

基于多级分区的路线搜索算法

Qi Hu

2019 年 12 月 26 日

1 简介

结合GIP CUP其他队伍的算法和看过的论文，对STP方法进行优化，采用新的基于多级分区的路线搜索算法。将研究区域根据不同的分区方法划分为三级区域，对于空闲出租车，依次选取每一级的目的地，第三级分区即Intersection作为目的地，采用最短路径算法得到搜索路线。在选择目的地时同时考虑当前的整体需求情况和目的地的距离，提升了算法的表现。

2 基本原理

通过对其他队伍的方法的仔细研究和对比分析，我们发现STP算法存在着如下不足：

- 采用圆环对目的地进行过滤没有考虑目的区域之间的相似性，同时圆环内外径的大小难以准确确定，可能会排除较近的热门区域。
- 区域划分的过细，一共有150个区域，很多区域之间的需求模式都是相似的，可以从整体上进行考虑。
- 根据时间段来划分高峰期和非高峰期的方法不够精确，难以应对突发事件。同时在选择目的地的时候没有考虑距离的影响。

针对上述不足，我们提出了一种新的基于多级分区的路线搜索算法。该方法的基本原理是，首先基于研究区域的整体订单分布密度将其划分为四个一级区域，每个一级区域内可以认为需求模式是相似的，从而作为一个整体进行建模。之后采用STP的聚类方法划分二级区域，将二级区域根据其聚类中心点的位置将其归到相应的一级分区中，用于对一级区域内的模式进行细化。然后取每个二级区域内的Intersection(不包括度为2的Intersection)和以

该Intersection为起点的roads作为第三级区域，用于确定区域内的热门目的地。在路线规划时依次选择第一、二、三级区域，得到最终目的地。

同时，相比STP，新方法摒弃了判断是否是周末以及按照时间来判断是否是高峰期的做法。而是根据预测模型得到的每一天的实时需求，根据预测的未来timeHorizon时间内平均每分钟的资源总数将供需情况分为三类，即undersupply, oversupply和supply-demand banlance。针对每一种情况采用不同的应对措施，例如当处于高峰期(undersupply)时，采用区域的订单密度作为权重，更多的让车去往资源密集的区域，而选择性的忽略相对冷门区域的资源，以减少搜索时间。

最后，该方法在考虑区域权重(热门程度)的基础上，增加了对于距离的考虑，对距离当前位置较近的区域给予较大的被选择的概率，通过一个距离衰减系数 γ 来控制，其大小与供需状态有关。

3 方案介绍

对方案的介绍依旧按照数据模型和路线规划两个模块来进行，这里我们默认已经有了资源预测模型，不再按照STP取平均值的方法进行预测。

3.1 Data model

3.1.1 三级区域划分

Data model首先做的仍然是利用KMeans聚类对区域进行划分，得到K个子区域，记每个区域的中心点为 $r_i, i = 1, 2, 3 \dots k$ 。

按照约定俗成，我们将曼哈顿分为上中下城，中城是商业中心，面积最小，但却是订单分布最多的区域，即最繁华的中心城区。上城占了曼哈顿的大部分区域，但是上城越往北，订单分布的越稀疏，而南部靠近中央公园的区域，订单分布的又较为密集。为了将区域作为一个整体进行考虑，我们将上城区再分为两部分，即南部的热门区域和北部的冷门区域，从而得到四个一级区域，从北到南依次记为 T_1, T_2, T_3, T_4 。

对每个 r_i ，我们按其所在位置属于哪个一级区域，将 r_i 对应的区域分到相应的一级区域中，从而得到前两级的分区，再对每一个二级区域，将内部的每一个Intersection和以该Intersection为起点的road作为一个区域，得到第三级区域，建立起三级区域划分。

3.1.2 需求状态确定

统计从当前时刻 t 到time horizon时刻 $t + \Delta$ 内的预测资源总数, 记为 s , 若 s 大于一定的阈值 θ , 则认为是undersupply(高峰期)。例如, 假设有5000辆出租车, time horizon长度为40分钟, 每趟trip的平均时间为15分钟, 假设在高峰期cruise time加approach time只需要3分钟, 则每辆车完成一趟trip需要18分钟, 因此18分钟最多满足5000个requests, 平均每分钟为277.8个requests, 可以定义若未来40分钟内的平均每分钟的资源总数大于280时, 则当前需求状态为undersupply。类似的可以确定oversupply的状态, 处于两者之间的则为supply-demand banlance。

3.1.3 各级区域的权重矩阵

对于第一级区域, 我们以行表示区域, 列表示时间片, 计算得到区域权重矩阵 W_T , 区域权重的定义依然采用STP的 $p - \lambda d$ 的方式, 在第一级区域中 λ 取0.3。值得注意的是当需求状态为1, 即处于undersupply状态下时, 将区域权重直接定义为上车点数除以区域内的路段总长度, 即上车点的密度。更多的让车驶向资源密集区域而不考虑可能造成的羊群效应, 因为高峰期不存在羊群效应, 所有车都能找到乘客。

第二级区域采用和第一级区域类似的方法计算权重矩阵, 得到的权重矩阵记为 W_R 。 λ 在三种需求状态下分别取0.2, 0.5, 0.6, 在高峰期时同样采用除以区域内路段总长度的方法定义权重。

第三级区域由于每个区域内数据的稀疏性, 因此将时间片取大, 我们将一天按小时分成24个时间段, 计算每一个时间段内各区域的权重。

3.2 Plan search route

3.2.1 第一级区域的选择

第一级区域的选择采用与STP相同的方法, 找到 W_T 中对应当前时间 t 的那一列, 记为 j , 采用轮盘赌算法从中选择一个区域, 即对于区域 m , 被选中的概率为 $\frac{W_{mj}}{\sum_{i=1}^4 W_{ij}}$ 。

3.2.2 第二级区域的选择

第二级区域首先按照与上面相同的采样方法从第一级区域包含的二级区域中选出 N 个candidate regions, 记为集合 S , 然后对于每个 $s \in S$, 以 $dist(l, s)^\gamma$ 作为权重, 其中 $dist$ 为距离函数, l 为当前位置, γ 为衰减系数, 在三种需求状态下

取值分别为-0.2, -0.5, -0.6。每个candidate region被选中的概率为 $\frac{dist(l,s)^\gamma}{\sum_{s' \in S} dist(l,s')^\gamma}$ 同样采用轮盘赌算法选出最终目的区域 r 。

3.2.3 第三级区域的选择

如果当前时刻为非高峰期(状态2和3),则选择区域的中心Intersection作为目的地;如果为高峰期(状态1),则计算第二级区域内所有三级区域的权重,选择权重最大的Intersection作为目的地。

采用当前位置到目的地的最短路径作为搜索路线。

4 12月22日更新

- 将曼哈顿按行政区划分为下城区、中城区、上东区、上西区和哈林区5个一级区域,与之前相比将上城区分为上东区和上西区,因为中央公园的分割导致这两个区域在空间上不连续,这样划分更好解释。
- 将密度的定义中除以区域总长度改为除以区域的所有路段的总的旅行时间,这样计算的结果更加准确,更能反映在区域能找到乘客所需的时间,算法的效果会有一些提升。
- 设Agents Number为 N , Average Trip Time为 T_1 , Average Search Time为 T_2 , Time Horizon为 Δ , 平均每分钟出现的资源数为 S , 则高峰期满足的条件可以定义如下:

$$S \geq \alpha \left[\left(\frac{\Delta \times N}{T_1 + T_2} \right) / \Delta \right] \quad (\alpha > 1) \quad (1)$$

由于 T_1 和 T_2 是不确定的,这里我们暂且认为每一天 T_1 和 T_2 的变化不会太大,可以看做一个常量,则对于给定的 Δ 和 N ,我们需要通过调参得到一个最优的 α 值。oversupply的情况也可以用相同的方式定义。

5 12月26日更新

- 将五个行政区中的上东区和上西区合并为一个区域,依然分为4个一级区域,因为上东区和上西区的出行模式基本相同。
- 在第三级区域的选择过程中,若为非高峰期,不直接选择center Intersection作为目的地,而是采用与前两级区域相同的采样的方式得到目的地。高峰期时仍然选择最热门的Intersection作为目的地。