1 引言

1.1 项目背景

本系统是在地理信息系统工程课程中进行的一次综合实战演练。地理信息系统工程主要是研究：GIS 分析、设计与项目管理的相关内容。在老师的指导下，我们 3 人组成一个项目开发团队，按照 GIS 项目开发的的模式流程实施开发过程。经项目组反复讨论研究，决定开发一个武汉市出租车 GPS 轨迹处理与可视化系统。系统主要实现 GPS 轨迹数据与武汉市道路数据进行可视化；实现轨迹数据与道路网的地图匹配算法；分上下班高分期、平峰期、深夜三个时段标注热点区域和热点路段；针对热点路段的十字路口，从 24 小时交通流量图和 24 小时行车速度折线图两个角度分析道路服务能力。

1.2 课程设计目的

本次地理信息系统课程设计是在完成《地理信息系统原理》、《GIS软件工程》等课程后开展的实践性课程设计。本次课程设计主要完成地图矢量化（空间数据库设计与建库）、系统功能设计、系统界面设计和部分功能的实现；通过本次实习对GIS平台用于二次开发的各个功能模块有了较深入了解、熟悉和掌握；通过这次课程设计也进一步熟悉了基于GIS平台提供的组件进行应用系统开发的一般方法、思路和流程。其主要的目的概括为：

1、 熟练掌握GIS公共平台及其开发环境；
2、 熟练掌握GIS二次开发组件开发GIS应用软件；
3、 掌握GIS数据库建库方法和地物编码方法；
4、 理解和掌握GIS工程原理及方法。

1.3 课程设计任务

已知武汉市 2014 年 5 月的三天的出租车轨迹数据、来自 OSM 网的武汉市道路数据，请将 GPS 轨迹数据与武汉市道路数据进行可视化，并完成以下功能：

（1）实现数据可视化；
（2）实现轨迹数据与道路网的地图匹配算法；
（3）分上下班高分期、平峰期、深夜三个时段标注热点区域和热点路段；
（4）针对某个热点路段的十字路口，从以下方面分析道路服务能力：
1）24 小时交通流量图；
2）24 小时行车速度折线图；

GIS系统开发完成后，提交书面报告（编码系统、数据库设计、软件设计流程和软件使用说明等）和GIS系统软件运行程序及原始程序代码等。

1.4 参考资料

数据库系统概论（第四版）王珊 萨师煊 编著
计算机图形学（第三版）， Donald Hearn,M.Pauline Baker 著， 北京：电子工业出版社，

2004

计算机图形学实习指导（第二版）， 郭际元、黄晓萍、曾文、龚君芳， 中国地质大学（武汉）信息工程学院，2005

GIS 分析、设计与项目管理 孙云峰 林珲 著

百度地图开放平台相关文档

1.5 定义、缩写词

DB：英文全称 data base，数据库，是依照某种数据模型组织起来并存放二级存储器中的数据集合。

API：API（Application Programming Interface,应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

pd：pandas，pandas 是基于 NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas 提供了大量能使我们快速便捷地处理数据的函数和方法。

np：NumPy，NumPy 系统是 Python 的一种开源的数值计算扩展。这种工具可用来存储和处理大型矩阵，比 Python 自身的嵌套列表（nested list structure)结构要高效的多。

pl：pymssql，一个简单的 Python 数据库接口，构建于 FreeTDS 之上，为 Microsoft SQL Server 提供 Python DB-API（PEP-249）接口。

2 总体设计

2.1 系统设计的原则

项目组尽可能开发出一个比较成型的 GIS 功能平台，应用所学技术使其优化，系统设计的首要原则为简单易用，效率尽可能优化，技术比较全面。

2.2 设计中应用的关键技术

热力图：热力图是以特殊高亮的形式显示访客热衷的页面区域和访客所在的地理区域的图示。在本题中，我将武汉市所在地理网格进一步分解细化为 100×100 的网格，统计出租车轨迹点落在对应网格中的数量，以网格中心经纬度坐标代表该网格坐标，绘制热力图。

逆地理编码：逆地理编码技术提供将坐标点（经纬度）转换为对应位置信息（如所在行政区划，周边地标点分布）功能。这项技术同时支持全球行政区划位置描述及周边地标 POI 数据召回。

均值漂移聚类：均值偏移聚类是一种概率密度估计法，算法利用像素特征点概率密度函数的梯度推导而得，通过迭代运算收敛于概率密度函数的局部最大值。对于武汉市出租车轨迹数据，算法会首先在二维平面空间中生成大量待拟合的质心，这些质心会向着轨迹点密集的方向移动，最终确定聚簇的数量和质心坐标。

2.3 总体结构

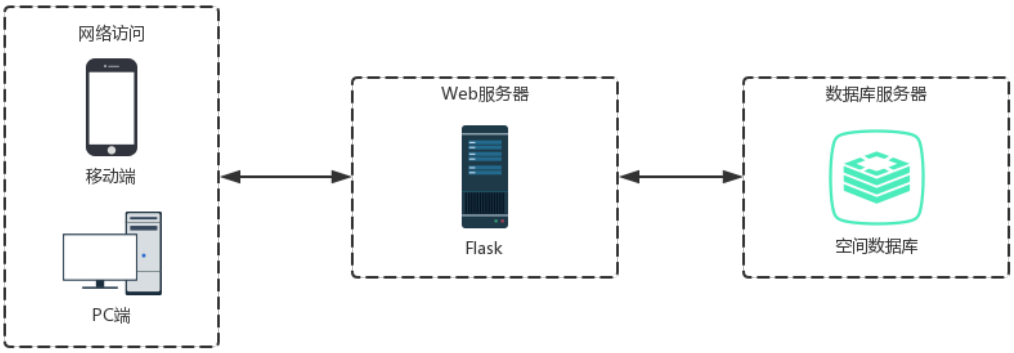

图 1 总体结构图

3 空间数据建库

对 20140507.csv 中的数据进行清洗，将经纬度明显在武汉之外的数据去掉，统一重整每条数据中的属性顺序后建库，数据库属性设计如表 1，数据库数据示例如图 2。

表 1 空间数据建库表

名称	类型	说明
mark	int	出租车编号
time	datetime2	时间
lng	float	经度
lat	float	纬度
d	int	方向
v	int	速度
acc	int	1 表示开空调，0 表示不开
heavy	int	1 表示重车，0 表示空车

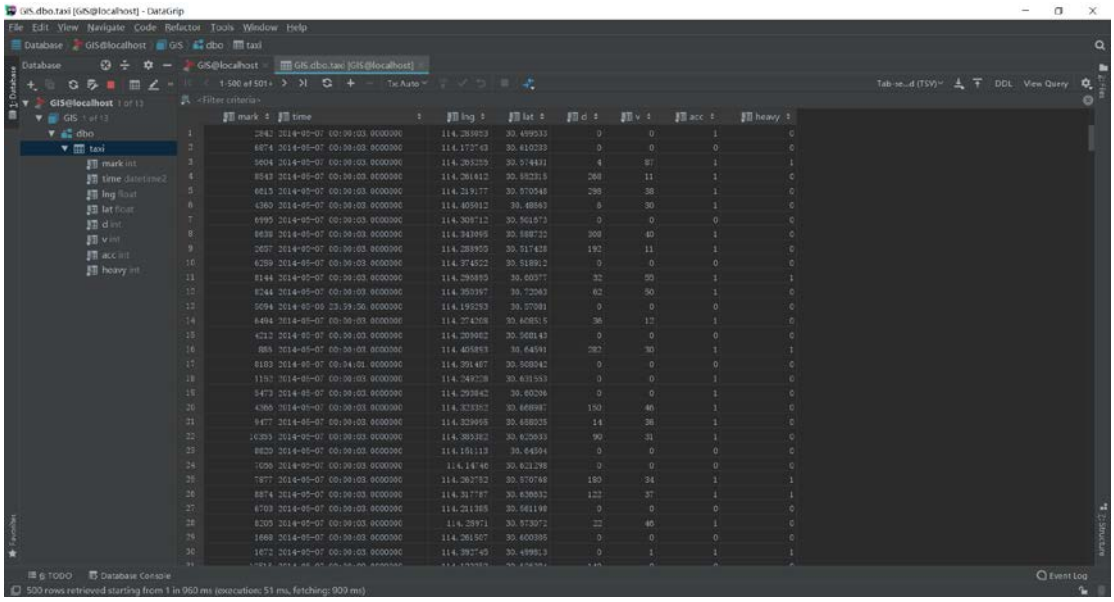

图 2 空间数据示例图

4 系统功能设计

系统采用 B/S 架构，系统构件图如图 3 所示。

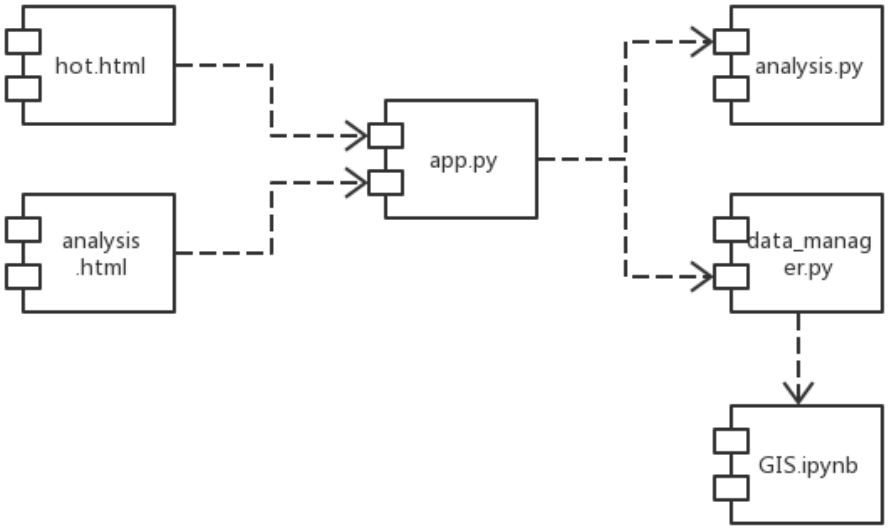

图 3 系统构件图

app.py：对外提供 Web 服务的程序入口，提供路由服务。

analysis.py：提供热力图分析和均值漂移聚类分析服务。

data_manager.py：提供数据库操作，用于获取指定时段的轨迹数据。

GIS.ipynb：对数据进行清洗重整。

hot.html：用于显示热力图的网页。

analysis.html：用于显示热点区域聚类结果的网页。

5 系统界面设计

界面设计基本原则：系统界面基于百度地图界面进行二次开发，即将分析数据显示在百度地图之上。

界面示意图如图 4、5 所示。

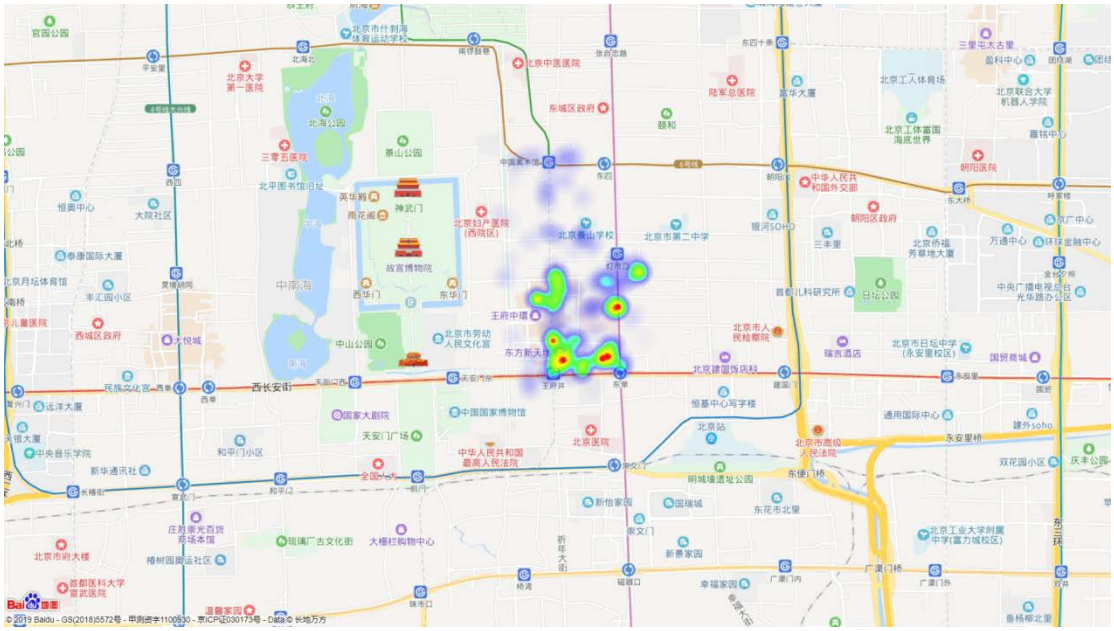

图 4 热力图示意图

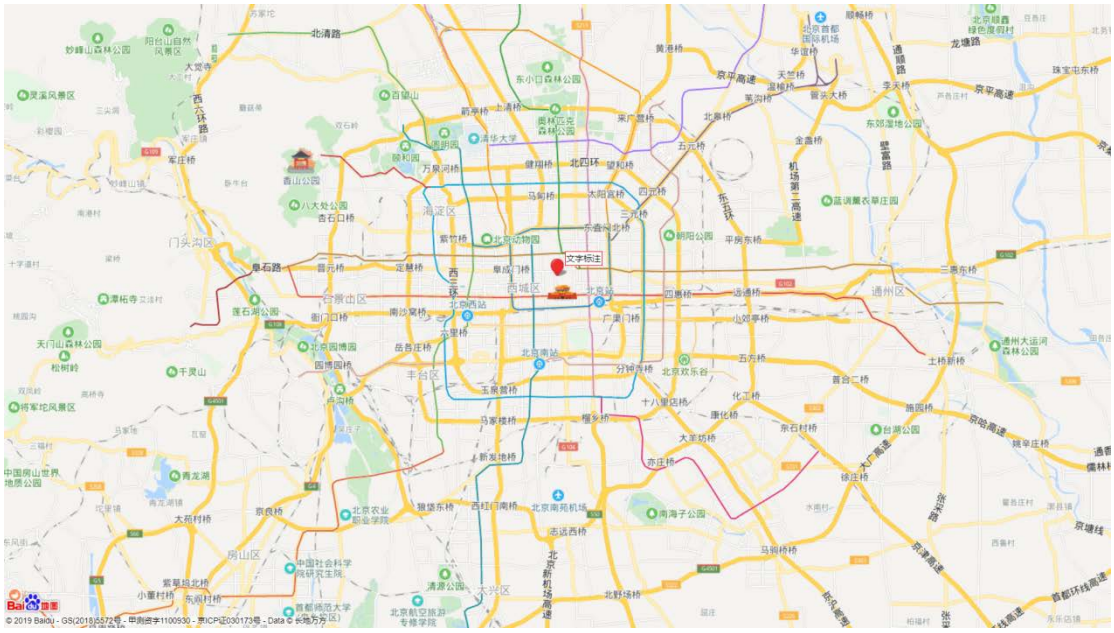

图 5 热点区域示例图

6 模块设计与功能实现

6.1 模块汇总表

表 2 模块汇总表

模块名称	功能简述
app.py	对外提供 Web 服务的程序入口，提供路由服务
analysis.py	提供热力图分析和均值漂移聚类分析服务
data_manager.py	提供数据库操作，用于获取指定时段的轨迹数据
GIS.ipynb	对数据进行清洗重整
hot.html	用于显示热力图的网页
analysis.html	用于显示热点区域聚类结果的网页

6.2 模块关系图

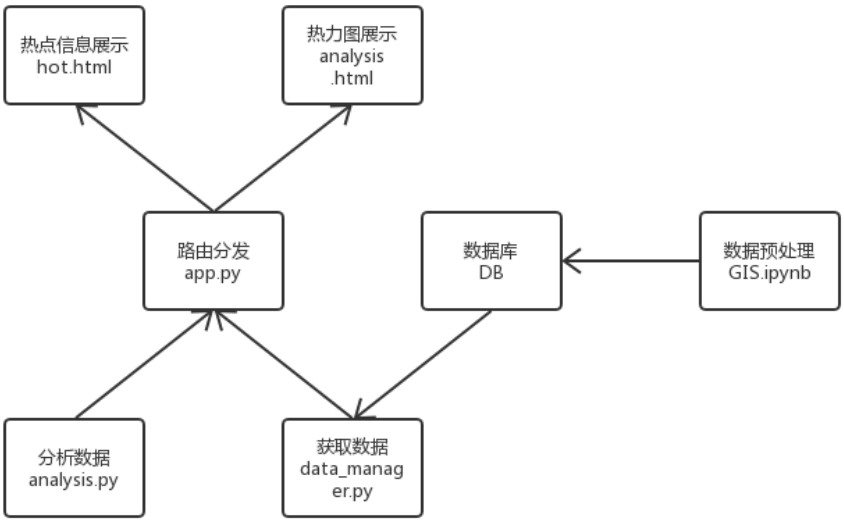

图 6 模块关系图

GIS.ipynb：预处理，按照GB18030编码标准逐行读入原始数据，将经纬度坐标不符合武汉市经纬跨度的点去掉，对每一行的字符串切割，统一重整属性的顺序和数量，写入新的CSV文件，使用数据库导入该文件。

```
import pandas as pd
wuhan_cars_csv = pd.read_csv('wuhan_GPS3Day/20140507.csv', encoding = 'gb18030', delimiter="\t", header=None)[0].tolist()
wuhan_cars = pd.DataFrame()
cac = [[], [], [], [], [], [], [], []]
locate = 0
for value in wuhan_cars_csv:
    value_list = value.split(',')
```

```
        if 113.41 <= float(value_list[2]) <= 115.05 and 29.58 <= float(value_list[3]) <= 31.22:
            cac[0].append(value_list[0])
            cac[1].append(value_list[1])
            cac[2].append(value_list[2])
            cac[3].append(value_list[3])
            cac[4].append(value_list[4])
            cac[5].append(value_list[5])
            if 'ACC开' in value_list:
                cac[6].append(1)
            else:
                cac[6].append(0)
            if '空车' in value_list:
                cac[7].append(0)
            else:
                cac[7].append(1)
            locate += 1
    wuhan_cars.insert(0, 'mark', cac[0])
    wuhan_cars.insert(1, 'time', cac[1])
    wuhan_cars.insert(2, 'lng', cac[2])
    wuhan_cars.insert(3, 'lat', cac[3])
    wuhan_cars.insert(4, 'd', cac[4])
    wuhan_cars.insert(5, 'v', cac[5])
    wuhan_cars.insert(6, 'acc', cac[6])
    wuhan_cars.insert(7, 'heavy', cac[7])
    wuhan_cars.to_csv('taxi.csv', index=0)
```

analysis.py： 热力图和热点区分析。

热力图，将武汉在经纬度上按跨度N（取N=100）等分，分割成N×N网格，对每个网格，统计落入该网格的轨迹点的数量，以该网格中心经纬度坐标代表该网格坐标，在坐标上，统计出的轨迹点数量就能表现该区域的热力。

```
    # 分析热力图
    def analysis_hot_power(points):
        points = np.array(points)
        max_lon = points[:, 0].max()
        min_lon = points[:, 0].min()
        max_lat = points[:, 1].max()
        min_lat = points[:, 1].min()
        width = max_lon - min_lon
        height = max_lat - min_lat
```

```
    calc_num = 100
    counts = np.zeros((calc_num, calc_num), dtype=np.int32)
    for point in points:
        width_offset = int(calc_num * (point[0] - min_lon) / width)
        height_offset = int(calc_num * (point[1] - min_lat) / height)
        if width_offset == calc_num:
            width_offset = calc_num-1
        if height_offset == calc_num:
            height_offset = calc_num-1
        counts[width_offset, height_offset] += 1
    hot_list = list()
    width_atom = width / calc_num
    height_atom = height / calc_num
    for i in range(calc_num):
        for j in range(calc_num):
            hot_list.append({
                'lng': float(min_lon + (i+0.5) * width_atom),
                'lat': float(min_lat + (j+0.5) * height_atom),
                'count': int(counts[i, j])
            })
    return hot_list
```

点聚合：使用均值漂移聚类算法对所有在时间段内的轨迹点聚类，为了在一定程度上防止因为质心随机初始化产生的异常，舍弃聚簇规模较小的质心，对于符合要求的质心，根据它们的经纬度坐标通过逆地理编码技术获取质心周围的热点区域和热点路段。

```
# 均值漂移聚类
def mean_shift(points):
    points = np.array(points)
    bandwidth = estimate_bandwidth(points, quantile=0.2, n_samples=500)
    ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
    ms.fit(points)
    centers = ms.cluster_centers_
    labels = ms.labels_
    laundry = dict()
    for label in labels:
        if label not in laundry:
            laundry[label] = 1
        else:
            laundry[label] += 1
```

```
    label_list = list()
    for label in laundry:
        label_list.append([label, laundry[label], centers[label]])
    mark_list = list()
    for label in label_list:
        if label[1] > 80:
            mark_list.append(True)
        else:
            mark_list.append(False)
    last_centers = list()
    for index in range(len(label_list)):
        if mark_list[index]:
            last_centers.append(label_list[index][2])
    nc = list()
    for center in last_centers:
        nc.append([float(center[0]), float(center[1])])
    return nc


# 逆地理编码
def get_address(points):
    mark = list()
    for point in points:
        params = {
            'callback': 'renderReverse',
            'location': str(point[1])+','+str(point[0]),
            'output': 'json',
            'pois': 1,
            'ak': 'pvhIiG4SN1PtV0ENPh2GMLvefAHvtHf3'
        }
        res = requests.get('http://api.map.baidu.com/geocoder/v2/', params=params)
        info = json.loads(res.text[res.text.find('(') + 1:len(res.text) - 1])['result']
        names = list()
        for name in info['pois']:
            names.append(name['name'])
        mark.append([info['addressComponent']['street'], names])
    return mark
```

data_manager.py： 获取数据，分高峰期（peak）、平峰期（normal）、深夜（valley）

三个时段从数据库中获取轨迹数据。

```
import pymssql as pl


# 获取轨迹数据
def get_points(time_interval):
    conn = pl.connect(server='localhost', user='sa', password='123', database='GIS')
    cur = conn.cursor()
    if time_interval == 'peak':
        cur.execute("select lng, lat from dbo.taxi where (time between cast('2014-05-07 07:00:00' as datetime) and cast('2014-05-07 09:00:00' as datetime)) or (time between cast('2014-05-07 17:00:00' as datetime) and cast('2014-05-07 19:00:00' as datetime));")
    elif time_interval == 'normal':
        cur.execute("select lng, lat from dbo.taxi where time between cast('2014-05-07 9:00:00' as datetime) and cast('2014-05-07 17:00:00' as datetime);")
    else:
        cur.execute("select lng, lat from dbo.taxi where (time between cast('2014-05-07 00:00:00' as datetime) and cast('2014-05-07 04:00:00' as datetime)) or (time between cast('2014-05-07 22:00:00' as datetime) and cast('2014-05-07 23:59:59' as datetime));")
    points = cur.fetchall()
    cur.close()
    conn.close()
    return points
```

app.py：程序入口，用于分发路由，Web服务器使用本地IP http://127.0.0.1:5000/,使用analysis/peak、analysis/normal、analysis/valley路径分别调用分析三个时段的热力图的服务网页，使用hot/peak、hot/normal、hot/valley路径分别调用分析三个时段的热点图的服务网页。

```
from flask import Flask, render_template
from analysis import *
from data_manager import get_points

app = Flask(__name__)


# 热力图路由
@app.route('/analysis/<time_interval>')
def analysis(time_interval):
    points = get_points(time_interval)
    hot_list = analysis_hot_power(points)
```

```
        return render_template('analysis.html', hot_list=json.dumps(hot_list),
time_interval=time_interval)


    # 热点区域路由
    @app.route('/hot/<time_interval>')
    def hot(time_interval):
        points = get_points(time_interval)
        centers = mean_shift(points)
        centers_info = json.dumps(get_address(centers), ensure_ascii=False)
        return render_template('hot.html', time_interval=time_interval,
                               centers=json.dumps(centers), centers_info=centers_info)


    # 程序入口
    if __name__ == '__main__':
        app.run()
```

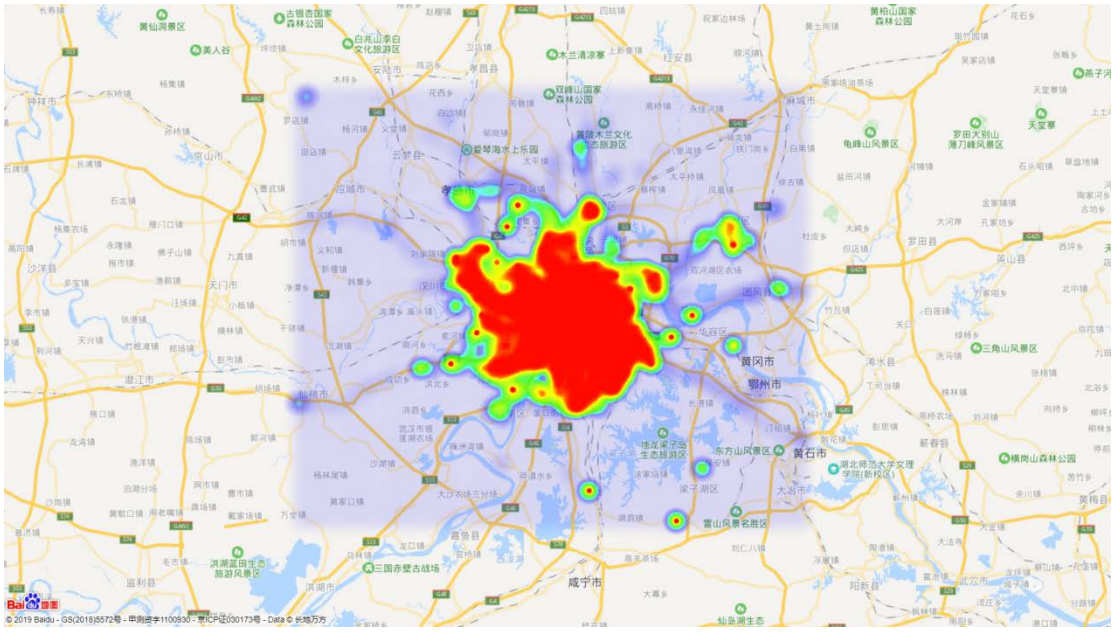

图 7热力图

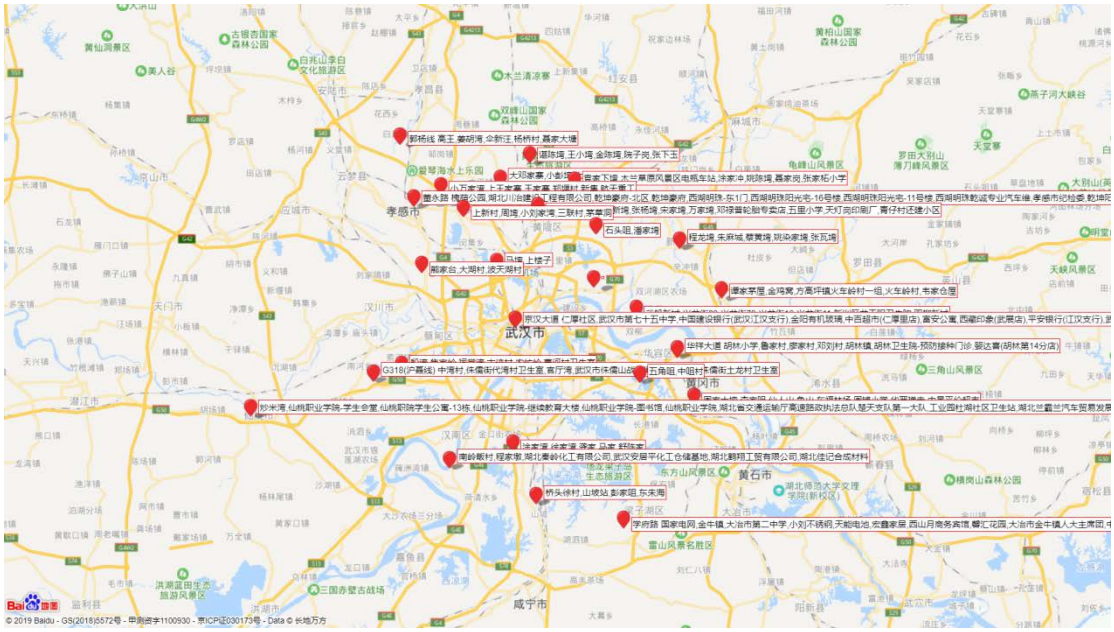

图 8 热点图

7 开发工具与系统运行环境

浏览器：Chrome
服务端操作系统：Windows 10
功能测试环境：Anaconda
集成开发环境：Pycharm
数据库服务器系统：SQL Server 2017
Web 开发框架：Flask
开发语言：Python、JavaScript
第三方库：json、requests、pandas、numpy、pymssql、sklearn
第三方平台：百度地图开放平台