会议纪要

2021/10/25

1. 汇报内容
2. 数据生成：

生成网格状路网🡪初始化每条路径的权重🡪在每条路上增加随机扰动🡪在新的加权图上随机选取两点🡪计算两点之间的最短路径作为生成的路径p(s,v)。

老师建议的方法：

固定两对起始点可分布在上下或左右对侧，随机生成路径，使得大部分路径分布在上或下，只有部分路径横穿过上下。

设定左右两点作为两个城市，两点之间只存在一条通路，在两个城市周围分别做随机游走。使生成的路径大部分遵循：随机，固定，随机的模式

这样的两个方法可以在路径集合中构建出所谓的popular path，这样方便炎症算法结果的正确性。

1. 复现代码

基本复现了动态规划的大部分代码，但是并未更新trajectory集合T，并不断迭代使用DP，求得最优值。

（Ps:在每次动态规划生成子问题时，需要考虑不切割的情况。此处是论文勘误。）

通过分析输入集合较小的情况，计算得DP的时间复杂度为O(kn2)，此处可凭借XX equation进行验证。

1. 问题？

如何验证代码输出结果的准确性？

可以通过在生成数据集的时候构建所谓的popular path，在输出结果中观察是否将该popular path找出来

该输出怎样的结果？

首先我们并不是将所有的pathlet均输出，而是找出Top-k的作为dictionary，观察我们能用这些路径重建多少路径。因为本问题实际是压缩问题，需要用尽可能少的信息保存尽量不损耗的消息。

其次算法也应当限制maximal\_pathlet\_length，可以减少算法的时间复杂度。但输出结果是否是less-optimal的需要证明，或者在算法运行过程中观察是否能保留部分重要的长路径，但过滤不重要的长路径。

每次都只对一条路径做动态规划是否会造成local optimal的问题？

并不会，首先在DP时我们已经考虑T(p)，这其实已经对其他的路径分割造成了影响。

更新过程更像是每次先固定所有的路径assignment，去求得在此路径集合上的最有dictionary（pathlet）；然后我们在固定所有的pathlet，即dictionary，去更新所有的路径assignment。

该问题还可以通过最短路径和trajectory集合更新的方法来求解。

具体求解如下，每次都将多个边的组合看作新的边，并计算他的T\_value，然后在改造后的图上计算不同点对之间的最短路径，就可以视作当前的最优分割，然后我们更新T\_value，反复迭代计算直至收敛。

但是以上两种方法是否等价未经证明。并且每次再将组合后的新增加的边直接加入trajectory集合还是将上一步的边代替掉，还需要思考。

1. 下一步计划

代码复现

需要在动态规划或最短路径搜索后结束后更新T集合，但是更新T集合的方式需要考虑，分为以下三种：

1. 直接增加
2. 代替上一次迭代的边
3. 将上一次的边与此次更新的边做线性组合等

对比以上两种算法是否等价

对于程序运行结果的验证：

1．通过可视化的方式，观察是否将大多数的路径和popular path都表示出来了

2．观察生成的dictionary是否能对新加入的路径做合适的decomposition。

证明算法是否可以收敛，并收敛到正确的最优解。

研究是否能将现在的整数规划做松弛，变成一个近似的实数域上的优化问题，如果可以我们就可以使用ADMM进行编程求解。