开放型实验报告
——基于 POI 的城市生活理解

1.实验方案设计

1.1 背景

POI 是“Point of Interest”的缩写，中文可以翻译为“兴趣点”。在地理信息系统中，一个 POI 可以是一栋房子、一个商铺、一个邮筒、一个公交站等。特别地， POI 是现代城市的重要组成元素，是人们生活和活动的主要场所，不同类别、规模、位置的 POI 可以刻画不同的生活方式和区域功能等。

城市在其发展过程中为满足城市居民不同的生活需求，逐渐形成居住区、工业区、商业区和混合功能区等不同的功能单元。为把握城市空间结构以及制定科学合理的规划，规划人员和学者对城市进行功能区划分。POI 数据具有样本量大，涵盖信息细致等优势。通过对 POI 数据进行处理分析，定量划分城市单一功能区和混合功能区，研究结果可以更好地理解城市空间结构。

1.2 原理

网络爬虫（简称“爬虫”）是一种自动化提取网页的程序，用于获取收集互联网上的信息。狭义上的爬虫一般由搜索引擎操作以进行网络索引，该类爬虫会从“种子”URL（统一资源定位系统， Uniform Resource Locator） 列表开始，访问这些 URL 对应的 Web 服务器，并识别检索网页中的所有超链接，进而递归访问更大范围的网页。通过这种方式，搜索引擎可以尽可能地覆盖更大范围的互联网网站，为用户提供网络索引服务。随着互联网上数据数量与数据价值的增长，对互联网上的数据进行自动化获取与分析变得越来越有意义，因此，广义上的爬虫泛指一切自动化提取网页、收集互联网上信息的程序。

为了规范网络爬虫的行为， 使站点能够拒绝搜索引擎爬虫的访问，互联网界发展出了一种通行的道德标准——robots 协议。该协议体现为网站根目录下的 robots.txt 文件，搜索引擎爬虫可以阅读该文件以获取站点不希望其访问的页面范围。不过，robots 协议只是一种道德规范，不是强制性命令，无法阻止不遵守该协议的爬虫访问其内容。

对于以获取所需数据为目的的爬虫，例如本实验将实现的北京房价数据爬虫，一般有两种实现方式： 1）获取网页完整的 HTML 数据， 再进行网页解析和异步请求分析来获取所需要的信息； 2）通过官方 API（应用程序编程接口， Application Programming Interface）的方式获取数据。

本次实验以通过美团实现收集北京 POI 数据的网络爬虫，采用直接获取网页 HTML 文件并进行解析的方式实现。

1.3 方案设计

POI 数据爬取流程如下：1.熟悉网页结构，通过 requests 库尝试自动化下载网页；2.通过 Beautiful4 库解析 HTML 文件，提取相关数据。了解不同类型兴趣点 URL 的组织方式，获取相应页面的 HTML 文件。通过 CSS Selector 定位元素，从网页文件中获取所需要的信息（名称，地址，评分），并进行数据处理和储存。需要注意网络爬虫的访问频率，以及设置 cookie。

需要将爬取到的地点名称转换为经纬度坐标，通过高德开发平台的地理编码功能实现。以便根据种类或评分划分 POI 在城市中的空间分布（热力图）。由于高德 API 获取的经纬度数据基于 GCJ-02 火星坐标系，需要将其转换为 WGS-84 坐标以便于进行进一步的分析和可视化。

采用 Express 作为互联网服务 API 实现所使用的框架，实现种类、评分热力图数据的 API，及通过经纬度与 POI 类型查询名称的 API。采用 React 框架作为动态网页实现的框架。基于该框架实现通过 POI 名称查询评分，查看 POI 热力图等功能的前端 UI。

通过 kmeans 聚类方法进行城市功能区的划分。先通过 POI 的功能将不同的 POI 点进行合并，最后通过计算不同种类 POI 的密度将得到的各大功能区域的实际功能进行标注。[1]

1.4 分析及验证方案

登录 local:3000 查看城市 POI 的热力图并进行部分 POI 评分情况的查询，得到正确的热力图及 POI 评分情况。

利用 POI 通过 ArcGIS 分析北京市的城市功能区划分情况，获得不同种类 POI 的缓冲区，计算各区域中不同种类 POI 缓冲区的面积，并据此对北京城市功能区进行划分。与 kmeans 聚类结果做比较。

1.5 时间安排

6.2 进行相关资料查询，并撰写设计文档。

6.3-6.9 基本完成美团一个页面的 html 文件爬取

6.10-6.16 完成递归爬取多个页面

6.17-6.23 完成 html 文件解析，并生成 json 文件

6.23 完成前端的基本功能开发

---

[1] 池娇,焦利民,董婷,谷岩岩,马雅兰.基于 POI 数据的城市功能区定量识别及其可视化[J].测绘地理信息,2016,41(02):68-73.DOI:10.14188/j.2095-6045.2016.02.017.

6.24 进行 kmeans 聚类及 ArcGIS 验证

6.25 完成实验报告撰写

2.实验过程记录

2.1 POI 数据爬取

通过 scrapy 框架进行美团网不同城市的信息爬取，从而获得用于爬取 POI 信息的 URL。根据此 URL 爬取北京市不同种类 POI 的 HTML 文件。


```
{
  poiId: 266050,
  frontImg:
    "http://p0.meituan.net/600.600/biztone/266050_1630238166959.jpeg",
  title: "鑫巴蜀川菜馆（车站路店）",
  avgScore: 4.1,
  allCommentNum: 2288,
  address: "新华大街",
  avgPrice: 82,
  dealList: [
    {
      title: "100元代金券1张，可叠加5张",
      price: 92,
      soldCounts: 0,
    },
    {
      title: "超值双人套餐，提供免费WiFi",
      price: 108,
      soldCounts: 0,
    },
    { title: "双人聚会A套餐，包间免费", price: 168, soldCounts: 0 },
    {
      title: "至尊烤鸭1份，提供免费WiFi",
      price: 98,
      soldCounts: 0,
    },
    {
```


图 1 爬取的 html 文件

2.2 html 文件解析

利用 selector.xpath 方法对 html 文件进行解析，首先在网页中找到感兴趣的数据，利用检查在开发者工具中找到其位置，并右键复制其 xpath 信息进行解析，得到相应的数据。同时，观察到美团 url 的组织方式为在相应分区后加“/pn{页数}/”，因此，除了关心的 POI 信息之外，还需要获取相关分类的总页数，从而爬取所有页面的 html 文件。

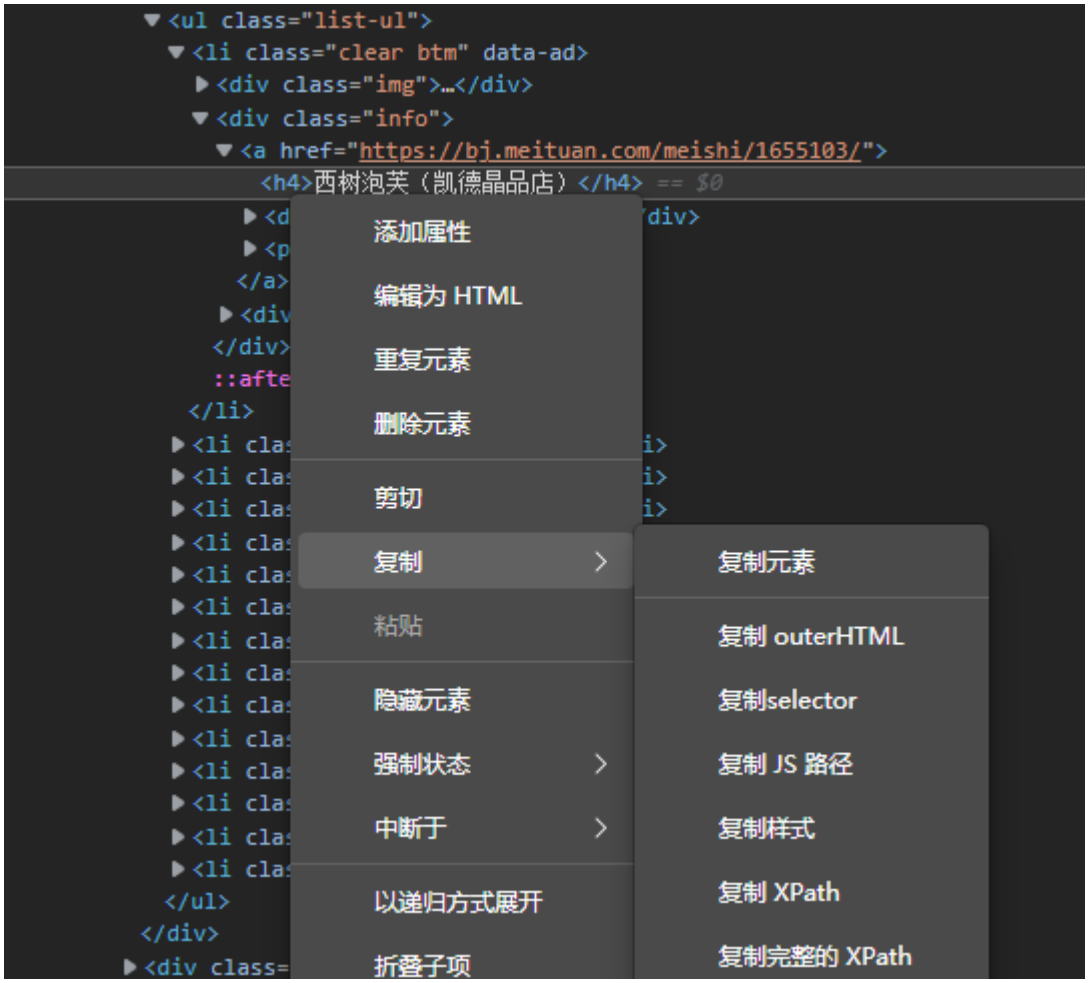


图 2 获取 XPath 方法

2.3 地理信息数据处理

最终共获取到北京市的 POI 数据一万余条，利用高德平台提供的地理编码服务得到每个 POI 对应的经纬度信息。但需要注意的是高德提供的经纬度信息是基于 GCJ-02 火星坐标系，需要根据以下公式将其转换为 WGS-84 坐标：

$$a = 6378245.0, \quad e = 0.00669342162296594323$$

$$lon = lon_{gcj} - 105.0, \quad lat = lat_{gcj} - 35.0$$

$$\Delta lon = 300 + lon + 2.0lat + 0.1lon^2 + 0.1lon * lat + 0.1\sqrt{|lon|}$$

$$+\frac{1}{3}(40\sin(6\pi * lon) + 40\sin(2\pi * lon))$$

$$+\frac{1}{3}(40\sin(\pi * lon) + 80\sin(\frac{1}{3}\pi * lon))$$

$$+\frac{1}{3}(300\sin(\frac{1}{12}\pi * lon) + 600\sin(\frac{1}{30}\pi * lon))$$

$$\Delta lat = -100 + 0.2lon + 3.0lat + 0.2lat^2 + 0.1lon * lat + 0.2\sqrt{|lon|}$$
$$+\frac{1}{3}(40\sin(6\pi * lon) + 40\sin(2\pi * lon))$$
$$+\frac{1}{3}(40\sin(\pi * lat) + 80\sin(\frac{1}{3}\pi * lat))$$
$$+\frac{1}{3}(320\sin(\frac{1}{12}\pi * lat) + 640\sin(\frac{1}{30}\pi * lat))$$
$$m = 1 - e * (\sin(\frac{180 * lat}{\pi}))^2$$
$$\text{lon}_{wgs} = lon - \frac{\Delta lon * 180}{a * \cos(\frac{180 * lat}{\pi})\pi} * m^{\frac{1}{2}} \qquad lat_{wgs} = lat - \frac{\Delta lat * 180}{a * (1-e) * \pi} * m^{\frac{3}{2}}$$

2.4 互联网服务 API 实现

将之前处理过的地理信息数据及 POI 评分等相关数据进行整合。采用 Express 作为互联网服务 API 实现所使用的框架，实现种类、评分热力图数据的 API，及通过经纬度与 POI 类型查询名称的 API。采用 React 框架作为动态网页实现的框架。基于该框架实现通过 POI 名称查询评分，查看 POI 热力图等功能的前端 UI。

2.5 分析城市功能区组合

利用 Scikit-learn 包对得到的 POI 数据进行 kmeans 聚类分析。设置聚类数设为 10，发现此时仅有线性分类面，尝试使用 SpectralClustering 分类方式，设置相关参数为 n_clusters=10, affinity='nearest_neighbors', assign_labels='kmeans'，得到聚类结果。

2.6 ArcGIS 验证

将得到的 POI 数据输入 ArcGIS 进行分析。首先获得不同种类 POI 的缓冲区，计算不同城市区域中不同种类 POI 缓冲区的面积，并与对应种类的 POI 数相乘。求得每个区域内，各种类 POI 面积与数量乘积占总和的比例，并根据这个比例对城市的不同区域进行功能区划分，即：

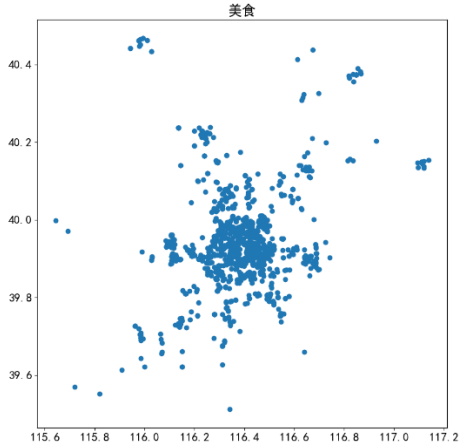
$$H(i|k) = \frac{\frac{S_i}{S_k} \times n_i}{\sum_{i=1}^{n=10}\left(\frac{S_i}{S_k} \times n_i\right)} \times 100\%$$

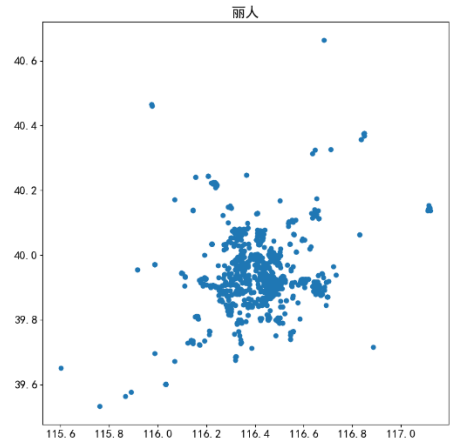
3.实验结果分析

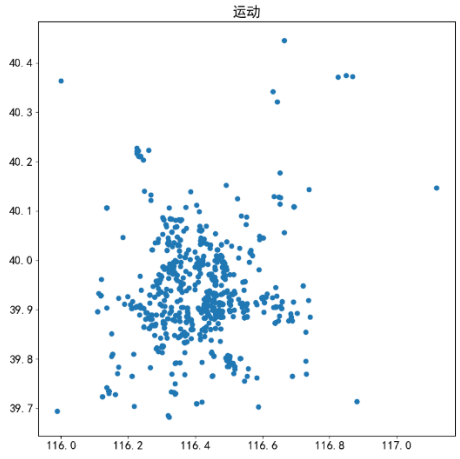
成功得到页面的 html 文件如图 1 所示，对其进行解析，得到的 json 文件如下图：

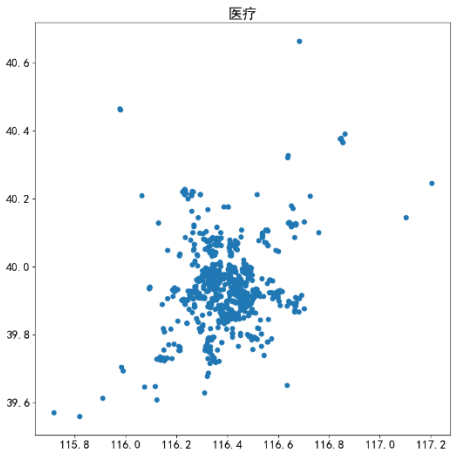
```
{"name": "新兴西饺手馆（望京店）", "avgScore": 4.4, "avgPrice": 51, "type": "0"},
{"name": "晋风庄园（上庄店）", "avgScore": 4.3, "avgPrice": 85, "type": "0"},
{"name": "starhot星火韩国炸鸡火锅料理", "avgScore": 4.3, "avgPrice": 38, "type": "0"},
{"name": "老诚一锅·麻辣香锅（鼓楼店）", "avgScore": 4, "avgPrice": 71, "type": "0"},
{"name": "四季民福烤鸭店（大栅栏店）", "avgScore": 4.8, "avgPrice": 103, "type": "0"},
{"name": "沿海405烤场（沿海赛洛城店）", "avgScore": 4.5, "avgPrice": 85, "type": "0"},
{"name": "鹤一烤肉自助餐厅（中关村店）", "avgScore": 4.6, "avgPrice": 157, "type": "0"},
{"name": "北京全聚德（天安门店）", "avgScore": 3.7, "avgPrice": 137, "type": "0"},
{"name": "辣莊重庆老火锅（东直门店）", "avgScore": 4.7, "avgPrice": 95, "type": "0"},
{"name": "猫眼披萨（阳光店）", "avgScore": 4.3, "avgPrice": 31, "type": "0"},
{"name": "春味手工春饼（长楹天街店）", "avgScore": 4.1, "avgPrice": 52, "type": "0"},
{"name": "虾吃虾涮（良乡大学城店）", "avgScore": 4.7, "avgPrice": 60, "type": "0"},
{"name": "鼓楼吃面 Punk Rock Noodles（创始店）", "avgScore": 4.5, "avgPrice": 79, "type": "0"},
{"name": "牛村来人潮汕鲜牛肉火锅（慈云寺店）", "avgScore": 4.5, "avgPrice": 128, "type": "0"},
{"name": "大成路九号·食府", "avgScore": 4, "avgPrice": 103, "type": "0"},
{"name": "云念·云水谣（延静里店）", "avgScore": 4.3, "avgPrice": 75, "type": "0"},
```

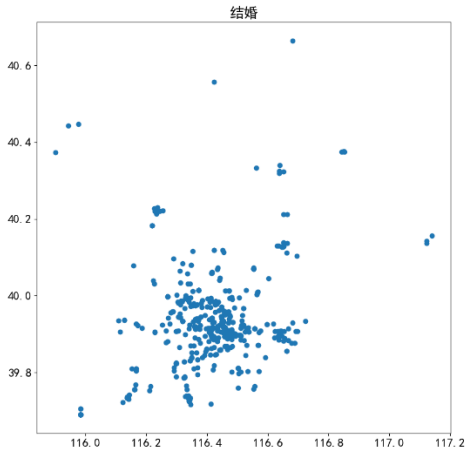

可以对得到的不同种类 POI 分布情况进行可视化，得到：

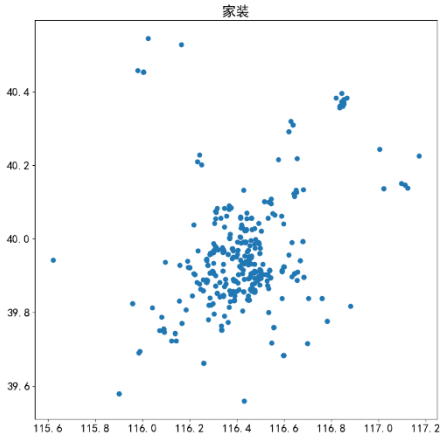


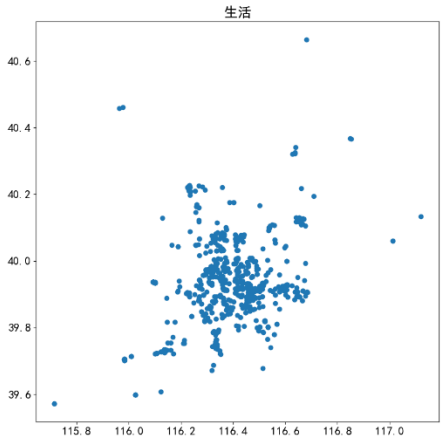


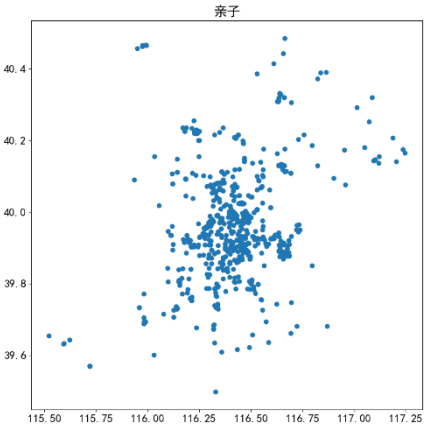


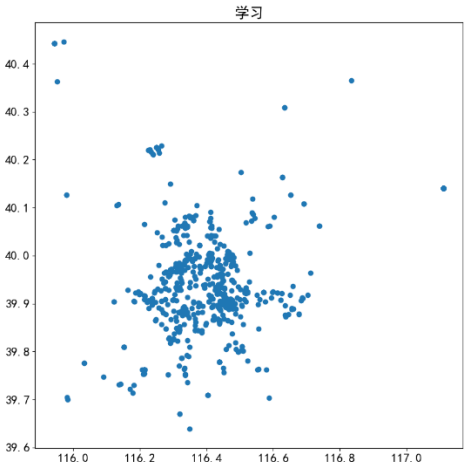


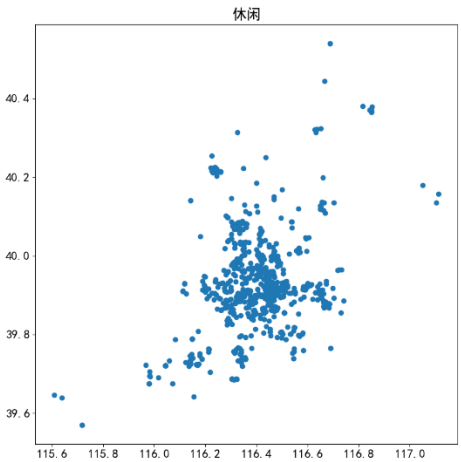

结婚
40.6
40.4
40.2
40.0
39.8
116.0
116.2
116.4
116.6
116.8
117.0
117.2


家装
40.4
40.2
40.0
39.8
39.6
115.6
115.8
116.0
116.2
116.4
116.6
116.8
117.0
117.2

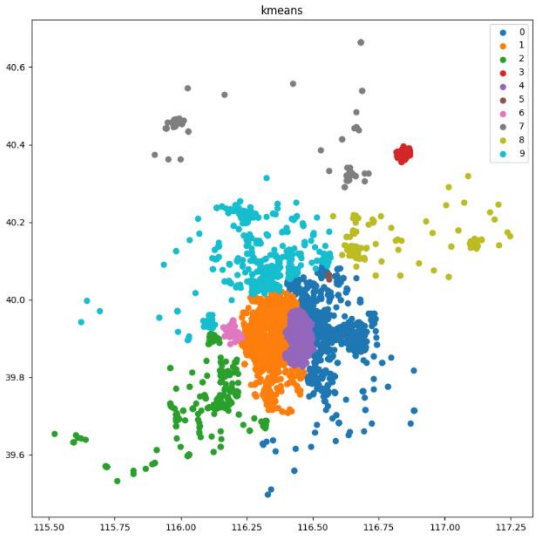

生活
40.6
40.4
40.2
40.0
39.8
39.6
115.8
116.0
116.2
116.4
116.6
116.8
117.0


亲子
40.4
40.2
40.0
39.8
39.6
115.50
115.75
116.00
116.25
116.50
116.75
117.00
117.25


学习
40.4
40.3
40.2
40.1
40.0
39.9
39.8
39.7
39.6
116.0
116.2
116.4
116.6
116.8
117.0


休闲
40.4
40.2
40.0
39.8
39.6
115.6
115.8
116.0
116.2
116.4
116.6
116.8
117.0

聚类结果如下:



可以发现北京的城市区域不能根据美团提供的 10 种 POI 分类进行功能划分，如此划分会导致部分类型的区域面积过小且并不具有明显区别于周边地区的特征，利用 ArcGIS 进行城市功能区域划分结果如下：

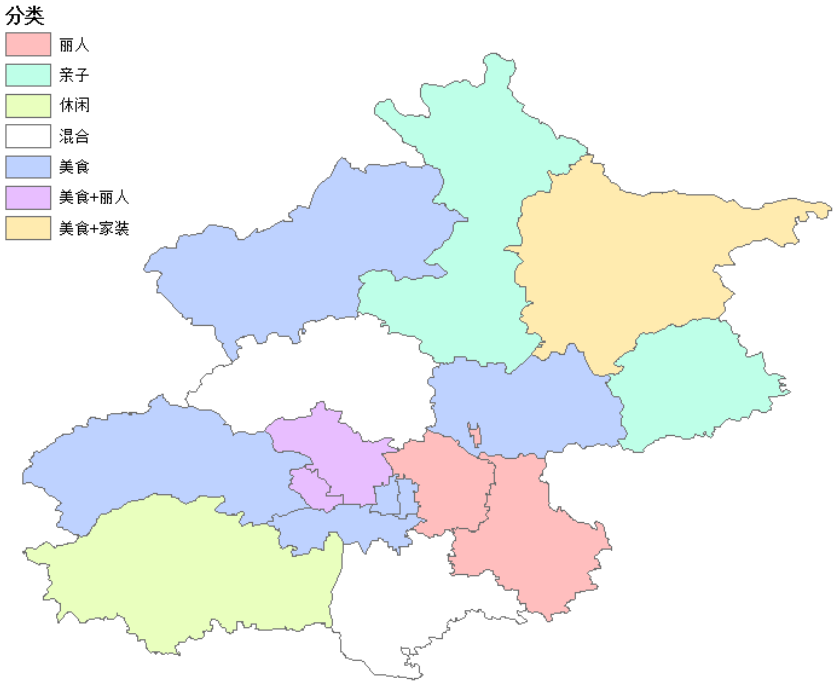
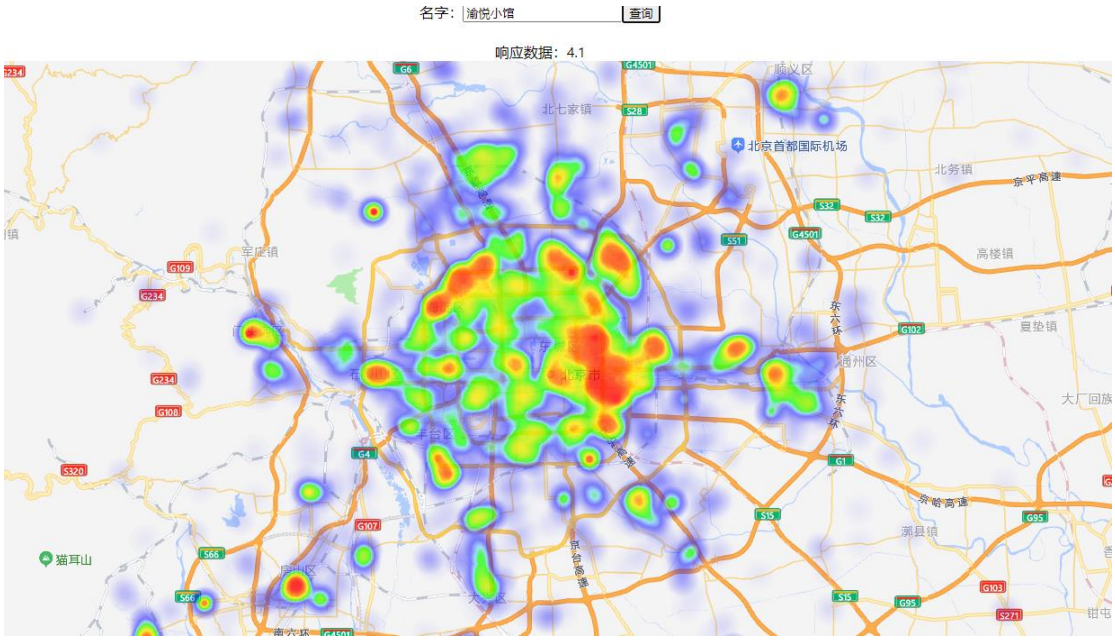


图 3 北京市城市功能区域划分

结果显示北京市仅有明显几种功能区（丽人、亲子、休闲、美食等），其余城市功能 POI 呈现混合分布状态，这与上述聚类结果类似。并且两种分析都得到了城市东北、西南等部分与其他部分承担不同城市功能的结论，结果较为类似。

同时也实现了基本的动态网络功能如下图：



4.实验总结

通过本次开放型实验，我了解了从数据获取到数据处理再到完成一个简易的互联网应用的全流程，也完成了从实验设计到进行实验再到撰写实验报告的全流程，对于网络应用层以及实验本身都有了更深刻的理解。

在数据爬取的过程中，我遇到了只能获取用户登录界面，html 解析失败等问题，通过助教的帮助和资料查找都较好地解决了。在后期的数据处理阶段，我通过已有的库对数据进行了简单的聚类分析并进行了可视化，但由于对于城市功能区划分的相关知识以及相关算法了解较少，分析的合理性和深度都有很多可以提升的空间。在构建互联网应用方面，我利用相关框架实现了基本的热力图绘制及 POI 评分查询的功能，但由于时间显示，没能实现输入经纬度匹配 POI 等功能，这些都有待进一步的实现。

但总体来说，本次实验我收获颇多，能够将之前学习到的一些方法进行应用，并且解决实验中遇到的问题。非常感谢助教和老师的耐心指导。