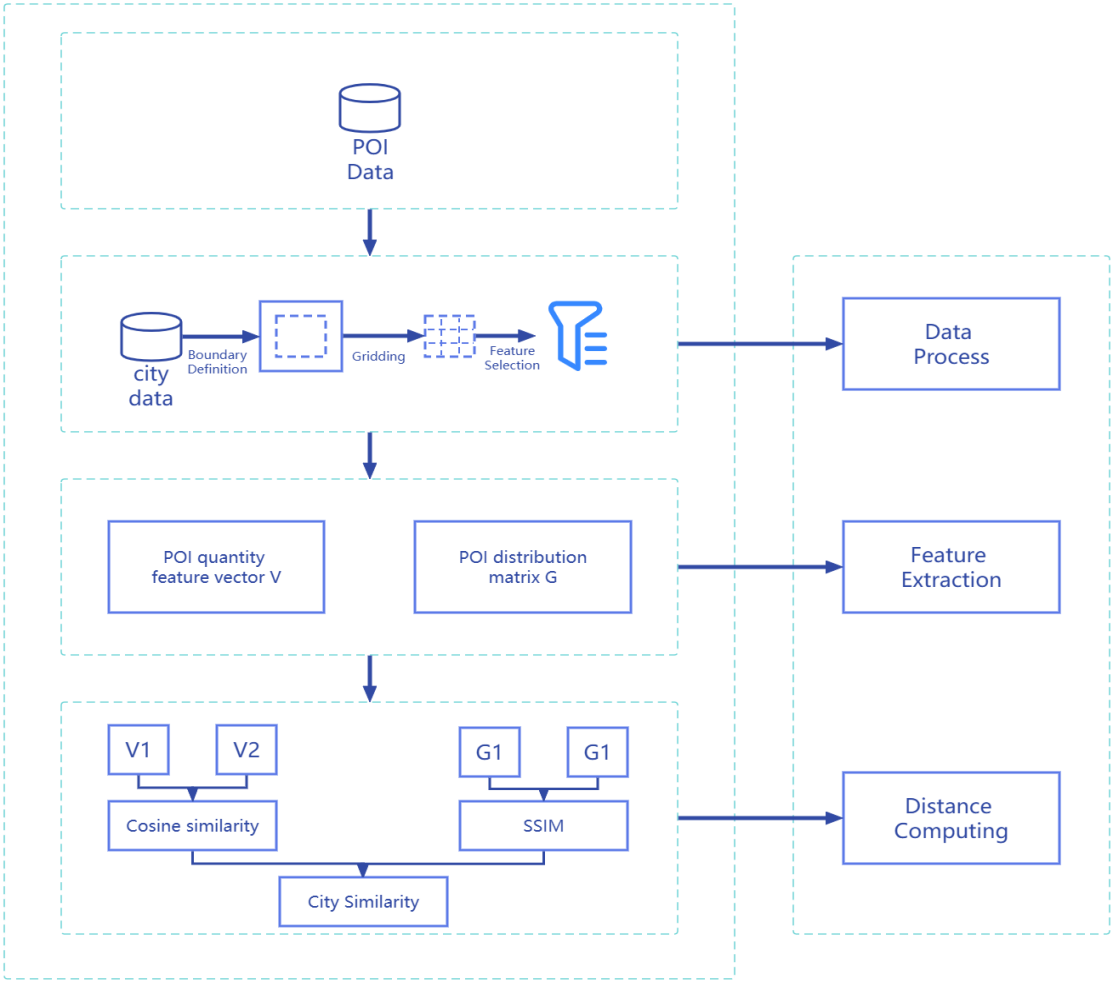

图 4-1 城市相似度计算框架

特征提取部分主要从宏观层面，即整个城市的角度提取两方面特征。第一种是 POI 数量特征，可以分析出城市更深层次的功能结构。第二种是 POI 分布特征，即 POI 分布矩阵，可以分析出城市在空间结构上的分布特征。

距离计算部分基于提取出的 POI 数量特征向量、POI 分布矩阵，使用余弦相似度计算两个城市 POI 数量特征向量之间的相似度，作为这两个城市相似度比较的一部分。接下来，我们计算两个城市 POI 分布矩阵之间的 SSIM 值。我们通过给予这两个相似度合适的权重，最终得到两个城市之间的相似度。

4.1.2 特征提取

本小节基于已处理完成的网格化数据，已完成筛选的 POI 类型，详细介绍特征提取的过程（以下“POI 类型”、“POI 特征”统指已完成筛选的 POI 类型）。

1.POI 数量特征：

我们定义$Num^{a}_{g_m(i,j)}$为网格$g_m(i,\ j)$中所有种类为 a 的 POI 数量。此时，$a \in \gamma$，即已经过筛选的所有特征。我们有如下公式：

$$f_m(a) = \frac{\sum_{g_m \in G_m} Num^{a}_{g_m}}{\sum_{a \in \gamma} \sum_{g_m \in G_m} Num^{a}_{g_m}},\quad m \in \beta \qquad (4\text{-}1)$$

其中$f_m(a)$表示在 m 城市的边界范围内，类型为 a 的 POI 数量占所有 POI 数量的比值。进一步我们有如下公式：

$$V_m = [f_m(a), f_m(b), f_m(c) \ldots\ldots f_m(o)],\quad \{a, b, c, \ldots\ldots o\} = \gamma \qquad (4\text{-}2)$$

V_m表示 m 城市的 POI 数量特征向量，其中包括：经过筛选的特征中，各个特征的 POI 数量占所有 POI 数量的值。

2.POI 分布特征矩阵：

我们在城市网格化这一小节中定义了G_m，G_m指保留有城市空间结构的网格集合。我们定义G^{a}_{m}为城市 m 中，类型为 a 的 POI 分布特征矩阵，G^{a}_{m}与G_m具有相同的矩阵结构，但G^{a}_{m}中的元素并非单个网格$g_m(i,\ j)$，而是$Num^{a}_{g_m(i,j)}$。我们基于这个定义，依次比

较两个城市之间各个 POI 类型的分布特征矩阵，最终得到城市间 POI 分布结构的相似度。

4.1.3 距离计算

1.POI 数量特征计算。

余弦相似度通过衡量两个向量之间的夹角来计算向量之间的相似度，如果夹角越小，则相似度越大，反之越小。它的值在 0~1 之间，若值为 1 则代表两个向量方向相同，若为 0 则表示两个向量方向相反。我们使用余弦相似度计算两个城市 POI 数量特征之间的相似度。

$$SIM\_NUM(m, n) = cos(V_m, V_n) \qquad (4\text{-}3)$$

2.POI 结构矩阵计算。

我们采用 SSIM 来计算两个城市之间的 POI 分布矩阵的相似度，将每个网格视为图像中的一个像素。然而，SSIM 最初是为了评估图像相似度而提出的，因此我们需要对 POI 分布矩阵进行一些预处理，使其能够契合 SSIM 的计算。同时，为了更好地体现不同 POI 类型对整体结构相似度具有不同的影响，我们按照各 POI 类型的重要性为其设置了合适的权重，具体步骤如下：

(1).计算两个城市各 POI 类型的重要性。我们以 m 城市为例：假设G^{a}_{m}有 m 行 n 列，则在 m 城市中，a 类型的重要性由以下公式给出：

$$w^{a}_{m} = \frac{|\{g_m(i,j) | Num^{a}_{g_m(i,j)} \neq 0\}|}{m \cdot n} \qquad (4\text{-}4)$$

即G_m^a中 a 类型 POI 数量不为零的格子占所有格子的数量的比。同样地，计算出w_n^a。以它们的平均数$w^a = (w_m^a + w_n^a)/2$，作为这两个城市中 a 类型 POI 的重要性。依次计算各个 POI 类型的重要性，得到权重向量，再进行归一化得到最终的权重向量w。

(2).对单个网格中的 POI 数值进行缩放。为了将网格视为像素，G_m^a中的元素应满足$0 \le Num_{g_m(i,j)}^a \le 255$。然而，$Num_{g_m(i,j)}^a$存在大于 255 的可能性。因此，需要对$G_m^a$进行缩放。

(3).对G_m^a、G_n^a进行填充。考虑到在比较 SSIM 值时，需要两张图像具有相同的形状，我们采用填充策略来处理这一问题。具体而言，在长度不同的维度上，我们在较小长度矩阵的该维度两侧同时添加 0，直至与另一矩阵在该维度上的长度相同。

以上，我们完成了进行 SSIM 计算前的准备工作。接下来阐述对 SSIM 的计算。

(1).首先，确定窗口大小，通常选择一个较小的窗口，例如 8x8 或 11x11。确保窗口大小适用于我们的 POI 分布矩阵。

(2).计算窗口内的均值、方差和协方差。分别表示为μ_m、μ_n、σ_m、σ_m、σ_{mn}。

(3).使用以下公式计算窗口内的 SSIM 值：

$$SSIM = \frac{(2\mu_m\mu_n + C_1)(2\sigma_{mn} + C_2)}{(\mu_m^2 + \mu_n^2 + C_1)(\sigma_m^2 + \sigma_n^2 + C_2)} \quad (4\text{-}5)$$

其中C_1、C_2为常数，用于避免分母为零时发生计算错误。

(4).在整个 POI 分布矩阵上移动窗口，依次计算每个窗口的 SSIM 值。

(5).最后，将所有 SSIM 值取平均，获得G_m^a、G_n^a之间的 SSIM 值。

重复以上步骤，依次计算不同类型的 POI 分布矩阵之间的 SSIM 值，得到各 POI 类型的结构相似度向量。具体如公式 4-6 所示：

$$SSIM(G_m, G_n) = [SSIM(G_m^a, G_n^a),\ SSIM(G_m^b, G_n^b).....SSIM(G_m^o, G_n^o)], \{a, b, ...o\} = \gamma \quad (4\text{-}6)$$

进一步，获得两个城市间的 POI 结构相似度。

$$SIM\_STRU(m,n) = SSIM(G_m, G_n) \cdot w^T \quad (4\text{-}7)$$

最后，将$SIM_STRU(m,n)$和$SIM_NUM(m,n) = cos(V_m, V_n)$分别乘以合适的权重相加，得到 m、n 两座城市的相似度。

$$SIM(m,n) = SIM\_NUM(m,n) \cdot w_1 + SIM\_STRU(m,n) \cdot w_2 \quad (4\text{-}8)$$

其中w_1和w_2表示两个相似度各自的权重，它们的和为 1。

4.2 基于权重的多源迁移模型

在经过数据预处理、城市相似度计算后，我们接下来详细介绍用于跨城市数据迁移的模型。

在当前的研究背景下，多源迁移学习已经成为解决不同数据源间分布差异问题的有效方法。为了进一步提高多源迁移学习的性能，我们提出了一种名为MSC(MultiSource-Collaborative)Tradaboost 的模型。本节主要包括三个部分：模型特点、模型结构、训练和优化方法。

4.2.1 模型特点

针对如何减小“不良”样本对于模型的“污染”，Dai 等人[40]提出了基于权重的迁移算法 Tradaboost，它的主要思想是放大有利样本的权重；同时，降低不利样本的权重，进而提高模型在目标领域的表现。尽管 Tradaboost 已在迁移学习领域取得一定成功，但它仍存在一定局限性，主要如下：

1.只能支持单源域迁移。传统的 Tradaboost 算法主要针对单源迁移学习任务，即只有一个源域和一个目标域。在这种情况下，它可能无法充分利用多源领域之间的信息来提高迁移学习性能。

2.基分类器的选择。Tradaboost 通常采用简单的基分类器（如决策树），在某些复杂的迁移学习任务中，这种基分类器可能无法捕获源领域和目标领域之间的复杂关系。这可能会限制 Tradaboost 的性能。

3.领域差异过大时性能下降。当源领域与目标领域之间的分布差异较大时，Tradaboost 可能无法有效地迁移知识。这是因为它依赖于源领域数据的权重调整，而在分布差异较大的情况下，这种调整可能无法弥补两个领域之间的差距。

对于问题 1，Yi Yao [41]等人在 Dai 的基础上提出了 MultiSource Tradaboost 和 TaskTradaboost 模型，但其核心思想也是在每次迭代中，选择表现最好的源域训练的分类器作为基分类器，没有充分利用多源域数据进行分析。

我们改进文献[41]，提出了 MSC Tradaboost。针对问题 1，该模型通过多个源域的基分类器协同进行判断，充分利用了多个源域的数据。针对问题 2，我们选择MLP(Multi-Layer perceptron)作为基分类器，可以更精确地提取源域以及目标域之间的深层特征。针对问题 3，我们通过预先进行城市相似度计算，选取相似度较高的城市作为源域，减小因领域差异过大带来的负迁移的风险。

4.2.2 模型结构

MSC Tradaboost 可以分为两步，具体步骤如下：

Phase I of MSC(MultiSourse Collaborative) Tradaboost

Input: 多个源域的数据T_{s_1}、T_{s_2}、T_{s_3}……T_{s_N}，目标域的数据T_d，最大迭代次数 M

Output: 初始基分类器集合δ

1.清空基分类器集合δ，给$\beta \leftarrow$

for k ←1 **to** N **do**

2. 初始化第 k 个源域的样本权重$w^k = [w_1^k, w_2^k, w_3^k \ldots . w_{n_k}^k]$，初始化各个元素为$\frac{1}{n_k}$

3. 初始化目标域的样本权重$w^d = [w_1^d, w_2^d, w_3^d \ldots . w_m^d]$，初始化各个元素为$\frac{1}{m}$

4. 结合w^d和w^k为 w，定义$\beta \leftarrow 1/\left(1 + \sqrt{2 \ln N / M}\right)$

for t ←1 **to** M **do**

5. 将w^t归一化为p^t

6. 将第 k 个源域的样本结合目标域样本，再乘权重p^t

7. 使用该数据训练分类器h_t

8. 计算分类器在目标域的错误率$\epsilon_t \leftarrow \sum_{i=n_k+1}^{n_k+m} \frac{w_i^t \cdot |h_t(x_i) - c(x_i)|}{\sum_{i=n_k+1}^{n_k+m} w_i^t}$

9. $\beta_t \leftarrow \ln(\frac{1-\epsilon_t}{\epsilon_t})/2$ **where** $\epsilon_t < 0.5$

10. $w_i^{t+1} \leftarrow \begin{cases} w_i^t \cdot \beta^{|h_t(x_i) - c(x_i)|}, & 1 \le i \le n_k \\ w_i^t \cdot \beta_t^{-|h_t(x_i) - c(x_i)|}, & n_k + 1 \le i \le n_k + m \end{cases}$

End for

11. 更新$\delta \leftarrow \delta \cup h_t$ **where** $t = argmax(\beta_t)$

End for

return δ

在本段中，我们详细阐述了 MSC Tradaboost 算法第一步的核心内容。该算法接受多个源域数据、目标域数据以及最大迭代次数 M 作为输入。我们逐个将源域与目标域进行迭代操作，并在迭代过程中选择表现最佳的分类器加入基分类器集合，以进入算法的下一阶段。

接下来是 MSCTradaboost 算法的第二步

Phase II of MSC(MultiSourse Collaborative) Tradaboost

Input: 初始基分类器集合δ，目标域的数据T_d，最大迭代次数 M

Output: 最终的强分类器$\hat{f}: X \to Y$

1.初始化目标域的样本权重$w^d = [w_1^d, w_2^d, w_3^d \ldots . w_m^d]$，初始化各个元素为$\frac{1}{m}$

for k ←1 **to** M **do**

2. 初始化各个基分类器的权重$w^h = [w_1^h, w_2^h, w_3^h \ldots . w_{|\delta|}^h]$，初始化各元素为$\frac{1}{|\delta|}$

for $t \leftarrow 1$ **to** $|\delta|$ **do**

3. 计算h_t在T_d上的错误率

$\epsilon_t \leftarrow \sum_{i=0}^{m} \frac{w_i^d \cdot |h_t(x_i) - c(x_i)|}{\sum_{i=0}^{m} w_i^d}$

4. $w_t^h \leftarrow \ln(\frac{1-\epsilon_t}{\epsilon_t})/2$ **where** $\epsilon_t < 0.5$

end for

5. 计算基分类器协同后形成的分类器H_k在T_d上的错误率

$\epsilon_k \leftarrow \sum_{i=0}^{m} \frac{w_i^d \cdot |H_k(x_i) - c(x_i)|}{\sum_{i=0}^{m} w_i^d}$ $H_k(x_i) \leftarrow \begin{cases} 1, & \sum_{t=0}^{|\delta|} \frac{w_t^h \cdot h_t(x_i)}{\sum_{t=0}^{|\delta|} w_t^h} > 0.5 \\ 0, & \sum_{t=0}^{|\delta|} \frac{w_t^h \cdot h_t(x_i)}{\sum_{t=0}^{|\delta|} w_t^h} \leq 0.5 \end{cases}$

6. $\beta_k \leftarrow \ln(\frac{1-\epsilon_k}{\epsilon_k})/2$ **where** $\epsilon_k < 0.5$

7. 更新目标域样本权重

$$w_i^d \leftarrow w_i^d \cdot e^{-\beta_k c(x_i) H_k(x_i)}$$

End for

8. return $\hat{f}(X) = \begin{cases} 1, & \sum_{k=0}^{M} H_k(X) \cdot \beta_k > 0.5 \\ 0, & \sum_{k=0}^{M} H_k(X) \cdot \beta_k \leq 0.5 \end{cases}$

在此部分中，我们基于步骤 I 得到的基分类器集合，在每次迭代中，首先根据各分类器在加权后的目标域上的表现，为它们在本次迭代中分配相应的权重，从而构建本次迭代的分类器 H_k。接下来，根据 H_k 在目标域上的表现，为 H_k 在最终分类器中分配权重 β_k，并更新样本权重。

在训练过程中，我们可以根据正负样本的比例适当调整分类器将输入判断为 1 的阈值，即调整H_k或$\hat{f}(X)$将输入判断为正向的条件。

从上述伪代码可以看出，我们的算法充分利用了多个源域的数据进行协同分析，最大化了数据的利用率以及模型在目标数据上的表现。

4.2.3 训练和优化方法

本小节从几个方面介绍模型训练和优化方法。

1.样本权重更新方法：

我们在两个阶段采用了不同的权重更新策略，降低了过拟合的风险。同时，在第一阶段采用较为温和的更新策略

$$w_i^{t+1} \leftarrow \begin{cases} w_i^t \cdot \beta^{|h_t(x_i) - c(x_i)|}, & 1 \leq i \leq n_k \\ w_i^t \cdot \beta_t^{-|h_t(x_i) - c(x_i)|}, & n_k + 1 \leq i \leq n_k + m \end{cases}$$

尽量避免因目标域和源域之间比例相差过大而导致目标域样本权重下降过快的问题；在第二阶段专注于目标域样本时，采用更快速的更新策略$w_i^d \leftarrow w_i^d \cdot e^{-\beta_k c(x_i) H_k(x_i)}$使得协同模型能更快地拟合。

2.权重计算方法：

相较于 Tradaboost 采用的$\beta_t = \epsilon_t/(1-\epsilon_t)$，本文采用更温和的$\beta_k = \ln(\frac{1-\epsilon_k}{\epsilon_k})/2$，与它类似，都要求$\epsilon < 0.5$，防止权重为 0 或者是负数。因为$\beta$同时用于后续的样本权重更新，使用更温和的$\beta$可以减少极端正确率对样本权重产生的影响。

3.基分类器的选择：

我们使用 MLP 作为系统的基分类器，它的主要结构如下图所示：

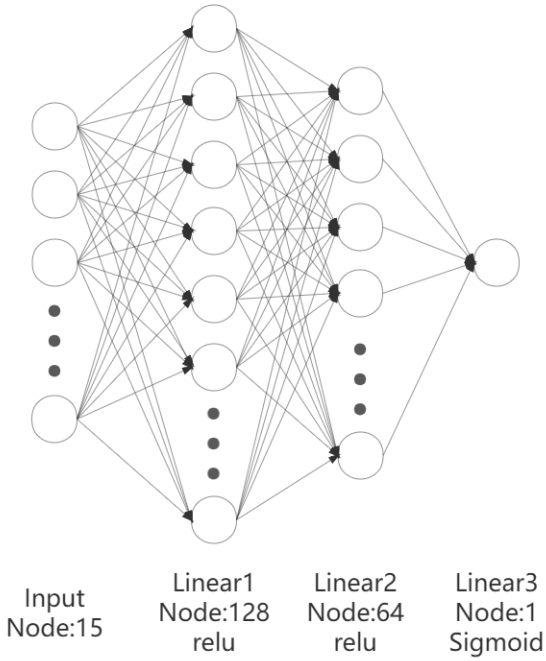

图 4-2 MLP 结构图

由图 4-2 所示，我们的基分类器 MLP 总共包括 1 层 15 维的输入层、2 层的隐藏层，其神经元数量分别是 128 和 64，以及 1 层 1 维的输出层。前向传播过程中，输入层接收各个网格经过筛选的 15 种 POI 类型，隐藏层的激活函数采用 relu 函数，输出层的激活函数采用 sigmoid 函数。

在网络训练过程中，我们采用以下几个技术进行模型训练：

(1). Kaiming 初始化。我们对各层参数使用 Kaiming 初始化进行初始化，与 relu 函数更加搭配。

(2).批标准化技术。经过隐藏层时，我们通过对每一层的输出数据进行归一化处理，使得每个特征的均值为 0、方差为 1，从而加快训练速度、提高模型性能，并有效缓解梯度爆炸以及梯度消失问题。

(3).BCELoss(Binary Cross Entropy Loss)损失函数。我们采用 BCELoss 作为损失函数，更适用于我们的二分类问题（判断网格中是否含有目标基础设施）。

4.5 本章小结

本章主要介绍了城市相似度算法和基于权重的多源迁移模型的构建过程。在城市相似度算法部分，我们首先通过图像形式概述了算法架构，然后根据算法架构进行了详细讲解。我们的城市相似度算法从 POI 数量和 POI 分布两个方面对城市间的相似度进行了比较，并创新性地将 SSIM 应用于比较 POI 分布。

在多源迁移模型部分，我们首先分析了其他基于权重的迁移模型的特点以及它们的不足之处。接着，我们以伪代码的形式详细阐述了算法的整体流程，突出了算法如何综合利用多个源域的数据进行迁移学习。最后，在训练和优化方法部分，我们讲解了一些实现细节，为读者提供了更多关于模型实际应用的信息。

通过本章的介绍，读者可以了解到城市相似度算法和基于权重的多源迁移模型的搭建方法，以及它们在解决实际问题中的应用价值。这为进一步研究和实践提供了有力的支持和参考。

5 实验与分析

在前文中，我们详细介绍了本研究针对跨城市数据迁移在基础设施领域的解决框架，为本章的实验开展奠定了扎实的理论基础。本章将结合上述理论框架，展示解决框架的具体应用流程。为了确保实验的严谨性和可靠性，本章将分别从实验目的、实验设计、实验环境与工具、过程与结果分析等方面进行详细阐述，以展现本研究在实际应用场景中的有效性和优越性。

5.1 实验目的

我们的实验目的主要有以下几点。

1.验证框架的有效性：实验旨在证明提出的跨城市数据迁移框架可以用于解决实际问题，本文以解决菏泽市的急救中心分布预测为例。

2.评估性能优势：实验需要证明提出的解决框架相较于其他的方法在性能上的优势，如预测准确性、分析时间。

3.检验方法的泛化能力：实验应评估该解决框架在不同城市、对于不同基础设施问题的泛化能力。

4.探究影响因素：探究影响实验结果的关键因素，为未来研究提供帮助。

5.2 实验设计

针对上述四个实验目的，我们结合解决框架中的几个主要方法与模型：城市边界界定、特征选择、城市相似度计算、MSC Tradaboost，设计了以下实验。

1.针对“验证框架的有效性”这一目的，我们将本研究提出的解决框架用于菏泽市急救中心分布预测，并使用专家方法对结果进行评估。

2.针对“评估性能优势”这一目的，我们基于对菏泽市的急救中心分布预测问题，拟设计多个对照实验，主要有：

(1).不采用任何方法，仅使用普通模型。

(2).不进行特征选择，但使用其他方法。

(3).不进行城市边界界定，但使用其他方法。

(4).不进行城市相似度计算，随机为目标城市选择源城市数据，但使用其他方法。

(5).在使用其他方法的基础上，不使用 MSC Tradaboost 而改为其他模型。

在进行实验后，从预测准确率和分析时间两个维度对实验结果进行分析。

3.针对“检验方法的泛化能力”目的，我们拟设计“对于菏泽市的图书馆分布预测”“对于宁德市的急救中心分布预测”与主要实验用例进行比较，仍使用专家方法对结果进行评估，证明该解决框架的泛化能力。

5.3 实验环境与工具

1.硬件环境

CPU(Central Processing Unit)：

Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 2592Mhz, 6 个内核, 12 个处理器

GPU(Graphics Processing Unit)：

(1).Intel(R) UHD Graphics

(2).NVIDIA GeForce GTX 1650 Ti

运行内存：16GB

2.软件环境

操作系统：Windows10

语言：Python 3.9.6 [MSC v.1929 64 bit (AMD64)]

相关软件包：Numpy1.22.4、Scikit-learn1.1.1、Pytorch1.10.1+cu102、Scipy1.8.1

3.实验数据

实验数据如表 3-1 所示，包括 15 座城市共计 30 种 POI 类型。

4.评估指标

专家方法：根据文献[42] 急救中心布局首先要满足服务半径的要求，宜紧靠城市交通干道并直接连接，宜面临两条道路，出入口不应少于两处，便于车辆迅速出发。本文将模型预测的急救中心所在网格的道路数量、出入口、以及 POI 总数是否满足要求，作为一个评价指标。

准确率：因为目标城市中原有的急救中心布局具有一定的参考价值，我们定义，当预测的网格的标签$\hat{y}$(0 或者 1，1 代表有急救中心)与真实标签y一致时，这一网格分类正确，计算该城市中的正确率作为一个评价指标。

运行时间：在上述的将菏泽市的急救中心预测与其他案例进行对比时，我们使用了运行时间作为评价的一个维度。我们定义的运行时间指：将数据输入模型开始运行这一时刻起，至模型输出预测结果。

5.4 过程与结果分析

本小节以预测菏泽市的急救中心为例讲述了解决框架的具体执行过程，并根据上述几个实验设计开展相应实验，对结果进行分析。

5.4.1 实验过程

1.网格化处理。

菏泽市所有 POI 中，最南的 POI 位于北纬 34.564°、最北的 POI 位于北纬 35.8545°、南北相距 143.1693km；最西的 POI 位于东经 114.8516°、最东的 POI 位于东经 116.4141°、东西相距 143.3951km。为了将城市区域划分成 1km * 1km 的网格，将南北距离、东西距离进行四舍五入，最终将城市划分成 20449 个网格。统计各网格中的 POI 种类及数量。

2.城市边界判断。

根据公式 3-1，得到菏泽市中心网格的经度为东经 115.4673°，纬度为北纬 35.2536°。具体如下图所示：

图 5-1 菏泽市中心图

将阈值θ设置为 10，得到的菏泽市城市边界分别为

南边界	北边界	西边界	东边界
北纬 35.1544°	北纬 35.3527°	东经 115.3801°	东经 115.6634

具体边界图如下图所示。

图 5-2 菏泽市城市边界图

3.特征选择。

我们将所有城市，已划分的，在城市范围内的网格整合至一块，共计 47586 个网格数据。通过公式 2-1，我们计算所有 POI 类别的 Fisher Score(注：因为急救中心属于医疗类型 POI，所以医疗类型提前去除)，结果如下：

POI 类型	Fisher Score
美食	5.83694861e-01
酒店	2.49950922e-01
购物	3.91683892e-01
生活服务	6.73126883e-01
丽人	4.83542985e-01
旅游景点	1.19544289e-01
休闲娱乐	5.81244150e-01
运动健身	6.80121262e-01
教育培训	6.39235161e-01
文化传媒	4.29833356e-01
汽车服务	7.07648615e-02
交通设施	8.47286930e-01
金融	4.34437161e-01

房地产	9.33830182e-01
公司企业	3.91593052e-01
政府机构	7.49843834e-01
出入口	8.38864728e-01
自然地物	1.01228593e-01
道路	4.91536497e-01
门址	8.41058939e-02
绿地	1.20114938e-02
行政区划	4.28445264e-02
地铁线路	2.26958501e-04
公交线路	3.45541848e-02
行政地标	2.96790382e-01
商圈	2.90284477e-03
铁路	4.16557838e-04
其他线要素	4.05618122e-03
水系	2.08194968e-05

表 5-1 各特征 Fisher Score 表

可以看到，房地产与交通设施这两种类型的 POI 具有更高的表现，这与我们主观判断相符合：急救中心应该位于交通方便、离住宅区相对较近的地方。我们

将阈值设为 0.1，滤除 Fisher Score 小于 0.1 的特征。同时，我们将拥有最高分数的房地产类型的 POI 作为初始特征，再进行 MRMR 算法构建，选择特征数量为 15。经过筛选，最后被选中的特征有：房地产、行政地标、旅游景点、文化传媒、金融、自然地物、运动健身、道路、酒店、休闲娱乐、政府机构、丽人、教育培训、公司企业、交通设施。

3.城市相似度计算。

逐一比较各个城市，关于上述十五个类型的 POI 在数量占比上、分布结构上的相似度。各城市之间的相似度如表 5-2 所示

	北京	长沙	成都	广州	杭州	合肥	菏泽	南京	宁德
北京	1	0.7550	0.7555	0.7411	0.7609	0.7478	0.7560	0.7617	0.7496
长沙	0.7550	1	0.7738	0.7848	0.7822	0.7519	0.7662	0.7467	0.7396
成都	0.7555	0.7738	1	0.7590	0.7701	0.7730	0.7714	0.7670	0.7447
广州	0.7411	0.7848	0.7590	1	0.7783	0.8045	0.8068	0.7570	0.7452
杭州	0.7609	0.7822	0.7701	0.7783	1	0.7848	0.7945	0.7785	0.7776

合肥	0.7478	0.7519	0.7730	0.8045	0.7848	1	0.7629	0.7440	0.7268
菏泽	0.7560	0.7662	0.7714	0.8068	0.7945	0.7629	1	0.7495	0.7380
南京	0.7617	0.7467	0.7669	0.7570	0.7785	0.7440	0.7495	1	0.7476
宁德	0.7496	0.7396	0.7447	0.7452	0.7776	0.7268	0.7380	0.7476	1
上海	0.7581	0.7777	0.7676	0.7542	0.7719	0.7793	0.7770	0.7790	0.7629
深圳	0.7370	0.7548	0.7637	0.8004	0.7869	0.7697	0.7689	0.7498	0.7295
天津	0.7543	0.7345	0.7490	0.7405	0.7767	0.7250	0.7359	0.7539	0.7578
武汉	0.7636	0.7552	0.7677	0.7614	0.7765	0.7548	0.7600	0.7629	0.7531
厦门	0.7381	0.7463	0.7630	0.8084	0.7825	0.7746	0.7755	0.7319	0.7409
西安	0.7580	0.7633	0.7681	0.7735	0.7738	0.7617	0.7858	0.7556	0.7622

	上海	深圳	天津	武汉	厦门	西安
北京	0.7581	0.7370	0.7543	0.7636	0.7381	0.7580
长沙	0.7777	0.7548	0.7345	0.7552	0.7463	0.7633
成都	0.7676	0.7637	0.7490	0.7677	0.7630	0.7681
广州	0.7528	0.8004	0.7406	0.7614	0.8084	0.7735
杭州	0.7719	0.7869	0.7767	0.7765	0.7825	0.7738
合肥	0.7793	0.7697	0.7250	0.7548	0.7746	0.7617
菏泽	0.7770	0.7689	0.7359	0.7600	0.7755	0.7858
南京	0.7790	0.7498	0.7539	0.7629	0.7319	0.7556
宁德	0.7629	0.7295	0.7578	0.7531	0.7409	0.7622
上海	1	0.7659	0.7711	0.7814	0.7729	0.7752
深圳	0.7659	1	0.7258	0.7469	0.7885	0.7627
天津	0.7711	0.7258	1	0.7578	0.7163	0.7438
武汉	0.7814	0.7469	0.7578	1	0.7446	0.7630
厦门	0.7729	0.7885	0.7163	0.7446	1	0.7593
西安	0.7752	0.7627	0.7438	0.7630	0.7593	1

表 5-2 城市相似度表

4.模型训练并进行预测。

首先，设置相似度阈值为 0.77，选择成都、广州、杭州、上海、厦门、西安这六个城市作为源城市。将迭代次数 M 设置为 50：基分类器 MLP、MSC Tradaboost 的第一步以及第二步的迭代次数都为 50。基分类器的学习率设为 0.01、批次大小设置为 32。预测结果如下所示：

图 5-3 菏泽市急救中心预测图

可以看到，急救分布基本覆盖主要高人口密度区。从南向北，从西至东，依次给各个急救中心所在网格标号，它们的出入口数量、道路数量以及 POI 数量总数如下表所示：

急救中心所在网格序号	出入口数量	道路数量	POI 总数
1	42	6	463
2	60	2	1033
3	78	2	697
4	46	6	883
5	28	5	481
6	50	6	1312

7	37	8	502
8	55	9	1155
9	63	10	2289
10	70	3	811

表 5-3 菏泽市预测结果表

可以看到，所有急救中心所在网格的道路数都满足国标，POI 数量也较多，表明这些网格人口流动较多，应建设急救中心。

5.4.1 结果分析

基于上一小节中介绍的实验步骤，我们依次对 5.2 中设计对照试验进行实现。结果如下图：

目标城市	基础设施	特征选择	边界界定	城市相似度计算	模型	满足国标	准确率	时间(s)
菏泽	急救中心	采用	采用	采用	MSC Tradaboost	基本满足	93.72%	2657.067
菏泽	急救中心	未采用	未采用	未采用	MLP	不满足	92.68%	2954.531
菏泽	急救中心	未采用	采用	采用	MSC Tradaboost	基本满足	94.69%	3161.905
菏泽	急救中心	采用	未采用	采用	MSC Tradaboost	不满足	93.45%	4821.631
菏泽	急救中心	采用	采用	未采用	MSC Tradaboost	不满足	90.12%	2661.307
菏泽	急救中心	采用	采用	采用	MLP	不满足	93.73%	2510.231
菏泽	图书馆	采用	采用	采用	MSC Tradaboost	基本满足	95.84%	2601.751
宁德	急救中心	采用	采用	采用	MSC Tradaboost	基本满足	93.17%	2159.845

表 5-4 实验结果表

从表 5-4 中可以观察到，采用所提出的解决框架后，在三个评价指标上均表现出优越的性能。通过使用特征选择以及边界划定方法，可以加速训练过程。同时，边界划

定方法有助于优化预测结果的表现。采用城市相似度计算和 MSC Tradaboost 模型则使预测结果表现更佳。从实验 8 和 9 中，我们还可以发现，该解决框架在预测不同城市和不同基础设施方面同样具有良好的表现，具备较强的泛化性。

5.5 本章小节

本章主要围绕实验目的、实验设计、实验环境与工具以及过程与结果分析进行了详细的阐述。在实验目的部分，我们提出了四个主要目标：验证框架的有效性、评估性能优势、检验方法的泛化能力以及探究影响因素。针对这些目标，在实验设计部分，我们设计了一个主要实验和 7 个对照实验。

在实验环境与工具部分，我们详细介绍了实验所需的软件、硬件环境、所使用的数据以及评价指标。具体而言，我们采用了专家方法、预测准确率和分析时间作为评估指标。在过程与结果分析部分，我们以主实验“预测菏泽市的急救中心分布”为例，详细地阐述了如何运用本文的解决框架来解决实际问题，并对过程中的一些结果和参数设定进行了详细的解释。

最后，通过对比主实验和对照实验在专家方法、预测准确率和分析时间等评价指标上的表现，我们证明了本文提出的解决框架的优越性。同时，在对实验过程与实验结果的分析中也表明，我们所提出的四个目标均得到了满足。

6 总结与展望

随着城市化进程的加速，城市基础设施规划成为了一个日益重要的议题。有效的城市基础设施规划能够为居民提供便利的生活条件，促进经济发展和社会进步。然而，由于城市间的差异和复杂性，如何利用大量跨城市数据进行有效的城市基础设施规划仍然是一个具有挑战性的问题。为了解决这一问题，本文提出了一个利用跨城市数据进行城市基础设施规划的解决框架，重点研究了网格化城市数据、界定城市边界、比较城市相似度以及模型构建这四个步骤。

在本章中，我们首先回顾本文的主要研究内容和贡献，然后展望未来在本文研究框架下的潜在发展方向和挑战。

6.1 本文工作总结

本文主要提出了一个利用跨城市数据进行城市基础设施规划的解决框架，涵盖了五个关键步骤：网格化城市数据、界定城市边界、特征选择、比较城市相似度和模型

构建。在界定城市边界的过程中，我们提出了一种基于兴趣点（POI）数量的城市网格判定方法。首先计算城市中 POI 数量最多的网格，然后从该网格出发，依次判断东西南北四个方向是否满足条件。通过不断迭代，确定所有城市的网格。这一步骤有效地防止非城市数据对城市数据的“污染”，同时提高了后续处理的效率。

在特征选择阶段，我们结合了 Fisher Score 和 MRMR 这两种常用的特征选择算法。首先计算各 POI 类型的 Fisher Score，过滤掉不满足筛选条件的特征，并将得分最高的特征作为初始特征进行 MRMR 算法。这一步骤提高了模型的性能、减少了计算时间、增强了模型的可解释性。

在计算城市相似度的过程中，我们综合考虑了 POI 数量特征和 POI 分布结构，创新性地使用结构相似性度量（SSIM）比较城市间的 POI 分布结构。这一步骤为后续的模型搭建筛选了源数据，有效缩小了源域和目标域之间的距离，防止了因距离过大导致的“负迁移”问题。

在模型构建阶段，我们提出了多源领域适应性增强（MSC Tradaboost）模型，充分地结合多个源域数据对目标域任务进行处理。最后，我们通过实验验证了该框架自身的有效性以及相较于其他方法的优越性。

6.2 展望

本文采用跨城市数据解决城市规划中的基础设施分布问题，但在本研究中仍存在一定的问题。以下是本文中存在的问题及我们未来期望探讨的研究方向：

1.在界定城市边界的过程中，我们仅考虑了从城市中 POI 数量最多的网格出发。这导致了算法只能界定一个城市区域。如果一个城市有多个相互疏离的城市区域，该算法只能识别出 POI 最密集的那个区域的边界，无法同时考虑多个区域。此外，如果城市中有大面积的公园、湿地、草地等 POI 数量稀疏的部分，可能导致边界判断错误，需要不断调整城市网格大小来拟合。未来，我们希望建立更精确的城市边界度量算法。

2.在选择城市数据时，我们仅使用了 POI 这种空间上静止、时间序列上变化不明显的数据作为分析依据，没有考虑城市中大量多维度的数据，如路网数据、各种轨迹数据、人流量数据、地形数据等。我们期望探讨更多的城市特征，为解决框架的各个步骤提供更多维度的数据支持，以获得更好的效果。

3.在判断基础设施分布时，没有考虑基础设施自身的性质。例如，文献[42]提出，急救中心所处的环境应安静、地形规整、工程和水文地质条件良好，并尽可能充分利用城市基础设施，应避开污染源和易燃易爆物的生产、贮存场所。我们希望未来在研究具体基础设施分布问题时能结合它们的自身特性进行度量。

参考文献

[1] Batty M. Big data, smart cities and city planning[J]. Dialogues in human geography, 2013, 3(3): 274-279.

[2] Caragliu A, Del Bo C, Nijkamp P. Smart cities in Europe[J]. Journal of urban technology, 2011, 18(2): 65-82.

[3] Cervero R, Murakami J. Effects of built environments on vehicle miles traveled: evidence from 370 US urbanized areas[J]. Environment and planning A, 2010, 42(2): 400-418.

[4] Kamruzzaman M, Baker D, Washington S, et al. Advance transit oriented development typology: case study in Brisbane, Australia[J]. Journal of transport geography, 2014, 34: 54-70.

[5] Neuman M. The compact city fallacy[J]. Journal of planning education and research, 2005, 25(1): 11-26.

[6] Hoornweg D, Bhada-Tata P. What a waste: a global review of solid waste management[J]. 2012.

[7] Batty M, Axhausen K W, Giannotti F, et al. Smart cities of the future[J]. The European Physical Journal Special Topics, 2012, 214: 481-518.

[8] Kitchin R. The real-time city? Big data and smart urbanism[J]. GeoJournal, 2014, 79: 1-14.

[9] Zheng Y, Capra L, Wolfson O, et al. Urban computing: concepts, methodologies, and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2014, 5(3): 1-55.

[10] Bao J, Zheng Y, Mokbel M F. Location-based and preference-aware recommendation using sparse geo-social networking data[C]//Proceedings of the 20th international conference on advances in geographic information systems. 2012: 199-208.

[11] 魏世尧. 基于迁移学习的城市酒店选址研究[D]. 北京交通大学, 2020. DOI:10.26944/d.cnki.gbfju.2020.003174.

[12] Batty M. The size, scale, and shape of cities[J]. science, 2008, 319(5864): 769-771.

[13] Bettencourt L M A, Lobo J, Helbing D, et al. Growth, innovation, scaling, and the pace of life in cities[J]. Proceedings of the national academy of sciences, 2007, 104(17): 7301-7306.

[14] Hall P. Cities of tomorrow: An intellectual history of urban planning and design since 1880[M]. John Wiley & Sons, 2014.

[15] Wu F. Planning for growth: Urban and regional planning in China[M]. Routledge, 2015.

[16] Kumar P, Morawska L, Martani C, et al. The rise of low-cost sensing for managing air pollution in cities[J]. Environment international, 2015, 75: 199-205.

[17] Sarker M N I, Peng Y, Yiran C, et al. Disaster resilience through big data: Way to environmental sustainability[J]. International Journal of Disaster Risk Reduction, 2020, 51: 101769.

[18] Yuan Y, Raubal M, Liu Y. Correlating mobile phone usage and travel behavior–A case study of Harbin, China[J]. Computers, Environment and Urban Systems, 2012, 36(2): 118-130.

[19] Goodchild M F. Citizens as sensors: the world of volunteered geography[J]. GeoJournal, 2007, 69: 211-221.

[20] Barthélemy M. Spatial networks[J]. Physics reports, 2011, 499(1-3): 1-101.

[21] Bettencourt L M A. The origins of scaling in cities[J]. science, 2013, 340(6139): 1438-1441.

[22] Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1).

[23] Li X, Peng L, Yao X, et al. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation[J]. Environmental pollution, 2017, 231: 997-1004.

[24] Yao Z, Fu Y, Liu B, et al. POI recommendation: A temporal matching between POI popularity and user regularity[C]//2016 IEEE 16th international conference on data mining (ICDM). IEEE, 2016: 549-558.

[25] Rozenfeld H D, Rybski D, Andrade Jr J S, et al. Laws of population growth[J]. Proceedings of the National Academy of Sciences, 2008, 105(48): 18702-18707.

[26] 许泽宁，高晓路. 基于电子地图兴趣点的城市建成区边界识别方法[J]. 地理学报, 2016, 71(06):928-939.

[27] Li J, Cheng K, Wang S, et al. Feature selection: A data perspective[J]. ACM computing surveys (CSUR), 2017, 50(6): 1-45.

[28] Gu Q, Li Z, Han J. Generalized fisher score for feature selection[J]. arXiv preprint arXiv:1202.3725, 2012.

[29] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy[J]. IEEE Transactions on pattern analysis and machine intelligence, 2005, 27(8): 1226-1238.

[30] Liu X, Long Y. Automated identification and characterization of parcels with OpenStreetMap and points of interest[J]. Environment and Planning B: Planning and Design, 2016, 43(2): 341-360.

[31] Preoţiuc-Pietro D, Cranshaw J, Yano T. Exploring venue-based city-to-city similarity measures[C]//Proceedings of the 2nd ACM SIGKDD international workshop on urban computing. 2013: 1-4.

[32] McKenzie G. Shared micro-mobility patterns as measures of city similarity: Position Paper[C]//Proceedings of the 1st ACM SIGSPATIAL International Workshop on Computing with Multifaceted Movement Data. 2019: 1-4.

[33] Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment: from error visibility to structural similarity[J]. IEEE transactions on image processing, 2004, 13(4): 600-612.

[34] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.

[35] 龙明盛. 迁移学习问题与方法研究[D]. 清华大学, 2014.

[36] Dai W, Yang Q, Xue G R, et al. Self-taught clustering[C]//Proceedings of the 25th international conference on Machine learning. 2008: 200-207.

[37] Samanta S, Selvan A T, Das S. Cross-domain clustering performed by transfer of knowledge across domains[C]//2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG). IEEE, 2013: 1-4.

[38] Fengmei W, Jianpei Z, Yan C, et al. FSFP: Transfer learning from long texts to the short[J]. Applied Mathematics & Information Sciences, 2014, 8(4): 2033.

[39] 庄福振，罗平，何清，史忠植. 迁移学习研究进展[J]. 软件学报, 2015, 26(01):26-39. DOI:10.13328/j.cnki.jos.004631.

[40] Dai W, Yang Q, Xue G, et al. Boosting for transfer learning[C]// Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007. ACM, 2007.

[41] Yao Y, Doretto G. Boosting for transfer learning with multiple sources[C]//2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010: 1855-1862.

[42] 建标 177-2016，急救中心建设标准[S].

致 谢

抽象百度 API，爬数据还要 money。