

安全高效的时空轨迹接触追溯系统

前言

云外包时空数据查询服务面临数据泄露、数据推断等安全威胁。时空数据包含用户行动轨迹、生活习惯等个人敏感数据，时空数据泄露与滥用将对个人生活造成极大困扰，对社会造成严重负面影响，阻碍时空数据要素资源的开发利用。

目前，基于隐私计算等相关技术可以达到“数据可用不可见”的目标。但是，由于这种技术采用同态加密、计算操作在密文上完成，计算的复杂度高、时间长的缺点。这就需要我们改进传统的比较算法 ($O(n^2)$)，从数据结构等角度出发，设计出减小关键算子计算量的算法。

使用和设计的数据结构

使用的基本数据结构

我们计划使用Java语言作为核心算法的开发语言。计划使用的基本数据结构包括：**数组、基于数组的List、哈希表、基于红黑树的TreeMap**。

我们一方面以这些基本数据结构作为类内的成员变量，设计合适的类结构和类关系。其中关键的类里面将拥有多个基本数据结构的成员变量。

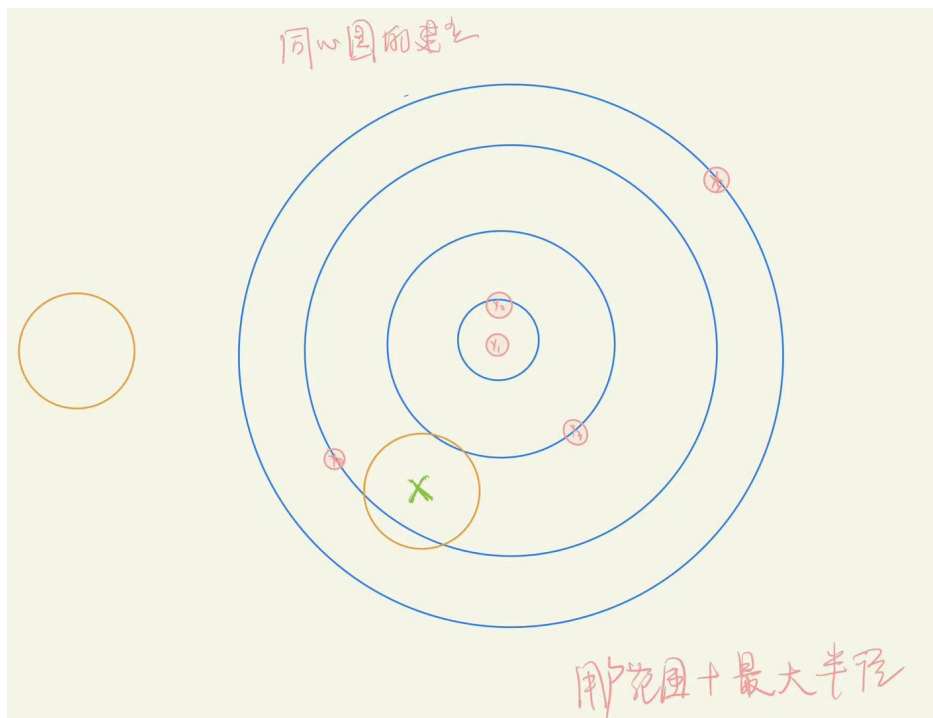
另一方面，我们以这些基本的数据结构为基础，设计出更高层次的数据结构。下面介绍我们设计的数据结构。

设计的高级数据结构

同心圆组

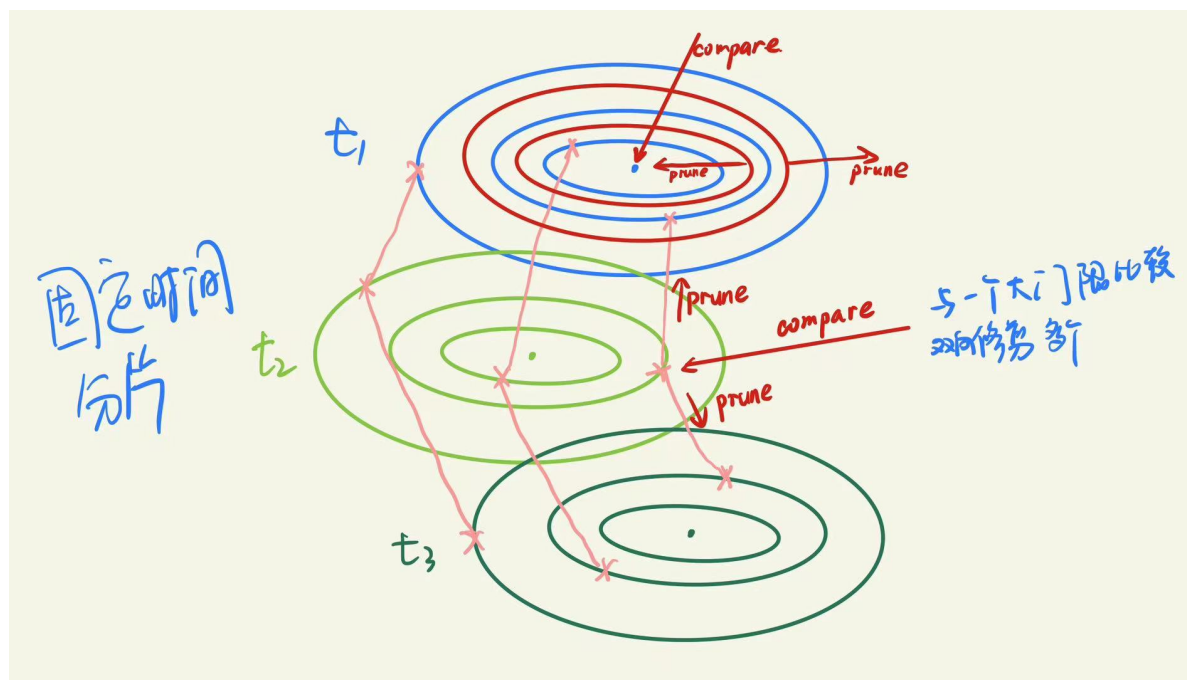
我们使用了ArrayList、HashMap、红黑树、这三个数据结构，设计出了能够在单一时刻下减少比较数量的同心圆组。

我们使用ArrayList存储所有时空信息，使用HashMap建立ArrayList的索引，使用红黑树排序并索引同心圆。基于这种数据结构，我们可以通过一次比较排除大量数据、我们称之为“一维裁剪”。



同心圆树

在上述同心圆组的基础上，我们考虑到轨迹信息里面包含的时空信息具有前后关联的特性，并且基于这一特性引入了**双向链表**的思想。每个同心圆组中的同心圆将存储与之相邻的同心圆的引用，建立起双向链表的结构，最终建立整个同心圆树。基于这一数据结构，我们能够在完成一次比较的时候同时尝试裁剪部分与之相邻的同心圆（一般位于不同的同心圆组中，即同心圆树的不同层），我们把这个称之为“**二维裁剪**”。



数据集

我们采用微软在Geolife 项目中收集的轨迹数据作为时空数据来源。

- **数据集：Microsoft Geolife GPS Trajectory Dataset**
- 这个 GPS 轨迹数据集是在 (Microsoft Research) Geolife 项目中由 178 个用户在四年多的时间里（从 2007 年 4 月到 2011 年 10 月）收集的。该数据集的 GPS 轨迹由一系列带有时间戳的点表示，每个点都包含纬度、经度和高度信息。**该数据集包含 17,621 条轨迹，总距离为 1,251,654 公**

里，总时长为 48,203 小时。这些轨迹由不同的 GPS 记录器和 GPS 手机记录，并具有多种采样率。91% 的轨迹以密集表示形式记录，例如每 1~5 秒或每 5~10 米/点。

- 该数据集记录了广泛的用户户外运动，不仅包括回家和上班等生活常规，还包括一些娱乐和体育活动，如购物、观光、餐饮、徒步旅行和骑自行车。

这个数据集符合我们算法的需求，同时具有真实、数据量大的特点，符合数据结构大作业的课程要求。

算法角度

数据预处理

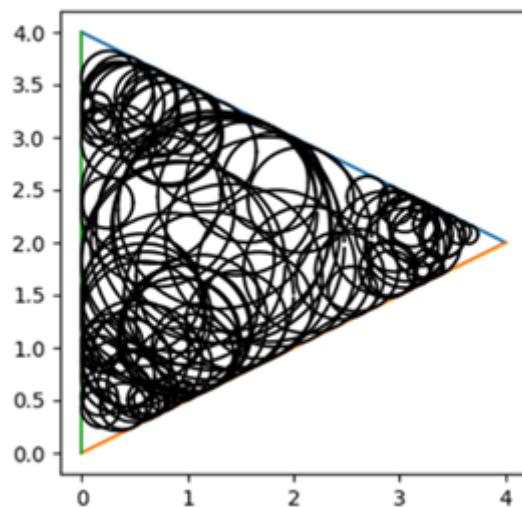
我们在读取数据集时会进行数据预处理，我们将时空数据中的时间信息分片，存储在**哈希表**中。这样在后续进行比较算法的时，可以利用哈希表做为第一轮的过滤。

范围的处理

对于不规则的范围，我们计划使用圆形这一基本图形拟合完成。我们设计了支持任意边数的**多边形拟合算法**。下图是使用Python尝试拟合的效果。

- 这个算法并不是随机生成圆形，我们设计了一种更偏向边角的拟合算法已达到更优的覆盖率。
- 在生成圆形的时候，我们会近似计算每个圆形对于整体拟合的贡献率。

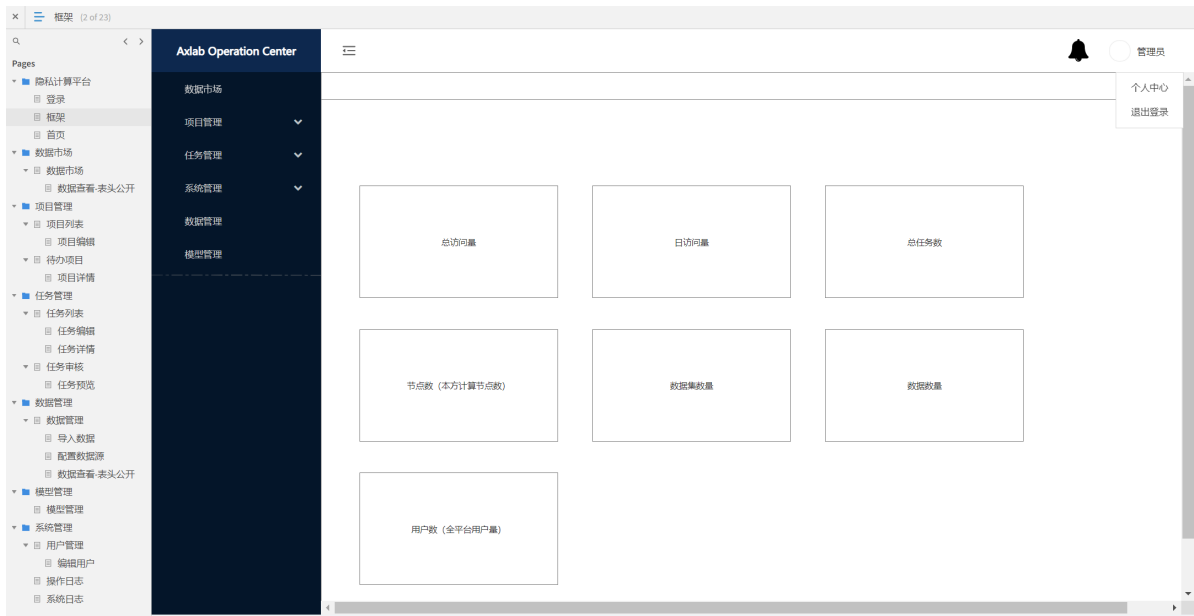
通过上面的策略以实现使用更少的圆获得更优的拟合效果。



预期成果形式

我们计划以**服务端和网页的形式**作为成果展示的主要形式，设计美观实用的前端交互界面。

我们将实现在密文状态下的时空轨迹接触追溯算法，并将**结果展示在网页中**（我们的前端界面将参考一些开源的框架设计，如下图）



同时为了展示，我们将**结合百度地图API，将数据可视化在地图中**（本图片截图自百度地图API）：



- **轨迹可视化。**在明文下展示所有或者某个的轨迹。
- **地图缩放功能。**可以对地图进行放大或者缩小。

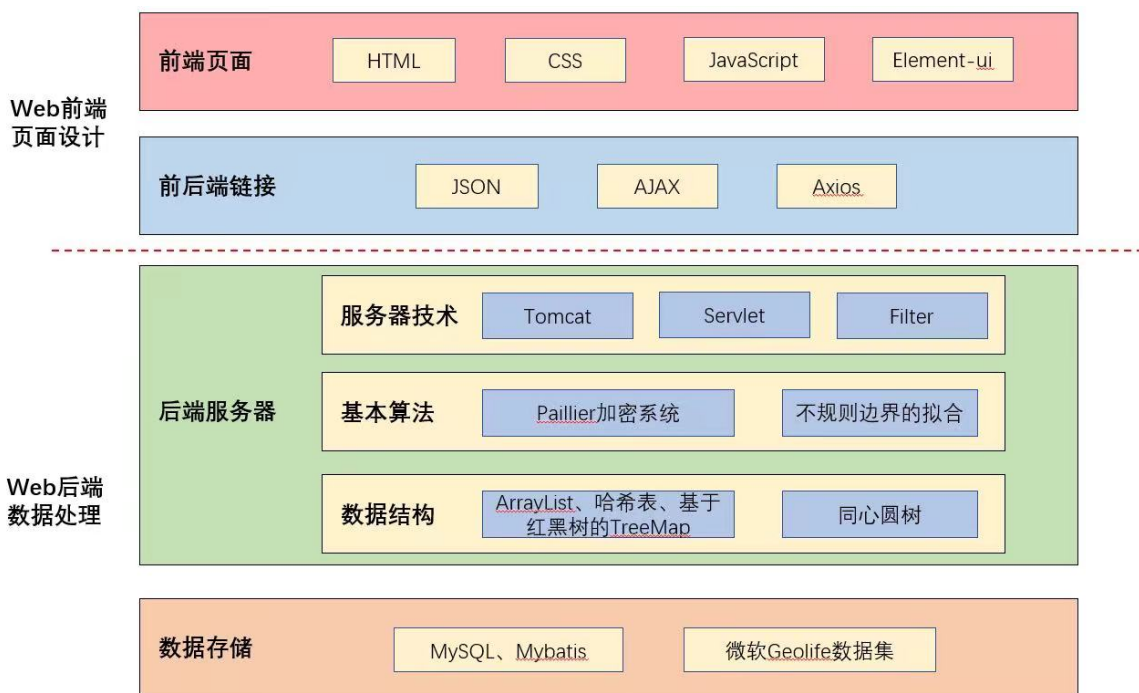
同时我们基于结果数据，进行**多角度的数据分析**。

- **区域密度分析：**分析某一时间段经过某一区域的人员及密度。
- **人员关联分析：**在多条轨迹中，分析提取相关人员的关联程度。

技术分析

系统技术架构

该系统使用了Java作为编程语言并应用了JavaWeb的技术。如下图展现了系统实现中涉及到的技术及其框架



前后端系统实现说明

后端系统实现说明

本系统使用的数据库是MySQL，通过持久层框架Mybatis简化、优化JDBC的开发：编写Mybatis配置文件和SQL映射文件，替换连接信息，统一管理SQL语句，解决传统JDBC开发中的硬编码问题。通过文件读入的方式，将数据集传输到数据库中，再通过系统功能分析完成对数据库的增删查改功能。

本系统使用的服务器是Tomcat，通过Maven导包能较好的在项目中安装相关插件。本系统涉及到权限管理，因此除了基本的动态资源组件Servlet以外，还增加了Filter组件，以区别用户和管理员对系统资源的访问权限。根据本系统需求，重写了各个组件的方法。

前端系统实现说明

本系统的前端网页界面使用基本的HTML，通过CSS文件控制组件的样式属性，并通过JavaScript语言实现对应组件的功能。实际开发的时候，我们使用了一套现成的组件库Element-ui，界面美观的同时也降低了前端开发的难度和工作量。

为了实现异步交互以及简化JavaScript开发，本系统使用了AJAX技术，使用HTML+AJAX优化传统的Servlet+JSP，完成与服务器的数据交换。在此基础上，我们使用Axios进行封装，进一步优化代码。本系统涉及到大量的实体类，于是我们采用JSON作为数据载体，实现Java对象和数据字符串的相互转换

在展示个人时空信息时，本系统调用了百度地图API，将用户的时空信息展示在地图上中，搭配API集成的组件，对地图进行标记说明。