

## 团队运行轨迹模型

### 一、模型概述

本模型旨在呈现 X 市旅行社团队以及带团导游在出团时间内的实时运行轨迹情况。通过收集、整合相关数据，以静态轨迹的形式在地图上直观展示全域内旅行社接待团队的当下位置、过往历史轨迹，同时提供团队详细信息，为旅游管理部门、旅行社、游客以及相关研究人员等了解旅行团队的行程动态、进行行程监管、辅助出行决策等方面提供全面且直观的数据参考。

### 二、数据来源与整理

**旅行社行程报备系统：**X 市各旅行社按照规定需向旅游管理部门报备旅行团队的出团计划，包括行程安排、出发时间、预计返程时间、游览景点顺序、交通方式、住宿地点等信息，这些构成了团队运行轨迹预测以及后续核对的基础框架，从中可获取团队计划的大致路线走向。

**导游移动端应用数据：**如今许多导游会使用专门的移动端应用来辅助带团工作，这些应用能够实时记录导游所在位置（通常借助 GPS 定位功能），同时还可上传团队的实时游览情况、停留时间等信息，是获取旅行团队实时位置以及行程进展情况的关键数据源。

**旅游交通票务系统：**涉及团队乘坐飞机、火车、长途客车等公共交通工具的票务信息，通过与交通运营部门的数据对接或旅行社提供的相关购票记录，能精准确定团队在交通环节的出发地、目的地、出发时间和到达时间等关键节点，进一步完善团队运行轨迹。

**景区及住宿场所登记数据：**团队进入景区、办理住宿时留下的登记记录，包含签到时间、离开时间等，有助于细化团队在各停留点的具体情况，使轨迹更加精准，也便于与其他数据源相互印证，确保数据的准确性。

**数据整理流程：**首先，将从上述多渠道收集到的数据汇总到统一的数据平台。然后，运用数据匹配算法，以团队编号、导游身份信息等作为关联字段，把分散在不同数据源中的同一团队相关数据进行整合匹配。接着，对整合后的数据进行清洗，去除重复、错误或逻辑不符的数据，例如位置信息异常跳动的数据点等。最后，按照时间顺序对数据进行排序梳理，为后续轨迹生成与分析做好准备。

### 三、核心算法与分析逻辑

#### 轨迹构建算法：

基于时间序列，按照数据整理后的顺序，提取每个时间节点对应的团队位置信息（经纬度坐标）以及相关行程事件（如到达景区、乘坐交通工具等）。

运用地理信息系统（GIS）中的路径规划算法，将相邻时间节点的位置信息进行连接，形成连续的轨迹线条，构建出旅行团队的运行轨迹，无论是实时的正在行进中的轨迹，还是历史已经完成行程部分的轨迹，都通过这样的方式逐步生成。

#### 轨迹优化与校准：

利用误差修正算法，比对不同数据源获取的同一时段团队位置信息，对存在偏差的数据进行校准，使轨迹更加贴近实际情况。例如，当导游移动端应用记录的位置与景区登记的团队进入位置有差异时，综合分析周边环境、交通线路等因素进行合理调整。

根据行程逻辑判断，对轨迹中不合理的节点（如不符合正常交通通行规则、景点游览顺序颠倒等情况）进行修正优化，确保整个运行轨迹符合旅行团队正常的出行和游览逻辑。

### 四、模型输出与可视化

**地图展示平台构建：**借助专业的地图可视化工具（如百度地图 API、高德地图 API 等）搭建可视化展示平台，以 X 市的地理地图为底图，根据团队运行轨迹的经纬度坐标，将轨迹线条绘制在地图上，实现地理空间上的直观呈现。

静态轨迹展示:

对于实时位置，用特定的图标（如不同颜色的圆形代表不同团队，图标闪烁表示正在移动等）在地图上标记出当前各旅行团队所在位置，方便快速定位查看。历史轨迹则以不同颜色、不同粗细的线条表示（例如，近期的轨迹线条颜色较鲜艳、较粗，较久远的相对暗淡、较细），清晰展示团队过往的行程路线，使其整个行程脉络一目了然。

**团队详细信息展示:** 通过在地图上设置信息弹窗功能，当鼠标点击团队对应的图标或轨迹线条时，弹出信息框，展示该团队的详细信息，如团队名称、导游姓名、出团时间、预计返程时间、已游览景点、下一站计划等内容，为用户提供全面深入的团队情况了解。

五、模拟数据示例与可视化展示

假设 X 市有以下一个旅行团队的模拟数据示例:

团队名称	导游姓名	出团时间	预计返程时间	行程安排
欢乐游团队	张导游	2024 年 12 月 1 日 8:00	2024 年 12 月 5 日 18:00	12 月 1 日: 从 X 市东区出发, 乘坐旅游大巴前往 A 景区, 10:00 到达 A 景区游览, 14:00 离开前往 B 酒店办理入住; 12 月 2 日: 9:00 从 B 酒店出发, 乘坐大巴前往 B 景区, 11:00 到达并游览, 17:00 离开前往 C 酒店办理入住; .....

通过导游移动端应用记录的实时位置数据（部分经纬度坐标示例）:

时间	经纬度坐标
2024 年 12 月 2 日 10:30	北纬 30.5°, 东经 120.3°（在前往 B 景区途中）
2024 年 12 月 2 日	北纬 30.6°, 东经 120.4°（已到达 B 景区）



```

# 创建轨迹数据的 GeoJSON 格式
geo_data = {
    "type": "FeatureCollection",
    "features": features
}

# 创建地图对象
m = folium.Map(location=[df['纬度'].mean(), df['经度'].mean()],
zoom_start=10)

# 添加时间戳轨迹层到地图
TimestampedGeoJson(geo_data,                                period="PT1M",
add_last_point=True).add_to(m)

# 保存地图为 HTML 文件
m.save("/mnt/travel_trajectory.html")

# 调用函数并传入数据文件路径
data_file_path = "/mnt/06.xlsx" # 请根据实际文件路径替换
create_travel_trajectory(data_file_path)

```

### 3. 代码解释

`pd.read_csv(data_file)`: 使用 pandas 库读取数据文件。

`pd.to_datetime(df['时间'])`: 将时间列转换为 pandas 的日期时间格式，以便后续处理。

通过循环遍历数据行，创建每个时间点的轨迹点信息，并将其存储在 features 列表中。

使用 `folium.Map` 创建地图对象，并设置初始中心位置和缩放级别。

`TimestampedGeoJson` 插件用于添加时间戳轨迹层到地图上，`period="PT1M"` 表示每个轨迹点之间的时间间隔为 1 分钟，`add_last_point=True` 表示添加最后一个轨迹点。

最后，使用 `m.save("/mnt/travel_trajectory.html")` 将生成的地图保存为 HTML 文件，可在浏览器中打开查看。

#### 注意事项

确保安装了必要的库，如 `folium`、`pandas` 等。可以使用 `pip install folium pandas` 命令进行安装。

数据文件的路径需要根据实际情况进行修改。

如果数据文件的格式不是 CSV，而是其他格式（如 Excel），需要使用相应的 pandas 读取函数（如 `pd.read_excel`）来读取文件。

这样就可以根据提供的数据生成一个带有时间戳的行程轨迹图，展示行程的路径和每个时间点的位置。

（此处应插入一个简单的地图示例，地图上显示 X 市地理范围，用线条和图标展示出欢乐游团队的历史轨迹以及当前实时位置，点击相应元素可弹出如上述详细信息的交互效果示例图）

## 六、应用场景与价值

**旅游管理与监管：**X 市旅游管理部门借助该模型，能够实时、全面地掌握各个旅行团队的行踪，便于对旅行社的带团服务质量、行程安排合规性等进行有效监管。例如，及时发现团队是否存在擅自更改行程、违规安排购物点等情况，保障游客的合法权益，规范旅游市场秩序。

**旅行社运营管理优化：**旅行社可以通过查看本社派出团队的运行轨迹和详细信息，更好地协调各团队之间的资源分配，如在遇到交通拥堵、景区临时限流等突发情况时，及时调整后续行程安排，优化车辆调度、导游调配等运营环节，提高服务效率和游客满意度。

**游客行程参考与保障：**游客或其家属可以通过模型查看团队的实时位置和历史轨迹，了解行程进展情况，在与团队暂时失联或对行程有疑问时，能够心中有数，增强出行的安全感。同时，也有助于游客提前规划个人在行程中的自由活动时间，更好地享受旅行。

**旅游研究与市场分析：**为旅游研究人员提供了丰富的实际行程数据，通过分析大量旅行团队的运行轨迹、停留时间、游览顺序等信息，可以深入了解 X 市旅游资源的热门程度、游客的出行偏好以及不同线路的受欢迎程度等，为旅游产品开发、线路设计优化以及旅游市场趋势研究提供有力的数据支撑。