学号 2014302590177

密级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**武汉大学本科毕业论文**

基于点空间格局的行业时空演化分析

及其并行算法实现

院（系）名 称：遥感信息工程学院

专 业 名 称 ：遥感科学与技术

学 生 姓 名 ：宋云婷

指 导 教 师 ：桂志鹏 副教授

二○一八年六月

**BACHELOR'S DEGREE THESIS**

**OF WUHAN UNIVERSITY**

**Analyzing spatiotemporal dynamics of industries based on point pattern analysis and its parallel algorithm implementation**

College： School of Remote Sensing and Information Engineering

Subject： Remote Sensing Science and Technology

Name： Yunting Song

Directed by：Zhipeng Gui Associate Professor

June 2018

**郑 重 声 明**

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名： 日期：

摘 要

分析不同产业的时空分布格局及其演化过程，有助于政府部门及企业了解产业的宏观发展趋势，进而为产业空间规划提供参考。然而，时空分析过程是一项富有挑战性的任务，它需要一种直观易于理解的信息呈现机制和能够应对计算密集型数据分析的强大计算技术的支撑。

针对上述问题本文选取标准差椭圆[1]（Standard deviational ellipse, SDE）和重心迁移轨迹方法建立行业时空演化规律分析方案，并进而设计了相应并行化算法以提高计算效率，同时提出了一个基于Web的可视化交互框架用于展示分析结果。本文以重庆市为例，基于重庆市1960年-2016年企业注册数据，利用所选方法进行了重庆市及其各分区的行业时空演化规律分析。重心和椭圆的计算采用Apache Spark[2]进行并行化加速处理。效率对比实验说明，并行化算法可以将时空演化计算速度提升到合理范围内，以支持实时/准实时的可视化分析。

本文主要工作如下：

1. 设计重心及标准差椭圆的并行化计算算法，并在Apache Spark平台上加以实现。此外针对算法本身特点及Spark计算方法特点，对算法中操作进行了合理选择，充分提高了计算效率。
2. 建立基于重心轨迹及标准差椭圆的重庆市行业时空演化web可视化系统。本文首先设计了基于web的可视化框架，并对各层的设计方案及实现效果进行了介绍。之后，基于可视化框架，实现了相应的可视化效果，并开发了可连接数据库及Apache Spark计算平台的相应服务接口。
3. 开展并行化计算效率对比实验。本实验用于探究影响集群计算效率的因素，及检验该算法是否具有支持实时计算的可能。
4. 基于开发可视化平台，分析了重庆市及其两个主要城区的行业时空演化规律形成原因，并为相关决策与未来研究方向提供参考信息。

**关键词：**时空演化分析；Spark；重心轨迹；Web可视化；标准差椭圆

**ABSTRACT**

Analyzing the spatio-temporal distribution pattern and evolution of different industries will help government departments and companies understand the industry trends and provide references for industrial spatial planning. However, the spatio-temporal analysis process is a challenging task that requires an intuitive and easy-to-understand information presentation mechanism, and the support of powerful computing technologies that can handle computationally intensive data analysis.

In order to solve the above problems, this paper selects the standard deviation ellipse [1] (SDE) and gravity center migration trajectory method to establish the analysis scheme of the spatial and temporal evolution of industries. This paper also designs the corresponding parallelization algorithm to improve the computational efficiency. At the same time, it proposes a web-based visualization framework. The visual interactive framework is used to display the analysis results. This article takes Chongqing as an example, and based on Chongqing's 1960-2016 enterprise registration data, using the selected methods to analyze the spatial and temporal evolution of the industry in Chongqing and its sub-regions. The parallelization acceleration processing of the calculation is based on Apache Spark [2]. The efficiency comparison experiment shows that the parallelization algorithm can increase the computational speed of the analysis process to a reasonable range to support real-time/quasi-real-time visual analysis.

The main work of this article is as follows:

1. The parallel computing algorithm for design center of gravity and standard deviation ellipse is implemented on the Apache Spark platform. In addition, according to the characteristics of the algorithm itself and the characteristics of the Spark calculation method, reasonable selection of operations in the algorithm is performed, which fully improves the computational efficiency.
2. Establish a system of spatial-temporal evolution network of Chongqing's industry based on the center-of-gravity trajectory and standard deviation ellipse. This article first designed a web-based visualization framework and introduced the design scheme and implementation effect of each layer. After that, based on the visualization framework, corresponding visual effects were achieved, and corresponding service interfaces that can connect to the database and the Apache Spark computing platform were developed.
3. Carry out parallel computational efficiency comparison experiments. This experiment is used to explore the factors that affect the computational efficiency of the cluster and to test whether the algorithm has the potential to support real-time calculations.
4. Based on the development visualization platform, this paper analyzes the causes of the spatial and temporal evolution of the industry in Chongqing and its two major urban areas. Those analysis provides reference information for relevant decisions and future research directions.

**Key words:** Spatio-temporal evolutionary analysis; Spark; Gravity center trajectory; Web visualization; Standard deviation ellipse

**目 录**

[1 绪论 1](#_Toc515308537)

[1.1 研究背景及意义 1](#_Toc515308538)

[1.2 国内外研究现状 2](#_Toc515308539)

[1.2.1 行业集聚及时空演化的研究现状 2](#_Toc515308540)

[1.2.2 时空可视化的研究现状 3](#_Toc515308541)

[1.2.3 时空数据分析计算加速的研究现状 4](#_Toc515308542)

[1.3 研究目标与技术路线 5](#_Toc515308543)

[1.4 论文组织结构 6](#_Toc515308544)

[2 行业时空演化分析方法概述 8](#_Toc515308545)

[2.1 点空间格局分析 8](#_Toc515308546)

[2.1.1 统计数据分析 8](#_Toc515308547)

[2.1.2 近邻算法 9](#_Toc515308548)

[2.1.3 多距离空间聚类分析 9](#_Toc515308549)

[2.1.4 热点及聚类分析 10](#_Toc515308550)

[2.1.5 邻接矩阵比较法 11](#_Toc515308551)

[2.2 分析方法的对比与选择 11](#_Toc515308552)

[2.2.1 常用方法对比 11](#_Toc515308553)

[2.2.2 本文分析方法选择 12](#_Toc515308554)

[2.3 行业时空演化分析 13](#_Toc515308555)

[2.3.1 重心轨迹分析 13](#_Toc515308556)

[2.3.2 标准差椭圆分析 14](#_Toc515308557)

[2.4 本章小结 14](#_Toc515308558)

[3 行业时空演化的Web可视化框架设计 16](#_Toc515308559)

[3.1 Web可视化技术综述 16](#_Toc515308560)

[3.1.1 Web框架 16](#_Toc515308561)

[3.1.2 Web可视化开源库 17](#_Toc515308562)

[3.2 可视化系统架构设计 18](#_Toc515308563)

[3.3 数据库设计 19](#_Toc515308564)

[3.3.1 注册信息数据库 19](#_Toc515308565)

[3.3.2 分析结果数据库 20](#_Toc515308566)

[3.4 服务接口设计 20](#_Toc515308567)

[3.4.1 RESTful架构风格介绍 21](#_Toc515308568)

[3.4.2 Spark计算服务介绍 21](#_Toc515308569)

[3.4.3 本文服务接口设计 22](#_Toc515308570)

[3.5 可视化界面设计 24](#_Toc515308571)

[3.5.1 可视化界面功能设计 24](#_Toc515308572)

[3.5.2 可视化界面布局设计 25](#_Toc515308573)

[3.6 本章小结 26](#_Toc515308574)

[4 基于Spark的时空演化分析算法并行化 28](#_Toc515308575)

[4.1 并行计算技术概述 28](#_Toc515308576)

[4.2 Spark框架介绍 29](#_Toc515308577)

[4.2.1 Spark生态系统介绍 30](#_Toc515308578)

[4.2.2 Spark工作原理 31](#_Toc515308579)

[4.2.3 Spark部署方式 33](#_Toc515308580)

[4.3 重心轨迹及标准差椭圆并行化算法 34](#_Toc515308581)

[4.3.1 算法设计与实现 34](#_Toc515308582)

[4.3.2 并行算法优化 36](#_Toc515308583)

[4.4 本章小结 37](#_Toc515308584)

[5 实验与结果分析 38](#_Toc515308585)

[5.1 实验数据 38](#_Toc515308586)

[5.1.1 数据说明 38](#_Toc515308587)

[5.1.2 数据预处理 38](#_Toc515308588)

[5.2 可视化方案实现效果 39](#_Toc515308589)

[5.2.1 可视化系统界面 39](#_Toc515308590)

[5.2.2 可视化系统功能 40](#_Toc515308591)

[5.3 并行化效率对比实验 41](#_Toc515308592)

[5.3.1 不同数据量下的计算性能试验 41](#_Toc515308593)

[5.3.2 不同集群配置下的计算性能试验 42](#_Toc515308594)

[5.4 重庆市行业时空演化分析 43](#_Toc515308595)

[5.4.1 重庆市整体时空演化规律 43](#_Toc515308596)

[5.4.2 渝北区时空演化规律 46](#_Toc515308597)

[5.4.3 九龙坡区时空演化规律 47](#_Toc515308598)

[5.5 本章小结 49](#_Toc515308599)

[6 总结与展望 50](#_Toc515308600)

[6.1 总结 50](#_Toc515308601)

[6.2 展望 51](#_Toc515308602)

[致谢 57](#_Toc515308603)

[附录 59](#_Toc515308604)

# 绪论

## 研究背景及意义

进入21世纪以来，互联网与大数据的飞速发展，给各行各业都带来了新的机遇与挑战。大数据分析被广泛应用于各个领域，如医疗、执法、交通、城市建设等。在大数据发展的趋势下，获取各类数据的相关技术日渐成熟，数据成为了一种新的亟待利用的宝贵资源。由各类平台、传感器获取的数据种类繁多，用途多样，具有体量巨大、更新速度快与价值密度低等特点。如何合理利用这些数据进行分析、挖掘，以获取更多信息，成为了新的研究热点。另一方面，由于大数据本身特点，处理过程中对于算法效率、存储效率与运算速度都有着较高的要求，在这些要求的促进下，新的计算技术与优化方法不断被提出与应用。

在众多大数据的应用领域中，经济属于较早且最为广泛利用大数据的领域之一。习近平总书记也曾指出，全面实施促进大数据发展行动，有利于为经济社会持续健康发展增添新动能。[3]对于国家经济发展而言，行业发展是其缩影与根基，从各行业的发展特点可以对经济发展健康程度进行评估，同时也只有当各行业蓬勃、稳健发展时，国家经济才能稳步提升。同时，2012年以来，我国经济由过去的快速增长变为中高速增长，经济发展过程中，区域协调发展以及共同富裕成为了关键的发展准则[4]。对行业发展空间格局的研究，乃至对其空间格局随时间变化情况的研究，可帮助政府部门了解行业发展的区域协调性，了解区域间经济发展差距。这样的研究在近年的经济发展政策下，显得格外重要与迫切。

针对行业进行分析，最为直接而有效的方法为对各行业的企业信息进行分析。企业数据有多种，包含有区域的统计数据和基于单个企业的点数据。利用统计数据可以了解区域的整体情况，但对其内部分布规律则难以分析。基于点数据可以计算特定区域内的统计数据或指标，或针对点的分布特征进行分析，因此在研究企业的空间格局时，往往利用点数据进行分析。

点数据分析方法有许多，例如表现集中趋势和范围的重心及标准差椭圆，表示点集聚程度的聚类分析，用于预测下一点位置的相关随机游动模型等。如何从众多点数据分析方法中，选取一个适用于获取用户所关心的行业时空演化规律的方法，也是一个值得关注的问题。在分析比较各类方法适用范围后，本文选用重心轨迹及标准差椭圆方法，针对重庆市的企业注册数据，进行了行业时空演化分析。由于企业数据具有量大且每年更新的特点，本文根据这两个分析方法，设计了并行化算法并在分布式计算框架Apache Spark实现，以提高运算效率；同时，要清晰地展现不同行政级的不同行业时空演化特点，仅仅依靠简单的图表是难以完成的，因此本文选用基于Web的可视化系统来进行行业点空间格局的时空演化特点展示。

本文的研究有利于用户获取重庆市及其各分区的行业点空间格局及其时空演化特点。行业空间格局可以帮助政府部门监测各行业运行的健康程度，避免了行业的不平衡发展。这一空间格局随时间的演化规律则可以帮助企业管理人员了解行业发展趋势，有利于他们从中发现新的机遇。

## 国内外研究现状

### 行业集聚及时空演化的研究现状

时空数据分析是近年来地理信息科学的研究热点[5]，相关研究将时间维度信息加入传统的地理学分析方法中，得出新的分析方法和思考角度。近年来时空分析相关的研究主要集中在人口、经济、交通及自然等方面，时空数据分析相较于单纯地理分析，可以提供更多的预测或是地理特征发展趋势分析信息。举例而言，交通方面，时空分析主要为行人或车辆的轨迹研究[6][7] 。这些研究提供了行人行为及交通流的特征信息，可以为进一步对行人兴趣点或是道路交通情况进行预测提供帮助。

时空分析的主要研究内容有如下几种：a) 对于不同时间的空间属性差别进行分析；b) 在时空中跟踪单个或多个对象，研究其轨迹；c) 对于随时间推移，点（或事件）模式的分析；d) 随时间推移，区域的模式分析。这些方法也被分别应用于行业的相关时空演化研究上。郑长娟等利用Moran’s I 指数及热点分析等方法分析浙江服务业发展的空间分异和集聚特征及其演化[8]，但研究只针对单一行业，且只是判断了集聚情况，没有考虑离散特点。相关研究利用偏离份额模型分析北京市各产业以城区为单位的分布规律及其随时间的变化，属于对于区域空间属性随时间变化的分析，是以城区为单位的研究[9]。张珣等利用核密度(Kernel)、标准差椭圆、Ripley's K(r)函数等点模式分析方法研究各年北京城区商业网点分布情况[10]，但此研究只针对了商业网点进行分析，对于其它行业以及其它区域的对比等都没有进行过多着重。相较于过去的从时间或空间单一维度来进行产业发展的研究相比，现在越来越多的研究将重心放在了产业分布的时空变化之上，这是因为大数据的发展，致使数据的获取较过去更为容易，且获取数据的质量也有了较大的提升。但在这些研究中，仍存在有分析粒度过粗、对于空间特性随时间变化的规律表现不够等问题。

考虑到现有研究的不足，与本文现阶段拥有的主要研究数据特性，结合时空分析研究的趋势，本文决定选用点模式分析方法，获得细粒度的行业分布规律时空演化。

### 时空可视化的研究现状

数据可视化的方法有许多，主要可分为基于几何投影的可视化、基于图形的可视化、基于3D模型的可视化以及基于动画的可视化等。而在时空分析结果的可视化研究上，学者们主要着眼于如何美观、清晰地表达更大的信息量。由于时空数据本身具有多维度，数据量大的特点，交互式的可视化界面显得尤其重要。

K.Hewagamage等于1999年就开始利用螺旋图形与几何投影对各种时空数据进行描述[11]；Diansheng Guo等曾提出过一种支持多变量选择与用户交互的时空数据可视化系统VIS-STAMP，其中主要采用的是可进行交互的图表与相关动画进行的展示[12]；之后相关的时空可视化研究开始向三维或动画进行发展，类似的研究有对游戏中玩家的位置随时间变化规律的研究[13]，对于海洋与大气中CO2含量时空变化的可视化[14]等。但无论选择什么样的可视化方法，这些研究所提出的可视化方案均具有如下三个特点：（1）选取合适的表达方式展现空间数据随时间的变化（如时间轴，动画等）；（2）尽可能美观、简洁地展现用户需要的信息；（3）提供交互方式，帮助用户找到自己关注的信息，避免不关心的信息。这三点也将是本文中所提出的可视化方案需要格外注意的地方。

而在大数据与互联网流行的当下，对数据可视化的要求除了清晰、美观之外，还有了更进一步的便于分享，便于使用的要求。这也是越来越多的研究着力于基于Web的可视化方法的原因。相较于基于软件的可视化，Web可视化具有如下优点:（1）用户使用成本低，用户只需连接互联网，而无需下载；（2）更便于用户间传播分享；（3）支持平台更多，支持的可视化方法也更多。较早期的Web可视化模型有1996年由K Brodlie提出的基于数据、用户和可视化的可视化方案[15]，M Bender等于2000年提出的基于Web的可视化框架[16]等。这些研究为最早的web可视化框架奠定了基础，虽然这些研究中的可视化模型一般功能较少且往往不具有对可视化对象的针对性，但是整体上已经与之后的可视化框架类似。而在之后的Web可视化研究中，研究者更多的将目光放在功能的实现与对可视化主题的适应上，而基本遵循了类似的框架结构。相关的web可视化研究主题集中于医学、生物、房产、地理信息及环境等，对于行业，尤其是不同尺度、不同区域的行业信息可视化仍较缺乏。综合上述研究背景，本文将采用基于Web的可视化系统来对行业时空演化特征进行展现。

### 时空数据分析计算加速的研究现状

在中国每天有超过12000个新企业注册，截至2017年底，全国企业总量已达2907万，由于存在有注销企业，实际累计企业注册信息存量可达1亿条[17]。在对所有企业的信息进行分析时，如此庞大的数据量会致使计算过程耗时长，无法及时获取分析结果。在进行较复杂的分析时，计算效率优化成为了无法避免的需求。另一方面，由于注册数据每日更新，且可视化系统中需要提供对于用户自选属性等的支持，实时计算成为了一个必要的需求。在上述需求的驱动下，本文将对此次时空数据分析的计算效率进行优化，以达到近实时计算需求。

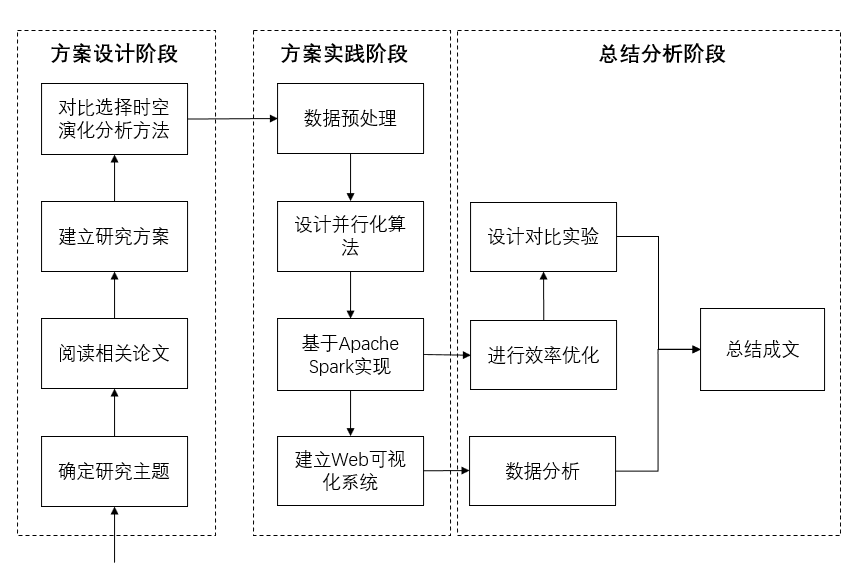
数据计算效率优化一直以来都是研究的热点，其中主要的几类优化方向有：

1. 并行计算。并行计算的研究从最初的硬件优化，即采用多核、GPU进行计算[18]，发展到后来的网格计算[19]、云计算[20]等，从单机性能的提升发展到了计算平台的横向扩展。
2. 算法优化。针对算法的目标，对算法的思路或步骤进行调整，可以在最大程度利用硬件的前提下，进一步从根本提高计算效率。这样的优化往往具有针对性，对于各种算法有不同的优化方案，如利用分治法来进行多边形合并算法的优化[21]，利用二叉堆结构进行Dijkstra最短路径算法优化[22]等。
3. 对数据索引方式及存储结构的优化。如Fellegara, R.等所提出的基于拓扑关系来存储数据点的地形树结构，可以有效地提升不规则三角网（TIN）的显示与分析效率[23]。相关研究利用空间结构对数据进行存储，很大程度上加快了分析计算中数据的读取与索引过程。

而在上述发展方向之中，并行计算自2003年MapReduce并行计算框架被提出后，得到了快速的发展，大量新分布式计算框架在那之后被提出。并且这类分布式计算框架大多提供丰富的接口，可与Web可视化系统连接，最终实现实时计算的效果。因此本文将采用并行化的方法对计算进行效率提高。传统的数据处理方法如R, Python在处理大规模数据时，往往耗时长且耗内存大。与之相对的，分布式计算平台和并行计算可以有效地提高计算效率并节省内存。最广为人知的几种分布式计算平台有Apache Hadoop[24](2007), Apache Spark[2](2012), Apache Storm[25](2012)等。其中Apache Spark因具有性能高，组件支持功能多以及与其他编程语言适应性好等优点，成为了许多大数据分析项目的最佳选择。

## 研究目标与技术路线

基于相关研究现状及发展趋势，本文将以重庆市为例，利用重心轨迹及标准差椭圆这样的点模式分析方法对当地行业的时空演化趋势进行分析，并建立合适的可视化方案。除此之外，为提高数据分析的效率，本文还将采用分布式计算平台Apache Spark进行并行算法的实现，并进行相关性能优化。本文研究的具体技术路线见图1.1：

图**1.1** 本文技术路线图

本文的具体研究内容如下：

* 1. 行业时空演化分析的并行算法设计。根据所选择的行业时空演化分析方法，设计基于Apache Spark计算框架的并行算法，并利用重庆市1950年-2016年企业注册数据进行数据分析。设计实验对所设计算法的运算性能进行对比分析及效率优化。
  2. 基于web的行业时空演化可视化框架的建立。为展示重庆市及其子行政区各行业发展格局的时空演化趋势，该可视化框架应当具有如下特点：a)能够展示多尺度多维度的信息；b)利用合适的交互功能以强调用户所关心的信息，而减少不需要的信息的干扰；c)利用动画或其他方式合适地表现空间特性随时间的变化。
  3. 深入分析时空演化原因。基于设计的算法和开发的Web平台，对重庆市企业注册数据进行分析与Web可视化，挖掘重庆市各行业及各子行政区时空演化趋势产生的原因，及不同行业或子行政区间演化趋势的差异及原因。

## 论文组织结构

本论文分为六章，各章节内容如下：

第一章，绪论。阐述行业时空演化分析的研究背景，总结国内外关于行业时空演化分析、可视化及时空分析计算加速的研究现状，最后阐明本文的研究内容与意义。

第二章，行业时空演化分析方法概述。其中首先分类对经典的点空间格局分析方法与其适用范围进行讨论，并从中选取适宜于行业时空演化分析的方法。之后本章介绍所选取的重心轨迹迁移和标准差椭圆方法进行具体介绍。

第三章，行业时空演化的Web可视化框架设计。本章主要介绍基于重心轨迹及标准差椭圆的web可视化方案设计，包括有：a) 对于所使用到的可视化技术的介绍；b) 可视化框架的分层介绍；c）从原始数据到可视化层的数据传递过程这三个部分的内容。

第四章，基于Spark的时空演化分析算法并行化。本章主要介绍计算所采用的分布式计算平台Apache Spark的特点，并展示重心及标准差椭圆计算的并行实现思路及算法流程。

第五章，实验与结果分析。本章主要分为三部分，第一部分为可视化方案具体实现效果，第二部分是并行计算效率的实验结果部分，第三部分是基于可视化结果的重庆市行业时空演化分析。通过这三部分，对于整篇论文的成果进行总结与展示。

第六章，总结与展望。总结本论文的研究成果与不足，对未来研究方向进行展望与思考。

# 行业时空演化分析方法概述

企业在区域中呈点状分布，因此在利用企业数据对行业时空演化进行分析时，一般优先采用点模式分析方法进行空间格局分析。利用点模式分析可以获取细粒度的行业分布信息，同时通过对于不同时间节点的点模式分析结果的对比，可以展现行业空间分布随时间的演化规律。针对行业时空演化分析这一目的，本章就相关的点空间格局分析方法进行讨论对比，并从中选取重心轨迹与标准差椭圆这两个方法作为本文的分析方法。

## 点空间格局分析

依照文献[3]中所提出的点数据分析方法分类标准，相关方法大致可分为五类：统计数据分析，近邻算法，多距离空间聚类分析，热点及聚类分析及邻接矩阵比较法。也有研究将这些分析方法分为两类，基于距离的方法与基于密度的方法。其中前者认为样方分析会丢失一定的点集信息，因此没有将其纳入讨论，而只讨论了针对一定有意义范围内的所有点事件的分析方法。而后者将样方分析纳入了基于距离的方法。由于本文研究对象以行政区划为单位，与第一种分类方式的研究对象相同，因此将根据第一种分类方法进行讨论。

### 统计数据分析

利用区域内的点数据可以计算一系列统计数据，一般用于衡量事件或物体集中趋势和传播范围。具体而言包括：确定点集的中心位置[26]（通常是算术平均值，也可为考虑地形的几何平均中心），中位数中心的确定，点集的方向确定，标准距离[27]，加权标准距离和标准偏差椭圆[1]等。这类分析方法既可以分析点数据集的距离特点，又可以根据点集密度进行分析。统计数据分析方法属于较初步的分析方法，可以描述点集的大概信息，为下一步深入分析提供方向。

点集的重心，即算数平均中心，一般用于表现点集的整体偏重，单个重心往往只能表现点集分布是否与地理分布一致，但如果加入重心的时间属性，则可表现点集的迁移特点和发展趋势。而中位数中心则是指能将所有数据均分的点，即到所有其他点距离和最小的点。相较于重心而言，中位数中心对异常值较为不敏感。相较于重心的计算，中位数中心的计算较为复杂，最早的有效计算方法为1962年由Kulin和Kuenne提出的迭代算法[28]。

点集的方向确定有两种方法，一种是标准差椭圆[1]，用于分析不考虑先后顺序的点集的方向分布，可表现数据集分布的方向与范围；一种是自相关随机游走模型，用于分析一连串事件点的位置，进而预测下一次事件发生的时间与地点。

总体而言，统计数据分析属于较为初步的分析，但是可对点数据集的分布特征进行基本的描述，帮助研究人员更加清晰地观察该数据集。

### 近邻算法

在许多领域的研究中，事件之间的距离（例如树木，疾病，鸟巢等）反映了潜在的过程，例如竞争食物或营养物质，分娩过程，感染或传染病。因此，相关研究往往特别关注离特点的事件或兴趣点位置最近的点。这个最接近的事件被称为最近邻居（NN）或一阶最近邻居，获取该事件的方法被称为最近邻法。

然后第二个最接近的事件是二阶最近邻居，以此类推到第k阶NN，即kNN算法[29]。kNN法相较于最近邻法更为成熟，其思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。这类方法一般计算区域内所有点的最邻近距离均值，再将其与完全随机空间分布（Complete Spatial Randomness, CSR）下的这一值进行对比[30]，进而判断点集分布的特点。若这一比值小于1，则说明数据点在空间上可能具有聚集的特点，若大于或等于1，则表示基本均匀分布或是随机分布。

近邻算法往往用于需要关注点间相互关系与影响时，这一方法所得结果可以很好地表现局部的特征，但是会一定程度上忽略整体的信息。同时，与CSR分布特征作比较的结果，可以展现点集分布上的特殊性，包括聚集程度和离散程度。而其中聚集程度的进一步分析则可由热点分析来完成。

### 多距离空间聚类分析

对于聚类分析而言，距离阈值会对结果产生较大的影响，对于同样的点数据集，不同的阈值会导致完全不同的结果。因此就产生了一种可以避免选取单一聚类距离的分析方法——Ripley's K函数[31]，这一方法可以对某一距离范围内的聚类情况进行汇总。Ripley's K 函数可表明点数据的聚集程度与离散程度在邻域大小发生变化时是如何变化的。该函数的基本计算流程为：

1. 在每个点周围以起始距离d画圆；
2. 计算在这个范围内的事件点密度；
3. 半径d增加距离增量s，以新的半径画圆；
4. 重复（2）、（3）步骤，直至新的半径达到最大值；
5. 计算这个范围内的事件点密度；
6. 将（5）中密度与每一个特定距离范围内密度比较，后者较大，则说明数据集在该范围内聚类，反之则是在该范围内扩散。

除此之外Ripley’s K函数还有多种变体，也被纳入在此类方法中。其中最为常见的一种变化称为Ripley’s L函数。这一类方法多用于分析种群或景观生态多尺度空间格局描述研究中，也有部分研究将其用于城市内商业点分布研究上。但是这类研究中的尺度分级是按照距离进行分级，对于自然相关研究较为合适，对于经济相关研究，缺少相应的行政尺度分级。

### 热点及聚类分析

聚类分析本质上是点集的划分，不同的聚类方法只是划分的依据不同。这一方法常用于空间或时空点集的分析，可有效地减少分析对象的数量。常用的聚类方法可分为划分法、层次法、基于距离的方法、基于密度的方法、基于网格的方法和基于模型的方法。举例而言，几种常用的聚类方法中K均值聚类[32]属于基于距离的聚类方法，而DENCLUE聚类[33]为典型的基于密度的聚类方法，小波聚类[34]则是基于网格的聚类方法。无论是何种聚类方法，其一般化的算法过程均为：（1）发现任意形状、大小的聚类；（2）处理孤立点；（3）重复扫描；（4）设置参数为终止条件。

聚类分析可用于发现数据点的聚集特点，并且通过分析聚类随时间的变化或随环境的变化进行更深一步的分析。但是识别空间和时空数据集中存在的聚类不能提供关于聚类的性质和模式的详细情况，因此研究可进一步将简单的热点识别[35]技术应用于这些数据集。热点分析是高低值聚类的延伸，探究的是高值的聚集点和低值的聚集点。即该方法不只是寻找点集中的高值或低值，而是要寻找周围也是高值或低值的极值点，这样就形成了热点和冷点。

从热点分析及聚类分析的定义中不难发现，这两种方法在分析点数据集的分布特点时，尤其是在分析点集的聚集情况时可起到极大的帮助。它们也同样适用于企业点数据的分析，通过聚类方法，可以分析出不同行业企业是否有明显的聚集情况，进而分析产业的发展情况。而通过热点分析，可以进行大规模企业聚集情况的探究，对于行业分析也可提供较大的帮助。但这两种方法同样有其缺点，在数据量较大时，利用这类方法获得的聚类数若过少，则不一定能反映实际聚类情况；若过多，则可能造成结果不够清晰，难以进行下一步分析。

### 邻接矩阵比较法

对于已知位置的点数据集，可以利用位置数据构建距离矩阵，并且类似的可以利用属性数据构建“接近度”矩阵[36]。可以通过“接近度”矩阵和距离矩阵之间的相关性获取该属性是否与位置相关。在空间生态学中，研究常用Mantel检验[37]来解决这一类问题。Hubert等[38]也曾就矩阵比较提出了一个用于衡量两个矩阵相似程度的指标及一个用于探究属性间可能有的依赖关系的重要性测试策略。

这一方法主要用于探究点数据的属性与空间位置的相关性，而在近年的研究中多被空间自回归和贝叶斯建模所替代。

## 分析方法的对比与选择

### 常用方法对比

从上述讨论不难发现，点集分析的方法有许多，并且相互之间存在有一定的联系。而这些相互联系的点模式分析方法，在实际利用中，具有一些共同的值得注意的问题：

* 1. 在分析点模式的过程中应当采用多种分析方法，因为单一方法往往会产生有误导性的结果；分析点模式中所选取的变量不同可能会产生差异较大的结果，具体取决于所选的值和模型。边界，区域定义，尺度，时间变化和非空间数据问题对描述和建模点模式具有重要意义；
  2. 点模式分析往往基于区域，基于区域计数的计算易受到极端值和计算比率问题的影响；
  3. 点模式分析应该被看作是探索性的，初步的，应该是整个深入分析过程的一部分；
  4. 这类分析不会提供有意义的因果关系及过程指示。研究者很难从点模式的分析中找到这种模式的成因，因此往往也需要结合其他的分析方法。

除去这些共性的特点之外，上述五类方法具有其各自的应用范围和相对的优缺点，具体总结对比见表2.1。

表**2.1** 点空间格局分析方法对比表

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 适用范围 | 优点 | 不足 |
| 统计数据分析 | 对于原始数据进行初步分析 | 可以结合多种方法进行分析；可进行不同时间的分析结果对比 | 难以获得根本性的结论 |
| 近邻算法 | 用于分析邻近点间相互作用与相关性 | 可以很好的展现局部聚集及离散特征 | 会忽略整体信息；难以进行时间维度上的比较 |
| 多距离空间聚类分析 | 可以对某一距离范围内的聚类情况进行汇总 | 可避免选取单一的聚类阈值 | 尺度划分不与行政分级对应 |
| 热点及聚类分析 | 探究点集在空间上的划分及聚集情况 | 可展现点集分布的聚集性；可延伸为时空聚类分析 | 聚类阈值难以确定；需要进行初步分析 |
| 邻接矩阵比较 | 进行属性间相关度的探究 | 可说明属性或种群间依赖性 | 可被其它方法替代 |

### 本文分析方法选择

对于本文对象——行业中的企业分布时空演化特征而言，若选用最邻近分析或是聚类分析可以较好地获得数据的集聚情况，但是由于数据量较大，包括各行各业不同时期的数据，直接地进行此类分析可能导致研究结果不明确，没有实际意义。对于这类数据的初步研究，采用统计分析便于研究人员快速、直观获取数据的整体情况与特点，为下一步分析建立计划。另一方面，热点分析及聚类分析在展现空间集聚性或者时空集聚性时具有较好的效果，但是这两种方法难以比较不同时间、行业等的分析结果，难以表现空间格局随时间的变化特征。

因此本文将采用统计分析的方法对现有企业注册数据进行分析。另一方面，为了能同时获取点数据集的集聚性与离散性两方面特征，本文将选取重心迁移轨迹与标准差椭圆这两种方法进行共同分析。

## 行业时空演化分析

基于2.2节中多种点数据分析方法的对比分析，本文决定采用重心轨迹和标准差椭圆对不同行业的企业分布时空演化规律进行分析。相较于其他方法而言，这两种方法简便、清晰，在处理大量的、未进行过分析的数据时仍可得出有意义的结论，且可以为进一步的分析提供方向。并且这两种方法与时空数据有更好的适应性，重心轨迹本身是对重心随时间的空间变化描述，另一方面标准差椭圆以重心为中心，沿重心的时空轨迹对数据集的特性进行了描述。这两种方法相结合，可以初步地展现点数据集的时空演化特性。

### 重心轨迹分析

重心利用的是各行业各年份的全部企业位置计算而得，具体公式如下：

**（2.1）**

其中)为所计算空间单元内所有企业的重心，()为第i个企业的经纬度，为该企业的权重。在数据允许的情况下，权重应当为一个可以表现企业规模的值，表现大规模企业与小规模企业对于区域行业重心影响程度的差异。重心轨迹由各年的重心连接而成，可体现行业重心随时间的迁移，展现行业的发展趋势。

### 标准差椭圆分析

标准差椭圆这一定义最早由Lefever于1926年提出[1]，可用于衡量点数据集的空间范围与趋势。之后，也有许多研究在此基础上对确定标准差椭圆的公式进行改良，其中较为重要的改进为1971年Yuill[39]所提出的加权标准差椭圆。其一般式如下：

+ =s **（2.2）**

其中，为椭圆在X，Y轴上的标准差，分别代表数据集的方向和范围，s为置信度，可根据需求及数据量来查询卡方概率表进行设置。在图上确定一个标准差椭圆，需要得到椭圆的中心，长半轴长，短半轴长及椭圆的方向。其中椭圆的中心，可以直接使用数据集的中心，而其他值的计算公式如下：

**（2.3）**

**（2.4）**

**（2.5）**

其中θ指椭圆长轴与坐标轴X轴的夹角，指数据集中所有坐标点的经纬度平均值，为点i的权重，本文中可以利用企业注册资金等衡量企业规模的值来确定。根据标准差，及置信度值s可进而确定标准差椭圆长轴和短轴的具体长度。置信度s本身对应覆盖点数的百分比，因此不影响椭圆形状，而只影响其大小。

对于计算所得的标准差椭圆而言，其长短半轴代表其方向，面积代表点集中点的分布密度。短轴代表数据分布的范围，而长轴代表趋势，两轴长度越接近，说明点集不具有明显分布趋势。在s确定的情况下，面积越大，对应说明该点集分布越广、越均匀，面积越小，说明分布越集中。

## 本章小结

时空演化分析是地理信息科学中空间分析的下一个发展方向，这类研究结合了时空两个维度，动态地表现了数据空间特征随时间变化的过程。而对于行业研究而言，时空演化分析主要利用的是不同年份间空间分析结果的比较，其中包含有聚类的比较、区域统计信息的比较等方案。在结合可获取的数据和目标研究的粒度考虑之后，本文选用的是基于点模式分析的时空演化分析，并具体从中选取了重心迁移轨迹和标准差椭圆这两种方法来完成本次分析。

# 行业时空演化的Web可视化框架设计

由于重心轨迹及标准差椭圆的计算结果本身难以比较与观察，需要选择合适的方法进行可视化才能真正起到分析效果。本文所选取的可视化平台为Web平台，具有便于分享，功能添加方便及交互容易等优点。本章对现有的常见Web可视化技术进行了概述，并详细介绍了本文所建立的可视化平台的框架及各层设计方法。

## Web可视化技术综述

数据可视化能形象地将原始数据或分析结果以图像形式展现出来，在数据分析中可以起到提供分析思路及展现信息等作用。在可视化方式的选择上，应当同时考虑信息传递的能力、信息获取的便捷性与美观性，这三者相辅相成，缺一不可。数据可视化的方式有许多，从最传统的纸质图表，单个电子图表到最后的电子可视化系统。这些可视化方法逐步使获取可视化结果、传递信息变得更加便捷简单，同时也使得信息表现得更加清晰。

而电子可视化系统又分为基于软件的可视化系统和基于Web的可视化系统。基于软件的可视化系统大多性能较高，可以处理自定义的数据，但是需要安装获取，不易于共享。而基于Web的可视化系统则是具有成本低、便于共享等优点，用户通过一个网址即可访问，降低了时间成本与对电脑性能的要求。而在自定义数据这方面，基于Web的可视化系统由于依赖网络传输的关系，计算与可视化效率受到了限制，有时会不如基于软件的可视化方法。而为解决这些问题，一般采用的方法是利用服务器上的数据库存储计算结果或是利用较高性能的计算平台来减少用户等待时间。

### Web框架

近年来随着互联网的普及以及大数据这一概念的发展，对于Web可视化的需求逐渐增多，也促进了Web可视化技术的发展。在移动设备发展后，为适应不同配置、平台的浏览器，使用开源前端框架来解决兼容问题成为了趋势。每年都会有新的前端框架产生，其中较为流行的前端框架有Node.js、React、Vue.js、Angular.js等。

React[40]是这之中较为特殊的一个，它本身只对应模型-视图-控制器（model-view-controller, MVC）模型中view这一层的构建，但是该框架对于组件同步和DOM控制的高性能还是使其成为了热门框架之一。利用React所构建的组件具有极强的复用性，可以很好地应用在大型项目的开发中，同时该框架中支持单向响应数据流，使得数据与组件等的绑定更为方便轻易。

Vue.js与Angular.js属于模型-视图-视图模型（model-view-viewmodel，MVVM）模式框架模式。其中view及viewmodel两层连接紧密，任一层更新可以与对方层同步。相较而言Vue.js更为轻量级，学习成本也较低，而Angularjs可自动触发脏检查，且可通过事件监听，进行数据更新，对于大型项目有更好的支持。

与上述框架相比，jQuery[41]框架属于最早流行的框架之一，仍属于MVC模式。但是该框架具有开创性的意义，对于常用的JavaScript方法及DOM操作进行合理封装，也对基于ajax的数据读取提供了有效支持。其以其轻量级，插件多，功能调用方便，兼容性好等优点成为了基于PC平台浏览器构建小型项目最常采用的框架之一。而对于本文而言，该框架足以支持可视化方案中的所有功能，且其所支持的大量插件可以很大程度上帮助本可视化系统构建。

### Web可视化开源库

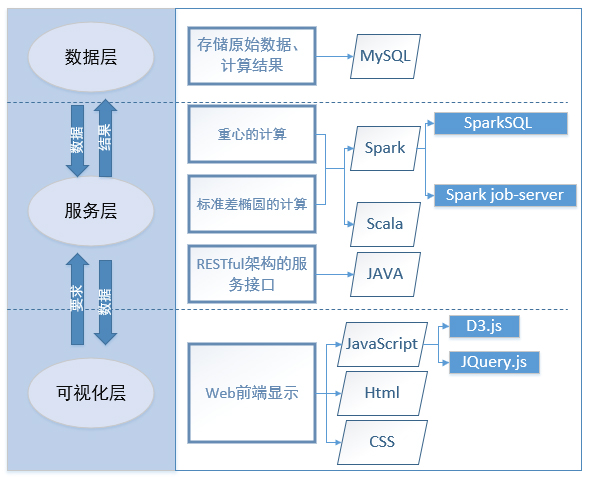
除去开源框架之外，越来越多的基于JavaScript的可视化库也在流行。其中包含有以图表库为主，提供大量个性化图表及功能选择的ECharts[42]，针对移动设备，高性能的leaflet[43]，可轻松创建动画、图像的Processing[44]以及对于各类型数据的处理都支持且DOM操作便利的D3.js[45]等。在众多的可视化图表库中，D3.js因具有数据驱动，实例丰富，支持交互功能及动画效果多等优点而流行。除此之外，D3.js还具有如下优点：

1. 可方便使用SVG相关操作。对于数据可视化展示而言，D3可帮助用户灵活利用SVG画布与传统HTML、CSS配置，任意画出目标矢量图形，而无需对SVG有过多了解；
2. 具有极高灵活性。相较于提供已成型图表的ECharts等可视化库，使用D3编写网页可以充分利用其基础API达到各类个性化的效果，而无需担心受限于已有图表特性或布局。
3. 数据驱动可视化。D3可方便的将数据与DOM节点绑定，建立对应可视化或交互，同时对于各格式数据文件读取都有对应的API，十分便捷。

## 可视化系统架构设计

本文采用的可视化架构为B/S架构，将可视化系统分为数据层、服务层与可视化层。本可视化系统具体框架结构及使用技术如图3.1所示，各层基本功能为：

1. 数据层：数据层作用为数据管理。该层包含有两个数据库，即注册信息数据库和结果数据库。前者包含所有企业记录的详细信息，后者存储计算结果，即不同空间单位和行业的重心和标准差椭圆参数。
2. 服务层：RESTful Web服务负责可视化层和数据层之间的通信。服务从数据层查询和访问数据，并将计算结果发送到可视化层。如果计算结果未预先计算，则接口将调用Spark执行计算。同时，使用Spark的嵌入式数据模型DataFrame，将原始企业注册数据和计算结果都缓存到Spark中，借助Spark SQL实现并行和内存计算。
3. Web可视化层：该层用作框架的客户端和图形用户界面（GUI）。它包含两个基本视图：地图视图和坐标系视图，为用户提供交互和可视化功能，以呈现重心和标准差椭圆的移动路线。为了获得可视化所需的数据，该层根据用户的要求隐式地向服务层发送HTTP GET和POST请求。

图**3.1** 面向行业时空演化分析的Web可视化系统结构及相关技术图

## 数据库设计

本层主要存储原始的企业注册数据及服务层的计算结果，用到的主要技术为MySQL。MySQL是一个常用的关系型数据库管理系统，目前属于Oracle旗下公司，具有体积小、速度快、可承受大量的并发访问等特点。在这一层中，建立有两个数据库，注册信息数据库与分析结果数据库。

### 注册信息数据库

注册信息数据库内存储企业注册信息的原始数据，包含多个数据表，如法人表、企业表、工商管理所表等。但本文研究只需要利用其中企业表即可，该表包含有各企业所属行业小类，经纬度，所属城市，市区，注册时间，注册资金等信息。通过对原始企业数据表进行相关数据预处理，可将行业小类划分为行业大类，将注册时间转换为年份属性，利用经纬度获取分区信息等。最终得到的企业注册信息表内数据结构如下表3.1所示。

表**3.1** 注册信息表样例

|  |  |
| --- | --- |
| F\_INDUSTRY | 批发和零售业 |
| year | 1995 |
| F\_WGS\_LNG | 108.239966 |
| F\_WGS\_LAT | 31.369264 |
| county | 渝北区 |

### 分析结果数据库

本文中实践过程中的分析结果数据库共存储有两张表，市级结果表（citydata）与区级结果表(countydata)。在不同研究对象的类似实验中，还可能会有省级计算结果表。两表结构类似，包括有区域内各行业各年份的重心（经度及纬度），椭圆沿X轴，Y轴的标准差，椭圆长轴与X轴夹角等信息。这里以县级表结构（表3.2）为例进行展示。结果数据由服务层计算所得，这些信息可在之后的可视化过程中确定重心轨迹与标准差椭圆。

表**3.2** 区级分析结果表样例

|  |  |
| --- | --- |
| IndusAndYearAndArea | 文化、体育和娱乐业2001江津区 |
| count | 5 |
| SigmaX | 0.080233 |
| SigmaY | 0.087744 |
| Lat | 29.22386 |
| Lng | 106.3172 |
| Theta | -0.65753 |
| year | 2001 |
| industries | 文化、体育和娱乐业 |
| county | 江津区 |

## 服务接口设计

服务层主要包含两部分，一部分为基于Spark平台的重心及标准差椭圆计算，另一部分为RESTful风格的接口，帮助数据在可视化层及服务层进行传输。服务层与数据层之间的数据传输主要依赖的是Spark SQL中自带的方法。对于基于Spark平台的计算服务设计将在第4章进行详细介绍，本节将主要介绍本文可视化设计方案中的服务接口设计。

### RESTful架构风格介绍

表述性状态转移（Representational State Transfer, REST）架构是目前比较成熟的一套互联网应用程序的API设计理论，最早于2000年由Roy Fielding[46]所提出。客户端通过四个HTTP动词，对服务器端资源进行调用，每一种URL代表一种资源的获取。符合这一架构风格的接口称为RESTful接口。这样的接口可以充当连接作用，通过唯一标识URL来识别资源。RESTful架构具有以下设计要求：

1. 接口统一：对于不同的资源和不同的访问者，其访问资源的接口都是一致的，必须使用规定的四个动词对应的方法来进行数据的增删查改。而各种资源所对应的唯一URL需要包含有资源的路径及相关参数指令。
2. 分层封装：系统各部分组件封装服务，相互直接只有接口暴露在外，提升了系统的可扩展性与实用性。
3. 无状态：即单次查询或更改的所有请求信息应当包含在请求本身中，而不会受其他因素影响，即不受前后请求影响。这样能尽可能的降低服务器压力，加快运行效率。

四个HTTP动词GET、PUT、DELETE、POST分别对应只读访问、创建新资源、删除原有资源和更新现有资源功能。

### Spark计算服务介绍

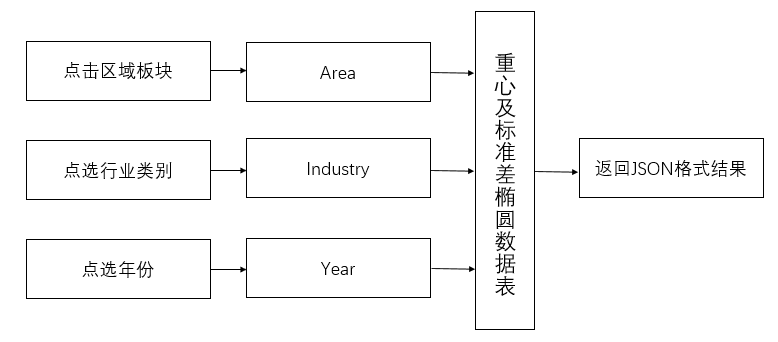
本系统的数据计算采用的是基于Apache Spark的分布式计算，由于Spark平台本身没有提供可直接供前端调用的返回计算结果的RESTful接口，在开发中还需要添加一个可提供RESTful接口来提交、管理Spark作业的开源项目包Spark-job-server。通过Spark-job-server，前端可以直接向Spark平台传递参数，运行相关代码包。同时这一部件也支持同步和异步的job API，既可以处理低延时的任务也可以进行较长时间的计算后传递结果。Spark-job-server类似于可插拔的管理工具，其部署独立于Spark集群，安装与运行时无需对Spark集群进行配置。

本管理工具为Spark计算任务管理提供了许多实用的API，包含有利用GET、POST、DELETE、PUT对Spark Context进行查找、创建、删除以及重建的方法，以及利用GET与POST对计算任务的启动和结果进行查看管理的方法。其中本次用到的主要方法为POST/jobs，利用该接口可向Apache Spark计算平台传递jar包所需的必要参数并运行jar包，同时利用参数sync可控制计算是否同步，即是否等待计算结束再返回状态与结果。在计算时间较短时，可以提前建立常驻context并选取同步获取计算结果，来达到实时计算效果。

### 本文服务接口设计

本可视化系统中用到的接口功能主要可分为，自可视化层向结果数据库请求数据和自可视化层向Spark计算平台提交计算任务获取结果这两种。前者包含有读取市级、县级企业数和重心、标准差椭圆参数等功能，后者为传递带有年份、行业的参数的请求，并进行实时计算获取结果的功能。

向数据库请求数据的接口设计原理为，将用户在可视化界面的操作对应转换为相应SQL语句。本文中连接数据库的接口共有三个，分别为：（1）根据行业、年份属性从县级表读取各县企业数；（2）根据行业、地名从县级表读取各年企业数，重心，标准差椭圆相关参数（3）根据行业从市级表读取各年企业总数，重心，标准差椭圆相关参数。其中输入参数的获取与对应操作设计如图3.2所示。

图3.2 可视化框架接口设计思路图

以“获取重庆市教育业重心及标准差椭圆信息”这一要求为例，其对应URL为：http://localhost:8084/getData/CitySDE?Industry=教育。其中http:// localhost:8084为本地所使用服务器接口地址，/getData/CitySDE为该接口路径，Industry=教育为输入参数。该URL对应返回json形式查询结果格式如下所示：

{"Result":[{"Lng":"106.4833873000","SigmaY":"0.07862449766681798",

"count":"10","SigmaX":"0.07671480377051708",

"industries":"教育","Lat":"29.6012178000",

"year":"2001","Theta":"-0.726249404174515"}, …]}

其中Lng，Lat对应重心经纬度，SigmaX，SigmaY，Theta对应确定标准差椭圆的三值，year，industries对应所选年份、行业，count指该地区该年企业总数。

向Spark计算平台提交计算任务相关的接口对应功能为，向Spark平台传递行业、年份及区域参数，并以这些输入参数运行已上传至服务器的jar包，等待获取同步运行结果。该接口的调用方式为点击“实时计算模式”按键，之后的行业、年份选择与非实时计算相同。本文主要实现了重心的实时计算，考虑到计算效率问题，标准差椭圆仍采用的数据库读取方式。

本文方案中所有接口方法具体参数及路径如表3.3所示.

表**3.3** 接口参数汇总表

|  |  |  |  |
| --- | --- | --- | --- |
| 接口路径 | 操作类别 | 输入参数 | 输出参数 |
| /getData/CityCount | GET | Year, Industry | count, year, industries, county |
| /getData/CitySDE | GET | Industry | count, year, industries, Lng, Lat, Theta, SigmaX, SigmaY |
| /getData/CountySDE | GET | Area, Industry | count, year, industries, Lng, Lat, Theta, SigmaX, SigmaY, county |
| /getData/ComputeCenter | POST | Area, Industry | count, year, industries, county |

## 可视化界面设计

本可视化界面设计目标展现不同空间单元内的重心和标准差椭圆。这一视图应有三个主要特点：（1）利用多尺度的分区密度图展现企业数这一变量；（2）利用交互式功能以避免不感兴趣的信息被显示和突出所需的信息；（3）视图中采用动画，有助于表现和强调空间特征随时间的变化。

### 可视化界面功能设计

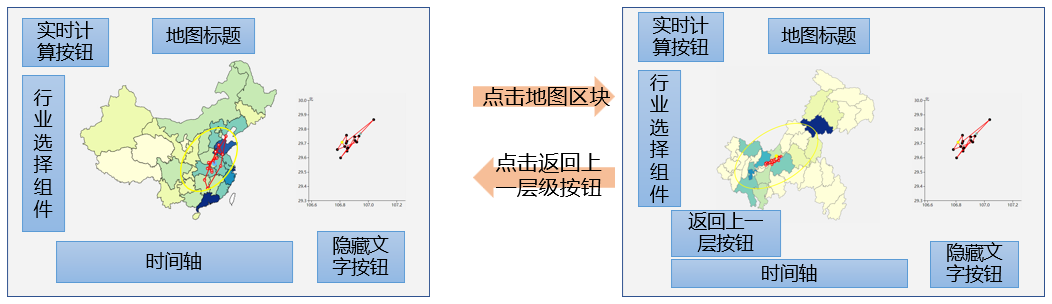
本可视化界面应当要清晰展现用户所关心的行业、年份的重心轨迹、标准差椭圆与分区密度图。在这样的需求下，本文使用动画而不是静态的箭头来表示重心移动方向，来避免重心间的重叠妨碍观察，并以此吸引用户的注意力。另一方面，本可视化界面还应当提供足够的交互功能，以支持用户进行相关属性选择或避免其它多余信息的显示。根据本次可视化的需求以及对于清晰展现信息的要求，本界面需要提供如下功能：

1. 行业、年份、行政级别选择；
2. 动画显示重心轨迹；
3. 显示所选年份的标准差椭圆及分区密度图；
4. 隐藏可能遮挡图形的文字；
5. 提供额外的文字信息；
6. 放大局部界面。

本文选择利用交互组件，如文字标签及图形等来帮助用户进行直观的属性选择，利用按钮等来完成其他交互操作，如隐藏文字，返回上一层级等。与这些组件的交互全部由鼠标完成，这一特点使得用户操作更加简单、方便。本可视化方案采用时间轴与播放、暂停按钮来对随年份的动画效果进行控制。默认情况下，重心、标准差椭圆及分区密度图随时间轴上年份逐年变化，但用户也可以点击任意年份跳转选择，或者单击播放、暂停按钮停止动画。

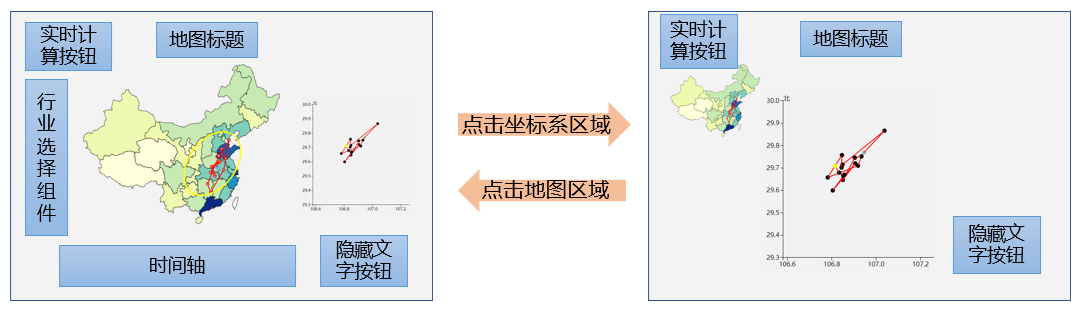
这些交互功能与动画的具体实现依赖于jQuery与D3.js相关响应函数、组件与动画函数。可视化界面应当能提供各行政区块的详细信息，并能支持用户进入下一行政层级。在本次可视化设计中，前者依靠mouseover事件响应实现，后者依赖click事件响应实现。地图上代表重心轨迹与标准差椭圆的矢量图形以动画形式显示，这一过程可利用setTimeout()建立计时器，自动间隔时间进行绘制来实现，相应绘制函数需以所选年份与所选行业为输入参数。

### 可视化界面布局设计

可视化界面由三个主要部分组成：地图部分、坐标轴部分与各类交互组件。其主要页面布局，及不同层级界面切换方法示意图如图3.3所示。

图**3.3** 时空演化分析可视化系统界面设计图

地图部分的基础为基于行政区划的分区密度图，其中所有行政区块都根据该地区所选行业的企业总数进行着色，以显示该行业发展水平的地区差异。这一地图的实现，首先需要利用D3.js中自带有的一系列投影相关函数，并读取服务器上的GeoJSON文件绘制行政底图。之后可视化层利用Ajax操作从接口读取所选条件下的各区域企业数，按照最大最小值依比例分段，为各区块着色而成。根据研究对象不同，分区密度图中有不同的地图级别，可能包含有国家级、省级、市级和区级。

坐标系部分有着与地图部分同步的重心轨迹矢量图形，用户可从中获取重心的详细信息。用户可以点击图表上的任何位置使该视图扩展到全屏模式，此功能可以帮助用户将注意力集中在重心轨迹迁移上。同时，为提供地图的概况，在坐标系放大时，地图部分的鹰眼图会显示在界面左上角，具体放大功能的调用方法与效果如图3.4所示。

图**3.4** 坐标系部分界面放大效果图

交互组件包含有行业选择标签，层级选择按钮和时间轴，使用户能够在不同的地图级别，空间单位，年份和感兴趣的行业之间切换。其中行业选择标签中各行业对应文字及背景块进行了绑定，点击两者之一会同时发生色彩变化，表明已被选中。层级选择按钮及本系统中其它按钮均利用的是jQuery自带组件按钮。时间轴可控制系统中所有重心、标准差椭圆及分区密度图按照年份顺序显示。

当用户将鼠标悬停在代表某一年重心的标记上时，该页面上将显示提供更多行业信息的提示框，提供如行业，经度，纬度以及该行业当年的企业总数等信息。由于有些参数选择下的企业重心随年份没有太大变化，它们的年份标签可能会与其他数字元素或标签重叠。为使重心移动路线图的整体模式更加清晰，此视图提供了通过相应按钮来隐藏和显示年份标签的功能。

## 本章小结

本次重庆市时空演化分析结果将采用基于Web的可视化框架进行可视化，该框架主要包含三层：数据层、服务层与web可视化层。其中数据层包含存储有原始数据和分析结果的两个数据库，服务层包含向数据库请求结果的接口以及向Spark计算平台提交任务的接口，Web可视化层包含基于地图的可视化和基于坐标系的可视化这两个模块。总体而言，本可视化方案提供一个具有交互性、清晰的行业时空演化可视化结果，有利于帮助用户进行后续分析。

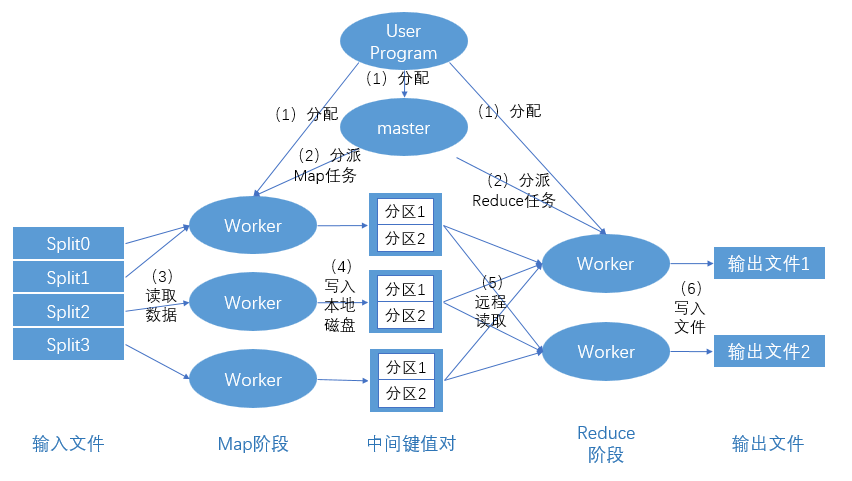
# 基于Spark的时空演化分析算法并行化

由于企业注册数据的数据量较大且处于随时更新的状态，普通计算耗时过长，另外若在可视化系统中要支持实时的重心、标准差椭圆计算，在短时间内对于大量数据的计算是不可避免的，本文采用了并行计算的方法来实现重心轨迹和标准差椭圆的相关参数计算。本章首先对现有并行计算平台发展进行了综述与讨论，并着重介绍了本文将要使用的平台Apache Spark的相关特点与工作原理。之后对于所采用的时空演化分析方法的并行算法设计进行了详细的叙述。

## 并行计算技术概述

并行计算是与平行计算相对的，一次在多平台、多核上执行多个指令的方法，目的是充分利用计算资源，提高计算效率。并行计算包含时间、空间两种并行方向，其中空间上的并行指的是多台处理器的协作共同完成任务，如多核计算与并行化平台计算等。并行计算实际为单机在横向上的扩展，通过分布式的方法来突破单台电脑硬件发展所无法达到的效率瓶颈。

2003 年美国 Google 公司提出了 MapReduce 框架作为并行计算框架[47]，之后被各类并行化计算方法所采用。MapReduce包含Map（映射）与Reduce（归约）函数，在计算过程中将海量数据切片处理分发给不同机器，属于分布式离线计算框架。计算过程主要包括Map、Shuffle、Ruduce三个阶段。具体而言，一个Map任务会被分发到任一计算节点执行，多个Map任务并行处理。Map过程将数据转换为key-value形式，其处理后的输出数据会在shuffle过程被分割至多个分区，即partition内。之后对于分区内数据，以键值排序，交至Reduce任务处理。多个Reduce任务并行处理，将key-value形式的结果迭代输出。MapReduce过程各阶段具体流程示意如图4.1所示。

图**4.1** MapReduce执行过程图

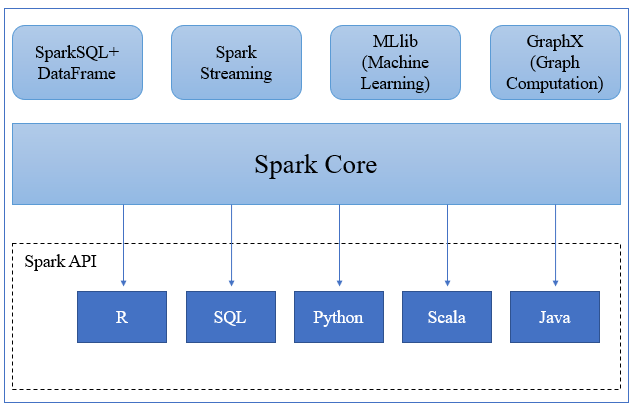
最早流行的分布式计算框架Apache Hadoop的设计思路就是来自于此。Hadoop主要包含HDFS与YARN这两个组件，前者用于在集群上存储不同类型的文件，方便计算中的数据读写，后者是Hadoop的处理单元支持并行处理数据。Hadoop的开源特性使其成为了之后的大多数分布式计算系统的基础或标准。

类似的分布式计算框架有由Nokia研究中心提出的DisCo[48],由斯坦福大学提出的Phoenix[49],由香港科技大学提出的Mars[50]等。这些框架都仍是基于MapReduce框架所设计，具有共同的在数据量增大时对于内存和CPU的使用率迅速增大的特点。之后出现的Apache Spark[2]在一定程度上解决了这一问题。这一由加州大学伯克利分校开发的分布式计算平台在原有Hadoop的基础上进行了改良，采用内存来存储数据，无需从外部存储介质中再读取数据，同时也不再利用MapReduce进行数据处理。Spark运算速度最高可达到Hadoop的100倍。而Spark在被提出后，另一个受到追捧的原因是，其支持多种语言如Scala, Java, Python等，自带多种API，同时可利用SparkSQL组件直接使用SQL语句，这些都大幅提升了Spark的易用性。

## Spark框架介绍

考虑到效率、易用性及实时处理需求，本文采用Apache Spark作为分布式计算及并行算法的实现平台。Spark采用类似MapReduce的并行计算方式，但由于其利用了基于内存的DAG（有向无环图）计算引擎，根据其官网[40]所提供对比，其计算速度可以较Hadoop MapReduce有近百倍提升。

### Spark生态系统介绍

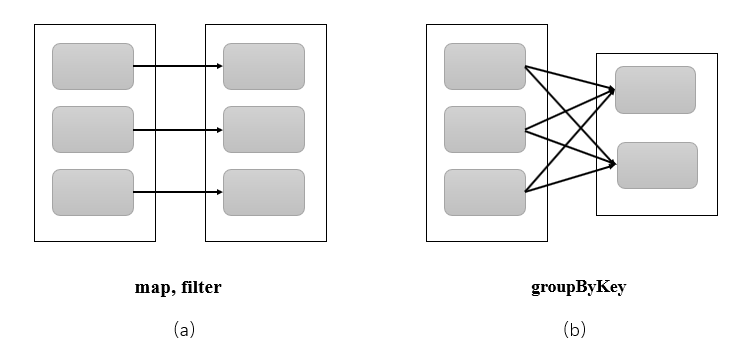
类似于Hadoop生态系统，Spark也有其相应的生态系统（如图4.2所示），包含若干相关组件，包含有用于处理、连接数据库数据的Spark SQL，处理流数据的Spark Streaming，提供机器学习相关计算方法的MLib及适用于图计算的GraphX。考虑到分析方法的特点和原始数据的组织方式，本文将使用Spark核心部分与Spark SQL[51]组件进行并行算法实现。

图**4.2** Spark生态系统图

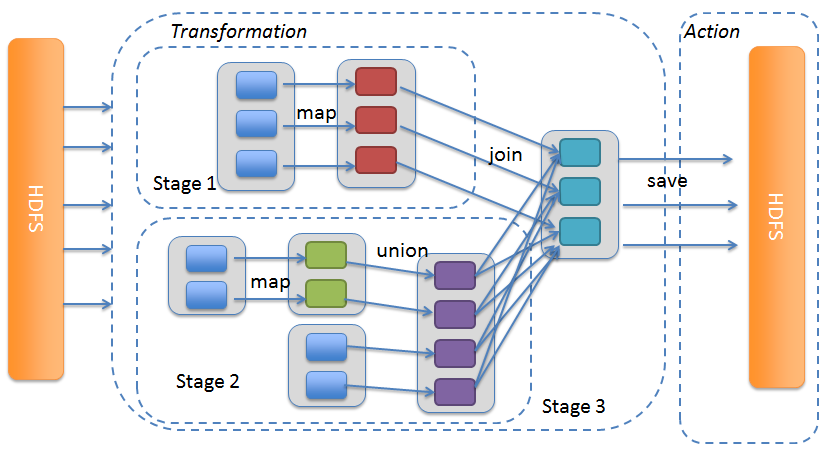
由于本文的可视化框架将原始注册数据存储在MySQL数据库中，所以可使用一个名为Spark SQL的模块直接使用SQL语句查询数据库中的结构化数据，并将查询结果存储至分布式内存中。Spark SQL与以前的系统相比有三个主要优势：（1）它提供了关系数据库和过程处理之间更紧密的连接，使Scala等编程语言更容易处理数据库内的数据; （2）基于Spark框架，每次查询花费的时间要少得多。（3）Spark SQL可利用SQL语句直接从关系数据库中请求感兴趣的企业的原始数据，以这种方式获取的数据将以结构化的存储方式存储在DataFrame中。 DataFrame是Spark中使用的数据模型，它以命名列来组织数据。这样的形式有利于计算过程中用MySQL语句和其他流行的编程语言（例如Scala，python）来对数据进行操作。同时，DataFrame也可以很容易地转换成Spark核心数据结构RDD，可以支持在Spark中进行更多的数据操作。

### Spark工作原理

Spark计算方法与Hadoop中的分布式计算MapReduce模块相对应，前者与后者类似，但速度更快且能提供更多功能。Spark计算中的核心抽象概念为弹性分布式数据集(Resilient Distributed Dataset, RDD) [52]，是Spark中数据操作的基本单位。并行计算过程中一个RDD会被分为多个partition，分配至各节点上进行运算。这样会导致无法对特定某一条数据进行处理，但是适合用于对大量类似结构数据进行同一操作，因此与并行计算概念相适应。

RDD具有多种特性。其一是不变性，每次操作后所得到的都是新的RDD，而不会对原有RDD造成影响。反过来，每一个新的RDD也一定是由已有RDD所得，因此而使RDD具有序列性，构造出一个RDD间相互依赖的有向无环图（DAG）。每一个有向无环图即对应Spark收到的一个Job。同时这一特点还使得分区错误维护变得方便，对于出错的分区可直接根据原有血统关系重新计算，而无需检查点维护。这些特性都确保了Spark计算集群的稳定性和高性能。

图**4.3** RDD间宽窄依赖示意图

Spark在具体运行时会把Stage包装成Job来提交，可能会存在有一个父Stage包含多个子Stage的情况。 每个由多个RDD构成的有向无环图会由SparkContext生成一个Job，之后划分阶段（stage），分区（partition），再在集群上运行。RDD之间存在有依赖，其中若新旧RDD分区关系为多对多的叫做宽依赖（如图4.3(b) 所示），是划分stage的标准。宽依赖中由于存在多个分区的依赖，需要等待所有节点的分区计算完毕，才能进行下一步处理，会造成延迟，这一过程也称为shuffle。分区作为划分task的标准，若每个新RDD分区对应单独的一个旧RDD分区，则称为窄依赖（如图4.3(a) 所示），可直接进行。考虑到这一特性，在效率优化时应当尽量避免shuffle操作，而采用窄依赖操作。图4.4为一个Job在Spark上运行时的Stage划分示意图，从中可以看出Stage 2中的map, union操作均为窄依赖操作，则这两步操作可以形成一个流水线操作，每个Partition在首先进行了map操作之后可以直接独立进union操作，而无需等待整个RDD计算完成。与之相反的是join操作，该操作为宽依赖，因此Spark会以这个操作为边界，将前后划分为不同的Stage。

图**4.4** Spark中的Stage划分示意图[55]

RDD有类似于MapReduce的两类操作，转换（Transformation）和动作(Action)。转换用于定义、生成新RDD，操作前后数据形式均为RDD，对应方法有map, filter, union, join, groupByKey等。动作用于返回一个结果至driver，数据形式从RDD转化为其它形式，对应方法有collect, reduce, count等。RDD常用基本操作的作用如表4.1所示。

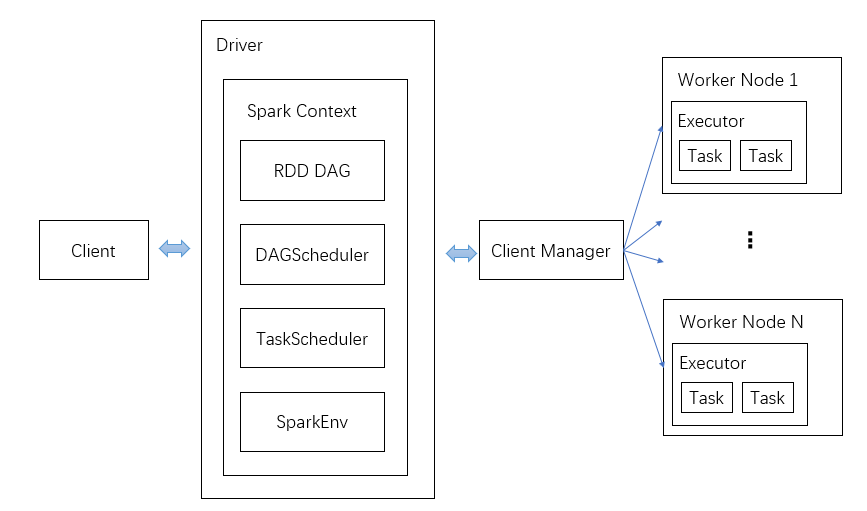
表**4.1** RDD基本操作表

|  |  |
| --- | --- |
| 操作名 | 作用 |
| map | 对RDD中每个元素进行操作，并返回一个新的RDD |
| filter | 对原有RDD依条件进行过滤，返回RDD |
| union | 将两个RDD联合，生成新的RDD |
| join | 两RDD依照key值配对联合 |
| groupByKey | 依照key值将RDD进行分组操作，处理后生成新RDD |
| collect | 返回RDD内所有条目 |
| reduce | 并行整合所有RDD数据，进行累加或类乘等操作。 |

在Hadoop中，只提供有Map与Reduce两个操作，且一个Job只包含这两个阶段，因此对于一个计算任务，往往需要生成大量的job。而相比较而言，Spark中每个RDD都是对数据集的一个抽象，提供有大量基本操作，同时一个Job中可以包含多个RDD，因此在运算效率上Spark具有显著优势。

### Spark部署方式

Spark的工作架构是采用的基本的Master-Slave结构，即一台机器作为Master控制整个集群，而集群中可以添加负责工作的Worker节点，各节点上可布置一个或多个实际进行Task处理的Executor。整个Spark集群包括客户端（Client），Master节点和Worker节点。而对于单个Spark应用程序而言，其需要的有一个位于客户端的驱动器（driver）和多个布置在工作节点(worker)上的Executor。驱动器需要通过Spark Context与Spark集群进行交互，一个工作节点上可以有一个或多个Executor，每个Executor可处理多个Task。其具体计算结构如图4.5所示。

图**4.5** Spark计算平台结构图

Apache Spark框架的运算基于集群，总共有三种部署方式。可以选择建立专属Spark的集群也就是standalone模式，也可以直接在Mesos或者YARN上直接部署运行。由于原本没有已搭建的集群，本文选取的部署方式为standalone模式，因此本文以Standalone模式为例，介绍Spark具体运行流程，其具体流程如下：

1. Spark客户端通过SparkContext连接Master，并进行本应用程序的配置，之后向集群申请资源；
2. SparkContext启动Standalone运行后台；
3. 分解应用程序中的操作，生成一个或多个Job；
4. DAGSchedule会将所有Job划分为一个或多个Stage，每个Stage对应一组操作相同但对象不同的Task；
5. 将Stage中的Task经由Task Scheduler发送到各个Executor进行运算；
6. 运行结束后释放资源。

## 重心轨迹及标准差椭圆并行化算法

### 算法设计与实现

根据本文第2.2节中所提到的方法及相关公式，可设计其并行化算法。由于标准差椭圆计算中需要用到重心数据，因此应首先完成重心计算。重心计算实际为符合条件的企业经纬度平均值。此条件在非实时的计算中，为年份、行业与地区名，而在实时计算中可以为年份、行业与数个地区的组合。在此基础上继续进行标准差椭圆计算。

确定标准差椭圆需要确定其中心，在X方向、Y方向标准差，，椭圆绕X轴旋转角度θ。其中椭圆中心即为上一步已经得出的重心，其他参数计算步骤如下：

1. 计算各企业经纬度到该行业注册年份重心的差值x,y;
2. 利用各企业差值计算xy,x2,y2;
3. 计算选择条件下xy,x2,y2的累积；
4. 根据步骤3中结果值计算tanθ；
5. 利用tanθ得到cosθ及sinθ值；
6. 根据步骤1中的差值x,y及步骤5中的cosθ，sinθ计算，。

本并行化算法实现依赖于Apache Spark计算框架与Scala语言。并行计算中，将数据依照年份与行业进行了分组，每组数据被分别派送到不同Worker上进行后续计算。该过程的实现方案伪代码如表4.2：

**表4.2 基于Apache Spark的重心与标准差椭圆计算伪代码**

|  |  |  |
| --- | --- | --- |
| 输入参数：Industry, Year, (Area) | | |
| 输出参数：Longitude, Latitude, Theta, SigmaX, SigmaY | | |
| 1 | conf 🡨 SparkConf() | //Spark的相关属性配置 |
| 2 | sc 🡨 SparkContext(conf) |  |
| 3 | sqlContext🡨 SQLContext(sc) | // 连接数据库的配置 |
| 4 | data🡨sqlContext.read.jdbc(url, TableName,properties) | //从关系数据库中读取数据 |
| 5 | gravityCenter🡨[data.groupby(year).mean(latitude), data.groupby(year).mean(longitude)] | //计算重心 |
| 6 | infoWithCenter🡨data.join(gravityCenter) | //利用year属性连接重心与企业注册信息 |
| 7 | TempResult🡨infoWithCenter.groupby(year). map{ | //利用Map过程计算中间参数 |
| 8 | x🡨ComputeDeviation(latitude) | //计算各企业经纬度到对应重心的差 |
| 9 | y🡨ComputeDeviation (longtitude) |  |
| 10 | xSquare🡨x\*x | //计算 x2, y2, xy等值 |
| 11 | ySquare🡨y\*y |  |
| 12 | xy🡨x\*y} |  |
| 13 | SumOfTempResult🡨 [Sums(xy)，  Sum (xSquare), Sum (ySqaure)] | // 计算选择条件下xy,x2,y2的总和 |
| 14 | Theta🡨SumOfTempResult.groupby(yearandindus). map{} | //计算各行业各年标准差椭圆的θ值 |
| 15 | SigmaXandY🡨ComputeStandradDeviation(x, y,Theta) | //利用x,y,Theta计算标准差值 |

### 并行算法优化

在初步设计算法后，对于其具体实现，应当充分考虑语言中不同方法效率，并加以选择才能真正达到效率最优。优化方法主要有两个方向，一是从选用的RDD操作和函数入手。如在实验过程中，最初尝试使用DataFrame操作来选取属性列，之后改为了SparkSQL所支持的直接使用SQL语句进行读取，更改之后整体计算时间明显缩短。这是由于后者避免了数据读取入DataFrame再进行操作这一过程，直接在数据库读取时进行，加快效率。而计算效率优化的另一方向，则是通过改变集群运算中的相关配置，如分区数、使用计算核数等。

综合上述优化思路和实际算法情况，本次算法优化过程中改进的地方有：

1. 避免join、reducebykey等shuffle类算子，及将分布在各节点上的数据进行聚合的操作；
2. Filter后进行coalesce操作，减少RDD的分组数。
3. 尽可能复用RDD。
4. 利用saprk.default.parallelism配置设置每个stage的task数，可以在节点数及总计算核数较多时，充分利用各节点。

在进行了上述优化后，同规模数据（2,000,000条）在相同集群配置下，进行重心与标准椭圆计算的处理时间从最初的111秒降低为74秒，效率有了显著提高。

## 本章小结

Apache Spark是现如今较为流行的数据并行化计算平台，也是本文所选取的并行化实现平台。对于重心轨迹计算以及标准差椭圆参数计算而言，并行化方法主要是将数据依照年份、行业进行分组生成相应的Job与Task，再将Task组分发到各Executor上同步地进行计算，最终将结果汇总输出。这一过程结束后，相应结果可以存储在数据库中，也可以以json文件形式传递至可视化系统，直接进行可视化操作。本文结合分析方法特点及Apache Spark工作原理设计了重心及标准差椭圆的并行计算方法，并根据Spark中操作特性进行计算效率优化。

# 实验与结果分析

本文以重庆市为例，进行行业时空演化分析。本章包括可视化系统效果展示、并行化计算效率对比实验与对于行业时空演化规律的形成原因的讨论。本文针对重庆市及其下属分区两级进行分析与可视化，并着重讨论了渝北区及九龙坡区的典型行业时空演化特征。

## 实验数据

### 数据说明

本文的实验对象为重庆市，所利用的实验数据为重庆市1950-2016年的企业注册数据。重庆市辖26个市辖区、8个县与4个自治县，共合计38个区级行政区划单位。重庆市原始企业注册数据中包含有企业法人、企业编号、企业类型、行业类别、具体地址、分属管理机构、所属片区、注册时间、经纬度（包括WGS与百度两种）与经营范围等信息。具体样例可见附表1。

### 数据预处理

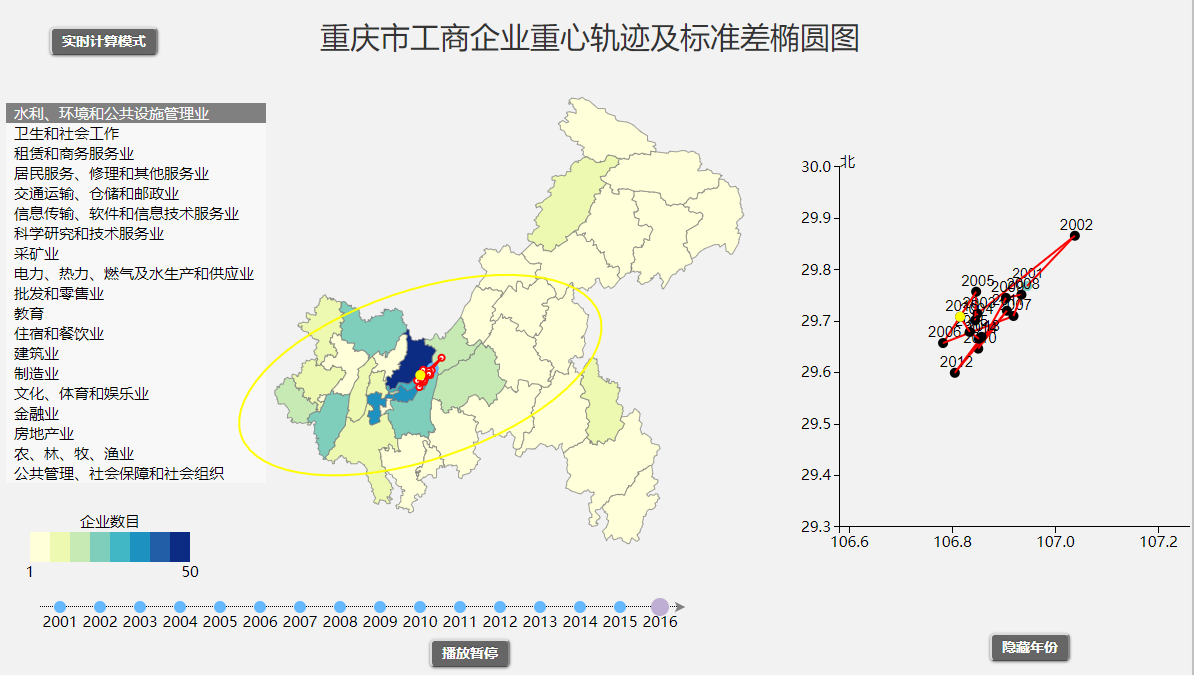
在利用原始企业注册数据进行时空演化分析前，需根据中国国家经济行业分类标准（GB / T4754-1994）[53]，将所有行业小类进行归类，统一到最终的19个行业大类。最终分类得到的具体大类种类为水利、环境和公共设施管理业，卫生和社会工作，租赁和商务服务业，居民服务、修理和其他服务业，交通运输、仓储和邮政业，信息传输、软件和信息技术服务业，科学研究和技术服务业，采矿业，电力、热力、燃气及水生产和供应业，批发和零售业，教育，住宿和餐饮业，建筑业，制造业，文化、体育和娱乐业，金融业，房地产业，农、林、牧、渔业以及公共管理、社会保障和社会组织。

由于分析目标为城市，最终分析中应包含的层级有市级和区级。但原始数据中并不包含区级信息，因而需要利用ArcMap进行数据预处理。具体预处理方法为：将所有公司注册点信息导入至ArcMap，之后利用重庆市的内部区划图和点的经纬度信息进行叠置划分，再利用ArcMap的连接功能为原始企业数据添加区级信息，最后再从ArcMap中将处理后数据导入至MySQL数据库中。

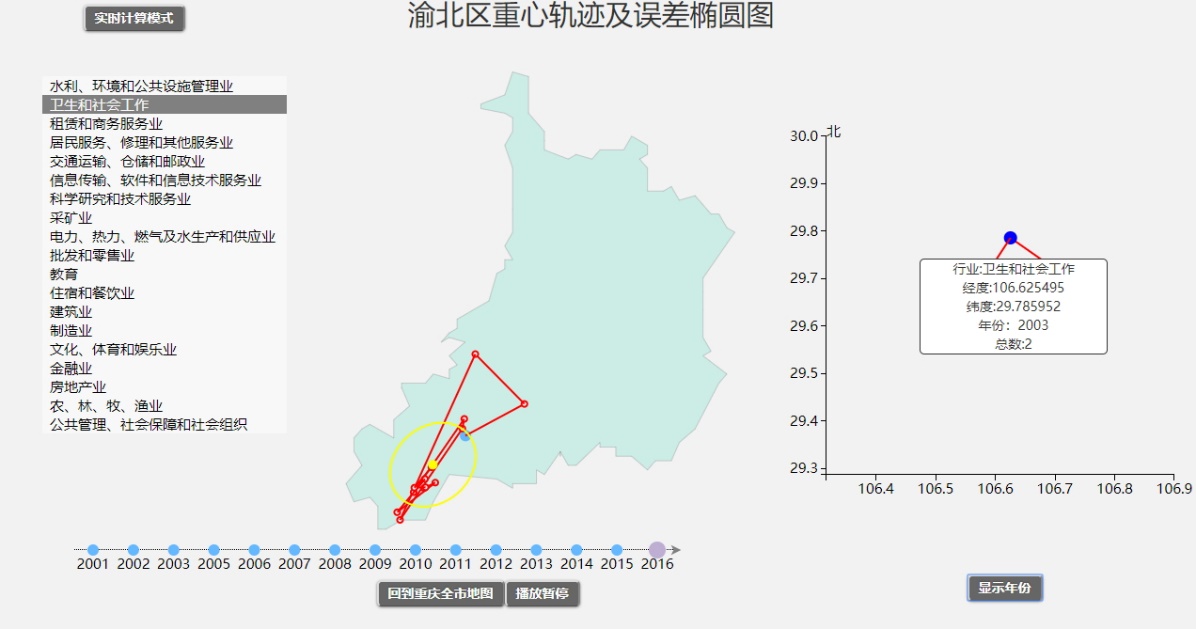
## 可视化方案实现效果

根据第3章所设计的可视化方案，本文利用相关可视化技术，建立基于Web的重庆市行业时空演化可视化系统。本次可视化系统的实现主要基于jQuery框架与D3.js库，最终所得系统包含有丰富的交互功能与动画效果，可清晰展现重心轨迹及标准差椭圆信息。

### 可视化系统界面

基于本文第3章所设计方案，利用jQuery框架与D3.js库等技术建立Web可视化系统，所得可视化界面如图5.1与图5.2所示。本界面中标题会随图中区域而对应变化，左侧行业选择组件可显示被选择的具体行业，时间轴上所选年份会放大变色显示。整体界面内的高亮重心、标准差椭圆和分区密度图与下方时间轴上变色放大显示的年份相对应。如图5.1所示，该图所展示的可视化结果为重庆市水利、环境和公共设施管理业的时空演化分析图，其中分区密度图和标准差椭圆均对应2016年的分析结果。左下方图例说明了数据分段标准和各颜色对应的数量段，用户可通过鼠标悬置在不同色块段上以获取该信息。界面右部坐标系内重心与左部重心相对应，当用户将鼠标悬置于右侧坐标系中的各年重心时，会显示相应提示框，提供更多行业内企业信息（如图5.2中所示）。

图**5.1** 市级行业时空演化分析图示例

图**5.2** 区级行业演化分析图示例

### 可视化系统功能

本可视化系统提供一个具有交互性、清晰的行业时空演化可视化结果，可以帮助用户进行后续分析。由于本系统所实现的交互功能及动画较多，此处以表格形式加以说明，详细功能列表见表5.1。

表**5.1** 可视化系统功能列表

|  |  |  |
| --- | --- | --- |
| 功能 | 调用方法 | 相关组件 |
| 行业选择 | 点击行业名或对应背景块 | 交互组件 |
| 跳转至区一级界面 | 点击地图上任意区块 | 地图 |
| 跳转至市一级界面 | 点击“回到重庆全市地图”按钮 | 交互组件 |
| 按照年份顺序显示重心、标准差椭圆及分区密度图 | (自动动画) 在选取了某一行业或跳转了地图层级后 | 地图、坐标系 |
| 显示某一年重心及标准差椭圆 | 在时间轴上点击对应该年份的圆形图标 | 地图、时间轴 |
| 显示、隐藏年份文字 | 点击“隐藏/显示年份”按钮 | 坐标系 |
| 显示信息框 | 在感兴趣的区域、图例块、重心上悬置鼠标 | 地图、坐标系 |
| 放大坐标系/回复地图部分为主 | 点击坐标系空白区域/点击缩小后的地图 | 地图、坐标系 |
| 将重心结果改为实时计算获取 | 点击“实时计算”按钮 | 地图、坐标系 |

## 并行化效率对比实验

为确定并行算法对于重心及标准差椭圆计算的加速效果及不同配置对计算效、率的影响，本文设计了不同配置、不同数据量数据的计算实验，并将计算时间进行了比较。本次实验中用到的集群最大规模可至9个Slave节点，1个Master节点。各节点最高配置均为4个2GHz的CPU核以及16GB内存。

### 不同数据量下的计算性能试验

本实验目的是对比基于Spark集群与基于单机的重心及标准差椭圆计算效率，通过比较在不同数据量条件下的计算时间来完成这一实验。重庆市总数据量为2119419条，以此为本次实验的最大数据量，其它实验数据量分别为1万、5万、10万、20万、40万、80万、100万、150万、200万条企业数据。具体各数据量情况下计算耗时如图6.1所示。其中9 slaves对应曲线为集群（共使用九个计算节点）模式下的计算耗时，local对应曲线为单机下的计算耗时。从图5.3中可以看出，在数据量较少时，单机和集群计算效率相近，甚至会出现单机计算耗时更短的情况。这一点主要是由于Spark集群在数据分发上需要耗时导致的，因此在数据量小时，集群计算速度提升不明显。

图**5.3** 不同数据规模下集群与单机计算效率对比

但在数据量逐渐增大直至40万条以上后，单机计算耗时迅速升高，而集群计算耗时增长较平稳。这一特点导致在较大数据量的计算情况下，Spark集群计算耗时明显小于单机计算，并且数据量越大时耗时差距会越明显。

### 不同集群配置下的计算性能试验

在数据量相同的情况下，不同的集群配置方案也会对计算效率产生影响。本文首先比较了不同节点数下数据计算耗时，以测试集群的可扩展性。本次实验共进行了两组比较，一组使用100万条数据进行计算，另一组使用全部2119419条数据进行计算。通过重复实验取平均数，最终耗时记录如图5.4所示。由此组实验可看出在处理全部数据时，节点数增加，计算时间有较为明显地减少；但在数据量较小，只有百万条时，时间减少不明显，但相较于单机计算均有所提高。从这一现象可以看出，对于较大数据量的数据集，横向扩展集群可以起到加速计算的效果。但在数据量较小，集群内计算节点性能未达到瓶颈时，该加速效果不明显。另一方面，这一结果也说明对于100万条数据的计算而言，由9个计算节点组成的小型集群就足以处理，并且计算速度不超过一分钟，可以满足网页加载需求。

图**5.4** 不同节点数集群计算耗时图

除此之外，本文还希望探究不同节点配置对于计算效率的影响，因此比较了当节点计算时使用的CPU核数与内存大小不同时的计算耗时。对比结果如图5.5所示，图中core指CPU核数，memory指节点使用内存大小。可以看出单纯的核数减半和内存量减半不会对计算耗时产生特别明显的影响，这说明现在集群的配置相较于数据计算需求还较高，还有继续扩大计算量的空间。但当继续降低CPU核数和内存大小后，计算耗时开始升高，证明这两项配置会影响计算速率与效果。

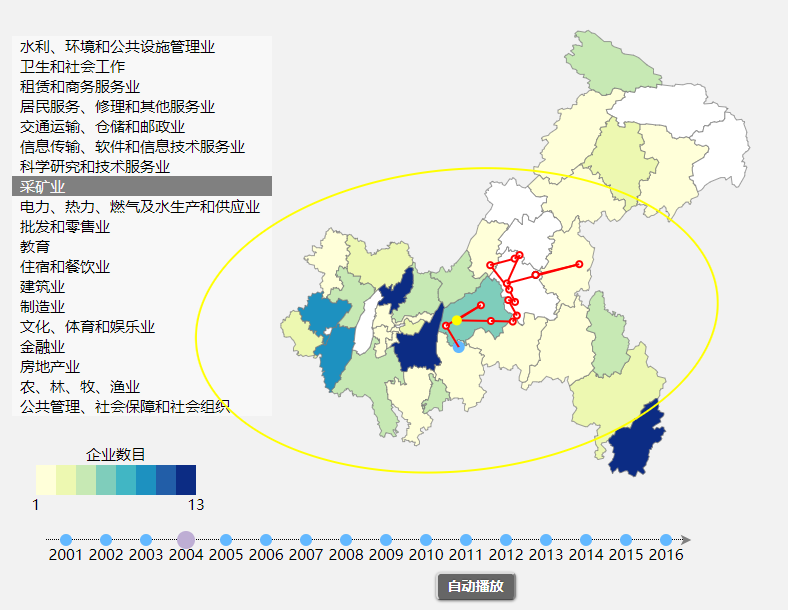
图**5.5** 不同配置下集群计算耗时对比图

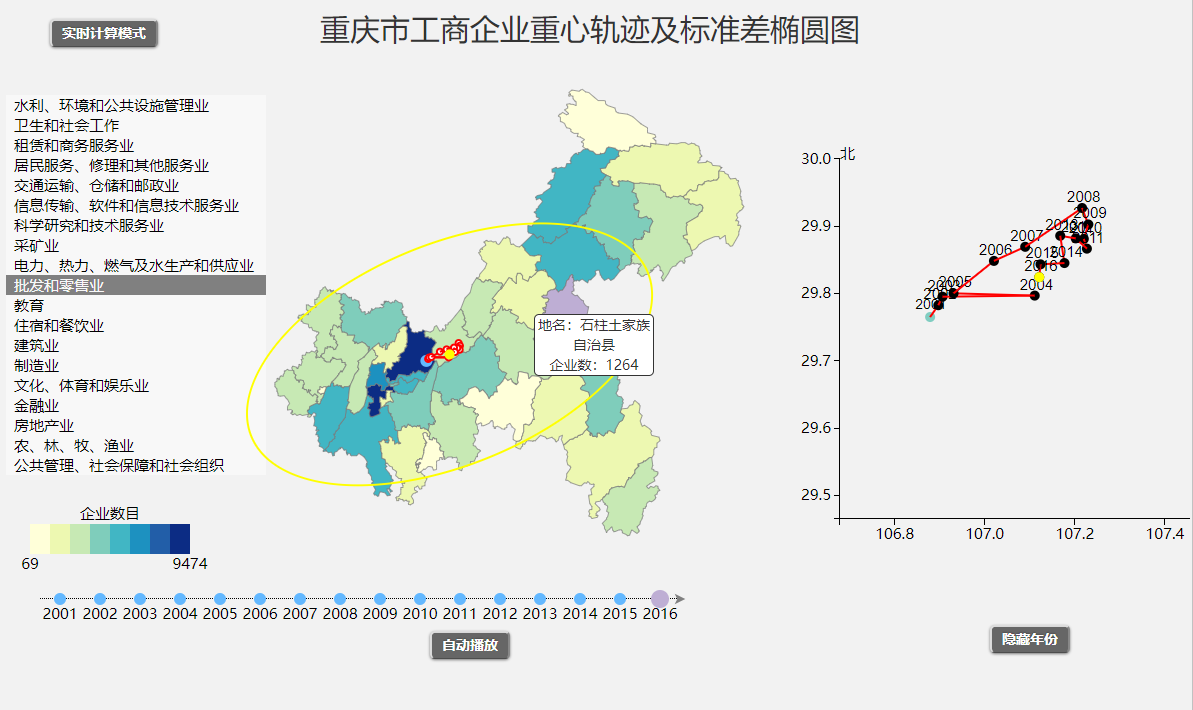
综合上述实验可以看出，节点数增加可以加快对于大数据集的处理效率，同时节点配置也会对计算结果产生影响。在合理地扩展集群和加强节点配置后，理论上可以将大数据集的计算时间限制在一分钟左右，基本足以满足准实时计算要求。而针对本文提出的可视化系统而言，全部数据在9个工作节点的集群上进行计算所需时间约为78秒，可接近准实时计算需求。

## 重庆市行业时空演化分析

结合计算结果和可视化结果，可获取重庆市及其各分区内19个行业大类的时空演化规律。由于区域及行业组合过多，本文在结合区域特点或其他信息进行进一步分析后，选取重庆市及其中两个典型区域的部分行业进行了演化规律成因的讨论。

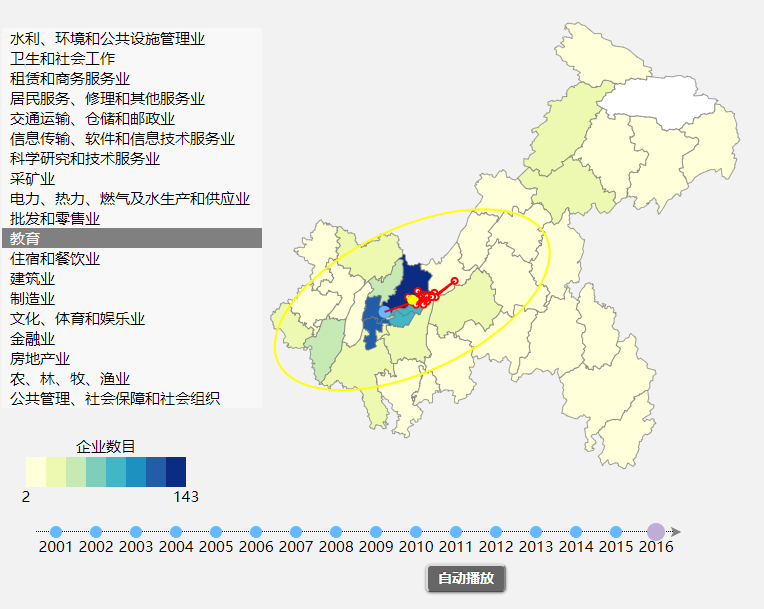
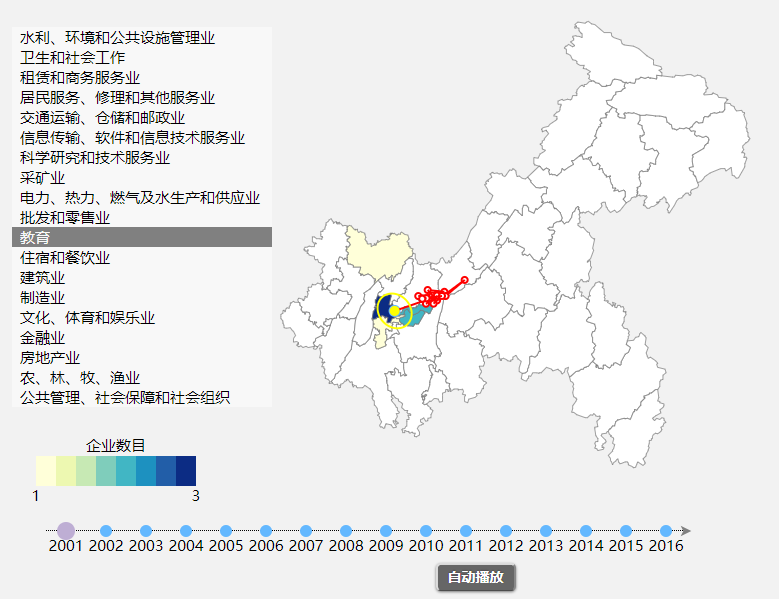
### 重庆市整体时空演化规律

通过对重庆市各行业时空演化图的观察，可发现大多数行业的标准差椭圆方向相似，长轴方向均是与横轴正方向（即正东方）夹角30度左右，这一方向与该地区地形方向相符，即说明该行业企业在全市范围内分布较均匀。但也存在有分布较为特殊的行业，如2001-2004年的采矿业的标准差椭圆（如图5.6所示）长轴方向均于正东方向相近。这一现象源于采矿业需要依赖特定的自然资源，其每年新注册企业较少，且会集中在部分区域，这一因素也导致该行业重心变化较为明显。采矿业本身属于工业，对于其空间分布的发展，不需要一味地追求去区域间平衡，而更应该针对分布规律，发现资源和需求规律，加以利用。

图**5.6** 重庆市采矿业时空演化图

图**5.7** 重庆市批发和零售业时空演化图

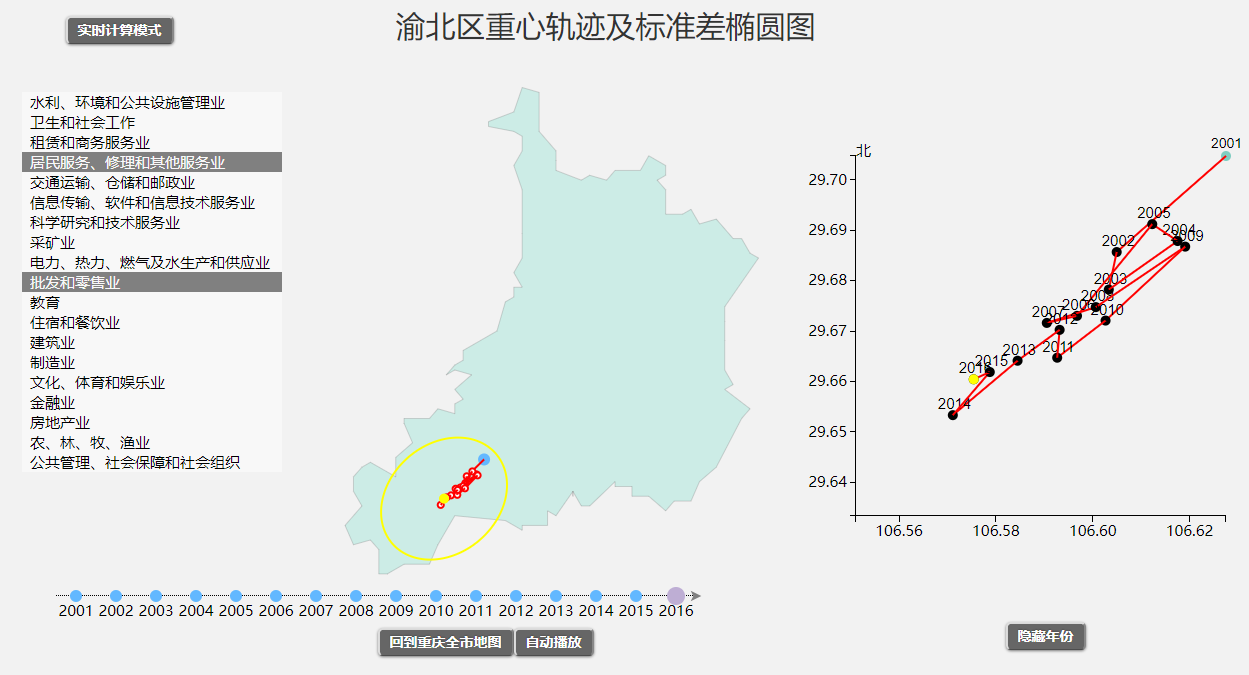
而对于与民众生活息息相关的服务业、商业等行业而言，其发展就应当更注重不同区域的平衡发展，即希望每年新注册企业的空间分布尽可能均匀。新注册企业越多且分布较均匀的行业，其重心变化越不明显，标准差椭圆方向越与地区主要方向相近，例如重庆市批发、零售业企业（如图5.7），其2001-2016年重心移动范围较小，且且标准差椭圆形状非常类似，只是大小发生了些许变化。这说明在该市批发、零售业新企业分布较为平均，且分布范围逐渐变广。对于这类分布均匀的行业而言，其发展并不存在明显问题，而只需继续维持均衡发展即可。



图**5.8** 重庆市2001年与2016年教育业标准差椭圆对比图

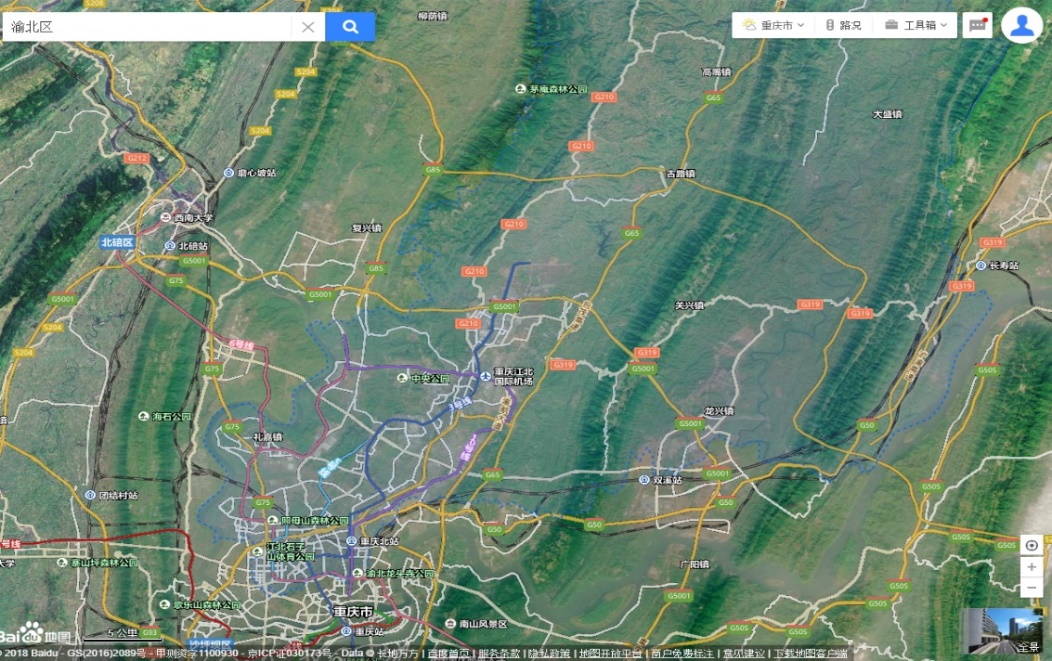
除去这些发展较均衡的行业之外，也有部分民生相关行业存在有局部发展不平衡而导致重心偏移以及标准差椭圆形状特殊的现象存在。以教育业为例，由不同年份的标准差椭圆对比图（图5.8）可看出，在2001年时重庆市教育类标准差椭圆极小，说明该行业企业较少，且分布非常集中，大部分其它地区都没有新的注册企业。而在之后重心连续向东北移动，展现东北部发展速度有所提升。但整体而言，即使是最为靠近东北部的2010年重心仍只是接近批发零售业所集中的相对城市中心。这说明了在整个16年间，重庆市教育行业的发展都较不均衡，西南部发展常年处于领先位置。但通过2016年与2001年标准差椭圆间的对比，可看出尽管新注册企业分布仍不均匀，但已较原来有了很大的提升，并正逐步朝向最接近均匀分布的椭圆（如批发和零售业相对应的椭圆）发展。对于教育业这类与民生息息相关的行业而言，区域间发展的不平衡极有可能会导致房价、人口流动的不平衡。因此，未来相关决策应当相对鼓励东北部教育企业，同时相关企业也可从中发现机遇，了解现有的教育行业发展趋势，而选择在东北部进行注册。

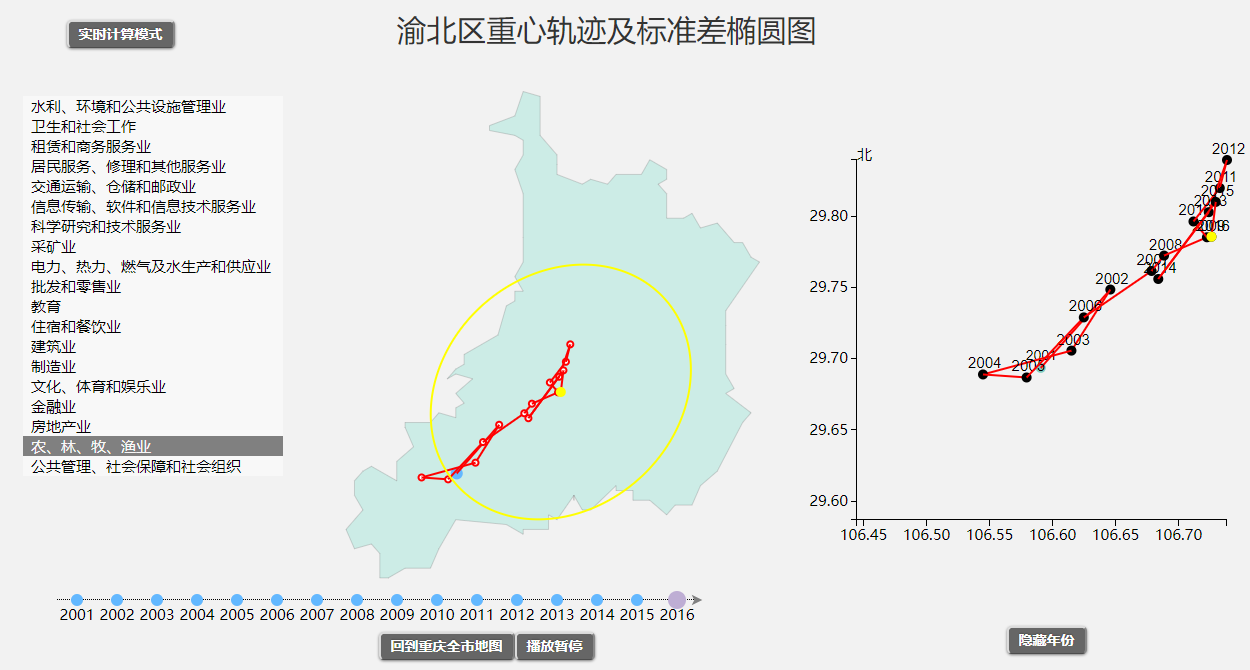
### 渝北区时空演化规律

渝北区作为重庆市主城九区之一，近年来GDP一直稳定为重庆市各区县第一。对于这一区域的研究有利于进一步发展重庆市的优势区域，并从中探究该区域发展规律，为其它区域发展提供参考。

图**5.9** 渝北区2016年批发和零售业时空演化图

如图5.9所示，2016年渝北区的批发零售行业的标准差椭圆位于区域西南部，且面积较小，这说明在整个区域中的该行业新企业更多的分布在西北部，且较为集中，分布并不是那么均匀。而另一方面，从重心迁移动画可以看出，该行业重心具有明显的向西南持续移动的趋势，这从另一方面反映出该区域零售批发业发展的不平衡。而结合渝北区地图及相关卫星地图（图5.10）研究，可发现该区域西南部靠近重庆市中心，属于较为繁华的区域，而北部以城镇、农田为主，属于农业较为发达区域。这也与图5.11中该区域农、林、牧、渔业重心及标准差椭圆图相符，整体重心较批发业更偏北部，且持续向北移动。

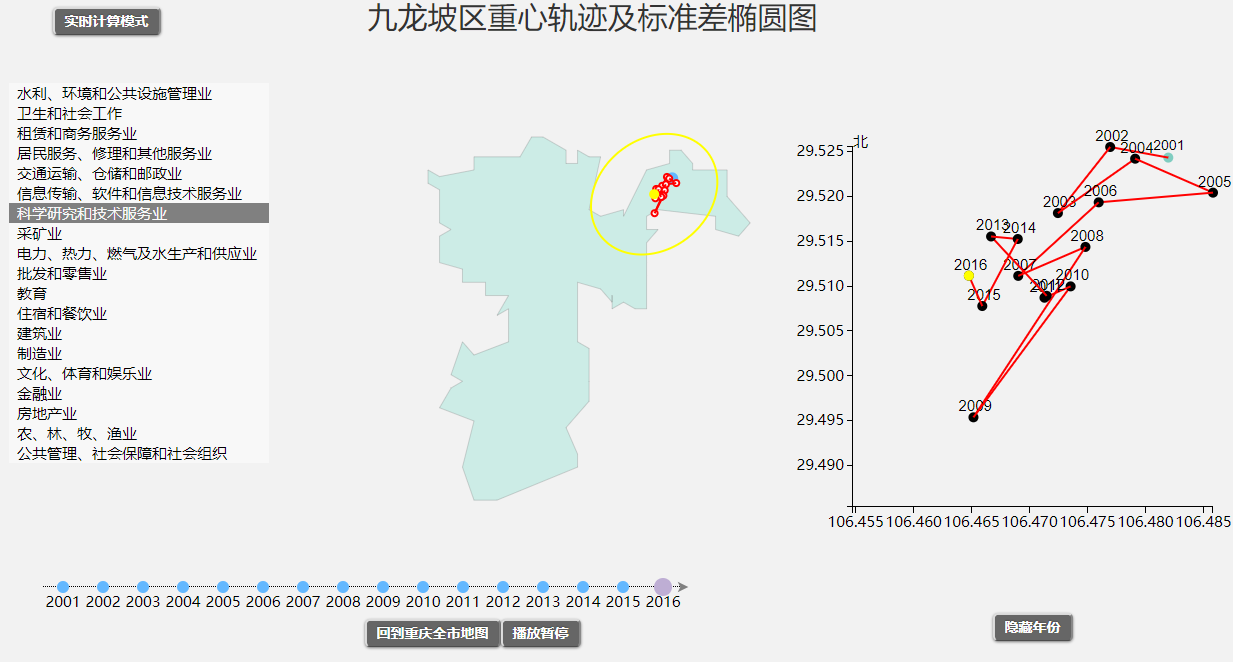
从两行业重心迁移轨迹可以看出，该区域企业分布集聚性十分明显且持续发展，即区域批发、零售业以及农林牧渔业发展不平衡程度持续增高。对于农业相关发生这一现象表明城乡分离程度日渐增高，属于正面变化。但对于零售业等民生基础行业而言，发展的极度不平衡则可能导致区域内市民幸福感相差过大，进而导致住房选择集中、交通拥挤等现象。对于这样的分布规律，有关部门需要有所注意，合理加以引导或控制。

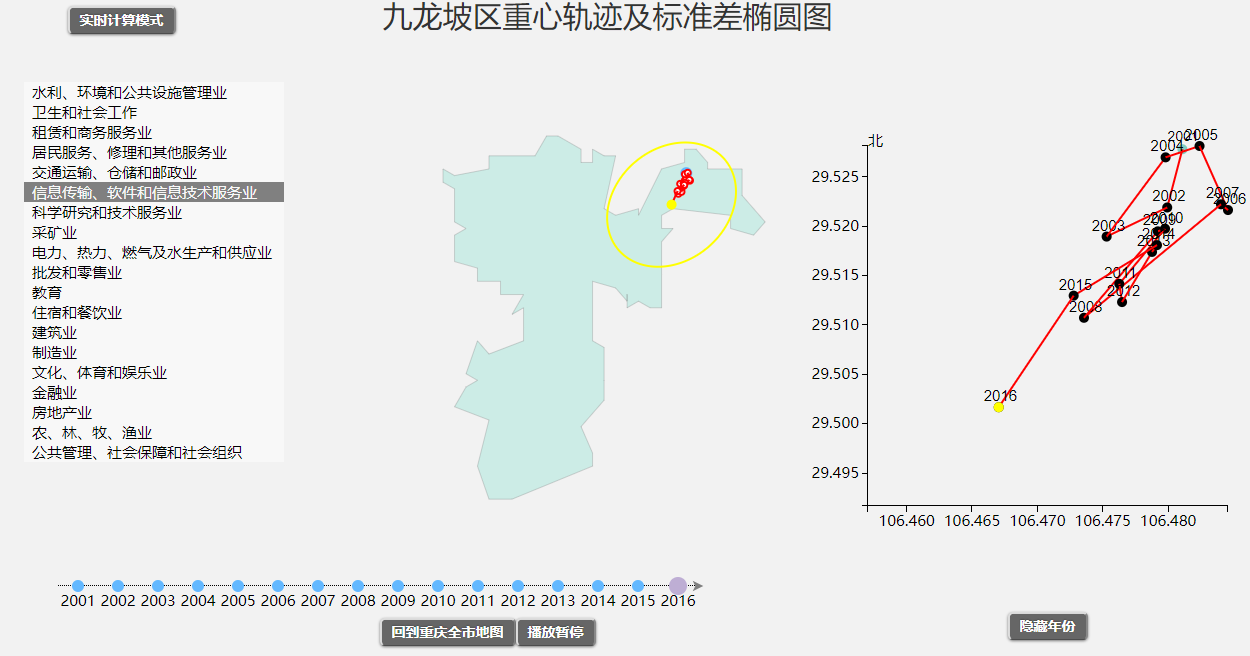
图**5.10** 渝北区卫星地图[54]

图**5.11** 渝北区2016年农、林、牧、渔业时空演化图

### 九龙坡区时空演化规律

九龙坡区同样属于重庆市中较为发达的区域之一，GDP在市内也是处于领先位置。而这一区域的主要特点是其高新技术产业发展较好，这一点也可以通过重庆市一级的分区密度图看出。而针对这一区域，本文进行了科学研究和技术服务业的相关信息图与信息传输、软件和信息技术服务业相关信息图的对比。通过图5.12与图5.13的对比，会发现两行业的重心迁移轨迹与各年标准差椭圆十分类似。这一特点在其他区域并没有这么明显，可能反映了在该区域相关行业企业分布高度的集中，且这两个行业间关系较为紧密，才导致了这样的分布特点。而针对这一特点，相关行业的企业管理者也可以加以利用，为自身选址提供参考。

由此结果，可以看出两行业时空演化规律类似，在此基础上，可以进一步开展聚类位置相关性分析或是邻接矩阵比较，以发现更深层次原因。此处基于重心轨迹和标准差椭圆的时空演化分析主要起到的就是初步分析，并提供下一步分析方向的作用。

图**5.12** 九龙坡区科学研究和技术服务业时空演化图

图**5.13** 九龙坡区信息传输、软件和信息技术服务业时空演化图

## 本章小结

基于Apache Spark的分布式计算可以有效提高时空演化分析相关计算的效率，在增多集群中计算节点以及提高单节点配置后可以达到对较大数据量数据的快速计算，可以支持用户在可视化平台上自定义选择后较快地获取结果。另一方面，在分析结果可视化的帮助下，可以更为清晰地展现区域时空演化的规律，有利于进行下一步分析。其中重心轨迹的迁移可以反映研究区域中是否存在有发展不平衡，局部发展较缓慢的现象，而标准差椭圆可以反映不同行业企业的密度与分布的方向。这些都可以帮助政府部门及相关企业进行决策，以及为之后深入分析提供思路和方向。

# 总结与展望

## 总结

行业发展规律一直是政府部门与相关企业运营者都关心的话题，而在当下这个大数据发展的潮流下，如何合理利用相关数据去探索行业发展规律也成为了重要的研究主题。另一方面，相较于单纯的从时间维度或空间维度探究行业发展规律，结合两方面进行分析，可以有效地探寻空间格局随时间的变化规律，进而能帮助对过去的发展进行反思以及对未来发展进行展望，符合可持续性发展的需求。

针对上述需求，本文以重庆市为例，选取重心迁移轨迹与标准差椭圆方法，进行了行业时空演化规律分析。另外由于企业数据量大，且可视化系统具有实时计算需求，本文针对重心轨迹与标准差椭圆计算设计了并行化算法，并基于Apache Spark分布式计算平台进行实现。具体而言，本文开展了如下工作：

1. 时空演化分析方法选择。讨论并总结五类点空间格局分析方法的适用范围及优缺点。选择其中适用于从企业信息中提取区域行业空间格局，并表现其随时间的演化特征的方法——重心轨迹及标准差椭圆进行本次分析。
2. 并行化算法实现与效率实验。 基于Apache Spark平台利用Scala语言，设计并实现重心轨迹及标准差椭圆的并行化计算算法，并结合语言特点对计算效率进行优化。本文设计相应算法效率对比实验，验证集群可以加快数据分析计算，同时证明该方案有支撑实时计算可能性。
3. 设计并实现基于Web的可视化方案。为清晰展示行业时空演化规律并支持进一步分析，本文设计了基于Web的可视化框架，并利用所选可视化框架及开源库实现。最终建立的可视化系统支持动画展现结果、交互选择年份、行业、层级等功能。同时在该系统中，还基于RESTful风格设计实现了由可视化层连接数据库或Apache Spark计算平台的端口，便于数据的直接读取和实时计算。
4. 对时空演化规律成因进行分析。利用建立的可视化系统，选取了其中具有代表性的行业和区域（渝北区与九龙坡区）进行了进一步的分析，探究其演化规律成因，并为下一步分析确定方向。

## 展望

本文虽然针对行业时空演化规律，完整地进行了数据分析算法设计、分析结果可视化与分析结果讨论，但仍有许多不足之处，需要进一步研究或完善，例如：

1. 本文的时空演化分析只利用了区域的企业注册信息和部分GDP信息，未能结合其他数据进行共同研究.对于本文中发现的规律，若能进一步结合其它数据加以验证或探究，将能得到更有价值的结论。
2. 并行化算法优化上，仅考虑了集群性能优化以及部分的代码优化，未能从根本上，如数据存储结构，分类方式上加以优化。
3. 在可视化系统上，尚有许多额外功能待实现，如允许用户自由选择区域进行相关计算等，这些可在实时计算的基础上加以实现。

参考文献

1. Lefever D W. Measuring Geographic Concentration by Means of the Standard Deviational Ellipse[J]. American Journal of Sociology, 1926, 32(1):88-94.
2. Zaharia M, Chowdhury M, Franklin M J, et al. Spark: cluster computing with working sets[C]// Usenix Conference on Hot Topics in Cloud Computing. USENIX Association, 2010:10-10.
3. 翟文华,周志太.人民日报新知新觉：推进大数据发展应更好发挥政府作用[EB/OL]. http://opinion.people.com.cn/n1/2018/0417/c1003-29930102.html
4. 人民网.盘点：十八大以来习近平关于重大经济问题论述摘编[EB/OL]. http://cpc.people.com.cn/xuexi/n1/2016/0112/c385475-28040684.html
5. De Smith M J, Goodchild M F, Longley P. Geospatial analysis: a comprehensive guide to principles, techniques and software tools[M]. Troubador Publishing Ltd, 2007.
6. Song Y, Miller H J. Simulating visit probability distributions within planar space-time prisms[J]. International Journal of Geographical Information Science, 2014, 28(1): 104-125.
7. Tang J, Song Y, Miller H J, et al. Estimating the most likely space–time paths, dwell times and path uncertainties from vehicle trajectory data: A time geographic method[J]. Transportation Research Part C: Emerging Technologies, 2016, 66: 176-194.
8. 郑长娟, 邹德玲, 王琳. 浙江服务业发展的时空演化和行业集聚特征[J]. 经济地理, 2015, 35(4):000114-122.
9. 焦新颖, 李伟, 陶卓霖,等. 北京城市扩展背景下产业时空演化研究[J]. 地理科学进展, 2014, 33(10):1332-1341.
10. 张珣, 钟耳顺, 张小虎,等. 2004-2008年北京城区商业网点空间分布与集聚特征[J]. 地理科学进展, 2013, 32(8):1207-1215.
11. Hewagamage K P, Hirakawa M, Ichikawa T. Interactive Visualization of Spatiotemporal Patterns Using Spirals on a Geographical Map[C]// IEEE Symposium on Visual Languages. IEEE Computer Society, 1999:296.
12. Guo D, Chen J, Maceachren A M, et al. A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP)[J]. IEEE Trans Vis Comput Graph, 2006, 12(6):1461-1474.
13. Ebert D S, Maciejewski R, Yun J, et al. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition[C]// Visual Analytics Science and Technology. IEEE, 2013:143-152.
14. Aloisio G, Cafaro M. A dynamic earth observation system[J]. Parallel Computing, 2003, 29(10):1357-1362.
15. Brodlie K. Visualization Over The World Wide Web[C]// Dagstuhl '97, Scien-tific Visualization. IEEE Computer Society, 1997:23-29.
16. Bender M, Klein R, Disch A, et al. A Functional Framework for Web-Based Information Visualization Systems[J]. IEEE Transactions on Visualization & Computer Graphics, 2000, 6(1):8-23.
17. 佘颖.十八大以来我国企业数量翻了一倍多[EB/OL]. http://www.ce.cn/xwzx/gnsz/gdxw/201710/25/t20171025\_26658536.shtml
18. Qin C Z, Zhan L. Parallelizing flow-accumulation calculations on graphics processing units—From iterative DEM preprocessing algorithm to recursive multiple-flow-direction algorithm[J]. Computers & Geosciences, 2012, 43(6):7-16.
19. Aloisio G, Cafaro M. A dynamic earth observation system[J]. Parallel Computing, 2003, 29(10):1357-1362.
20. 平利强. 基于云计算的海量时空数据存储及挖掘方法的研究和应用[D]. 杭州电子科技大学, 2014.
21. 范俊甫, 马廷, 周成虎,等. 分治法在GIS多边形快速合并算法中的应用及效率提升评价模型[J]. 地球信息科学学报, 2014, 16(2):158-164.
22. 唐文武, 施晓东, 朱大奎. GIS中使用改进的Dijkstra算法实现最短路径的计算[J]. 中国图象图形学报, 2000, 5(12):1019-1023.
23. Fellegara R, Iuricich F, De Floriani L. Efficient representation and analysis of triangulated terrains[C]// ACM Sigspatial International Conference on Advances in Geo-graphic Information Systems. ACM, 2017.
24. Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System[C]// IEEE, Symposium on MASS Storage Systems and Technologies. IEEE Computer Society, 2010:1-10.
25. Storm, distributed and fault-tolerant realtime computation[EB/OL]. https://www.storm-project.net/.
26. 颜俊, 王章华. 30年来中国人口、经济、产业重心演变的轨迹及其对比[J]. 人口与社会, 2010, 26(1):34-38.
27. Wan X, Wang W, Liu J, et al. Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range[J]. Bmc Medical Research Methodology, 2014, 14(1):135.
28. Kulin H W, Kuenne R E. An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics[J]. Journal of Regional Science, 1962, 4(2): 21-33.
29. Maron M E, Kuhns J L. On Relevance, Probabilistic Indexing and Information Retrieval[J]. Journal of the Acm, 1960, 7(3):216-244.
30. Cantoni E, Hastie T. Goodness‐of‐fit test for complete spatial randomness against mixtures of regular and clustered spatial point processes[J]. Biometrika, 2002, 89(2):411-421.
31. Haase P. Spatial pattern analysis in ecology based on Ripley's K‐function: Introduction and methods of edge correction[J]. Journal of Vegetation Science, 1995, 6(4):575-582.
32. Hartigan J A, Wong M A. Algorithm AS 136: A K-Means Clustering Algorithm[J]. Journal of the Royal Statistical Society, 1979, 28(1):100-108.
33. Hinneburg A, Keim D A. An efficient approach to clustering in large multimedia databases with noise[C]// International Conference on Knowledge Discovery and Data Mining. AA AI Press, 1998:58-65.
34. Sheikholeslami G, Chatterjee S, Zhang A. WaveCluster: a wavelet-based clustering approach for spatial data in very large databases[J]. Vldb Journal, 2000, 8(3-4):289-304.
35. Ord J K, Getis A. Local Spatial Autocorrelation Statistics: Distributional Issues and an Application[J]. Geographical Analysis, 1995, 27(4):286-306.
36. Chávez E, Navarro G, Baeza-Yates R A, et al. Searching in metric spaces. ACM Comput Surv[J]. Acm Computing Surveys, 2001, 33(3):273-321.
37. Smouse P E, Long J C, Sokal R R. Multiple Regression and Correlation Extensions of the Mantel Test of Matrix Correspondence[J]. Systematic Zoology, 1986, 35(4):627-632.
38. Hubert L J, Golledge R G. Comparing rectangular data matrices[J]. Environment and Planning A, 1982, 14(8): 1087-1095.
39. Yuill R S. The Standard Deviational Ellipse; An Updated Tool for Spatial Description[J]. Geografiska Annaler, 1971, 53(1):28-39.
40. React A JavaScript library for building user interfaces[EB/OL]. https://reactjs.org/
41. De Volder K. JQuery: A generic code browser with a declarative con-figuration language[M]//Practical Aspects of Declarative Languages. Springer Berlin Heidelberg, 2006: 88-102.
42. ECharts[EB/OL]. http://echarts.baidu.com/
43. Leaflet - a JavaScript library for interactive maps[EB/OL]. https://leafletjs.com/
44. Processing.org[EB/OL]. https://processing.org/
45. D3.js - Data-Driven Documents[EB/OL]. https://d3js.org/
46. Fielding R T. Architectural styles and the design of network-based software architectures[C]// University of California, Irvine, 2000:303.
47. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.
48. Papadimitriou S, Sun J. DisCo: Distributed Co-clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining[C]// Eighth IEEE International Conference on Data Mining. IEEE, 2009:512-521.
49. Yoo R M, Romano A, Kozyrakis C. Phoenix rebirth: Scalable MapReduce on a large-scale shared-memory system[C]// IEEE International Symposium on Workload Characterization. IEEE, 2009:198-207.
50. He B, Fang W, Luo Q, et al. Mars: A MapReduce Framework on graphics processors[C]// International Conference on Parallel Architectures and Compilation Techniques. IEEE, 2017:260-269.React A JavaScript library for building user interfaces. https://reactjs.org/
51. Armbrust M, Xin R S, Lian C, et al. Spark sql: Relational data processing in spark[C]//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM, 2015: 1383-1394.
52. Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing[C]// Usenix Conference on Networked Systems Design and Implementation. USENIX Association, 2012:2-2.
53. GB/T4754-2011，国民经济行业分类[S] . 北京：中国统计出版社，2011.
54. 百度地图[DB/OL]. https://map.baidu.com/.
55. Spark内核研究[EB/OL]. http://www.aichengxu.com/other/7898880.htm

# 致谢

转眼已是四年过去，我的本科生涯也到了画上句点的时候。回首这四年，有过收获也有过挫折，但不变的是老师同学一路以来对我的关心与帮助。在这篇累积了我本科四年所学的文章结尾，我衷心地向所有曾经给予我帮助的人表达我的感谢与尊敬。

首先我要感谢的是一直指导我、帮助我的桂志鹏老师。他不仅是本次毕业论文的指导老师，也是我这几年来科研路上的重要导师。从最初的大学生科研项目，到之后的GeoCommerce相关项目研究，以及最后的毕业设计，桂老师一直为我指点方向，并认真帮我提出各类改正意见，帮助我不断进步。这次的毕业设计时期正值桂老师工作繁忙之时，但是他仍坚持抽出时间定期与我探讨毕业设计发展方向，关心我的进展，同时在论文大体成稿后，还逐句逐段指出问题、进行修改。这些帮助都令我受益匪浅，也使得我在这个过程中更加深入地理解了网页设计与Spark相关内容，我相信这些都会成为我日后珍贵的财富。这两年来桂老师对我的科研和未来去路上都给予了非常大的关心与帮助，在与他以及同科研组学长、学姐的相处中，我收获了知识与快乐，同时也了解了科研的方法与精神。他向我展现了一个热爱学术、不断钻研进步的人应当如何提升自我并影响周围的人，也是我值得学习的榜样。

然后我要感谢的是大学四年来所有为我提供指导和帮助的老师。其中，我需要感谢吴华意老师和姚剑老师在我留学申请过程中给予无私帮助，没有他们的帮助，我也无法收获最后的录取。另外还要感谢我们的班主任石文轩老师，在我大一大二对未来迷茫之际，是他耐心地向我提供建议，让我确定了未来的方向。除此之外，我对于每一位曾教导过我的老师都怀抱有由衷的敬意与感谢，是他们的培养让我在这四年间对遥感、GIS产生了浓厚的兴趣，并收获了许多的知识。

感谢各位在桂志鹏老师科研组认识的学长、学姐们，尤其是栗法学长和王源学长在我毕业设计期间对我的悉心帮助与指导，他们两位在Spark和可视化方面的丰富知识是我最终完成本次毕业设计所必不可少的支撑。

感谢我身边认识的各位2014级的同学们，他们在我这大学四年的生活里带来了许多的精彩回忆，同时也在我毕业设计陷入瓶颈时，耐心倾听、鼓励我，才让我能最终克服一个个难题。

最后我一定要感谢的是遥感女篮队的所有队员，经理还有历任教练，无论这几年成绩如何，这四年来和她们一起的训练与相处给我留下了本科期间最珍贵的回忆。在我因为生活、学业陷入迷茫时，是女篮的大家给予我的关怀与支持让我重新振作起来，而在我无数次想要放弃篮球队的时候，也是因为有了她们每一个成员，我才能坚持到最后。走过这四年，一路以来的欢笑、泪水、疲惫与感动我都将牢记于心。

再次感谢所有帮助过我、关心过我的老师、同学，或许我没能一一提及你们的名字，但是我的心中永远有你们的位置。

世上没有不终止的旋律，离别总会到来。但是离别是新的开始，我会带着这四年的收获与回忆开始下一段旅程，愿所有人前程似锦。

# 附录

**附表1 企业注册原始数据样例**

|  |  |
| --- | --- |
| F\_ENTITY\_GUID | 357A773272824CB1A63CE1E10A7744D8 |
| F\_ENTITY\_NAME | \*(张三) |
| F\_SUPERVISE\_LEVEL | A |
| F\_CREDIT\_CODE | 5.00234E+15 |
| F\_ENTITY\_TYPE | 个体 |
| F\_INDUSTRY | 批发和零售业 |
| F\_LEGAL\_PERSON | 张三 |
| F\_ADDRESS | 重庆市开县高桥镇八仙居委会 |
| F\_CONTACT\_INFO | 5280\*\*\*6 |
| F\_GOVERN\_ORG | 重庆市开县工商行政管理局 |
| F\_STATION | 正坝工商所 |
| F\_GRID | 高桥镇 |
| F\_LONGITUDE | 106.558 |
| F\_LATITUDE | 29.574 |
| F\_TAGS | NULL |
| F\_FOUND\_DATE | 2009.4.21 0:00 |
| F\_IF\_LEVEL | NULL |
| F\_IF\_NEW\_ADD | NULL |
| F\_BIZLIC\_NUM | 5.00235E+14 |
| F\_BIZ\_SCOPE | 预包装食品、散装食品、乳制品（不含婴幼儿配方乳粉）、卷烟、雪茄烟（在相关许可证核定事项内从事经营）、百货零售。 |
| F\_ADD\_DATE | 2015.12.15 4:29 |
| F\_GOVERN\_ORG\_CODE | 500234 |
| F\_STATION\_CODE | 500234029 |
| F\_GRID\_CODE | 5.00234E+11 |
| F\_UNISCID | NULL |
| F\_NOT\_DEAD | 1 |
| F\_IS\_CITY | 0 |
| F\_CHECK\_ITEM | ,K01 |
| F\_CHECK\_POINT | 1 |
| F\_BD\_LONGITUDE | 108.2444 |
| F\_BD\_LATITUDE | 31.3671 |
| F\_PRECISE | 0 |
| F\_CONFIDENCE | 25 |
| F\_WGS\_LNG | 108.239966 |
| F\_WGS\_LAT | 31.369264 |
| F\_DB\_ID | 0 |