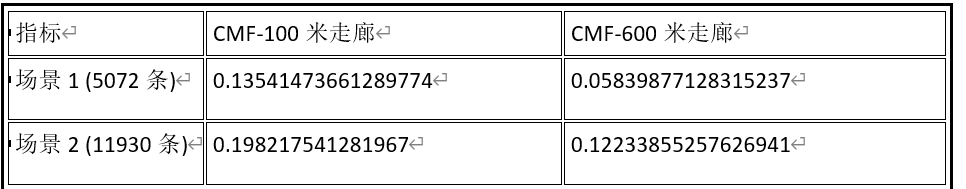
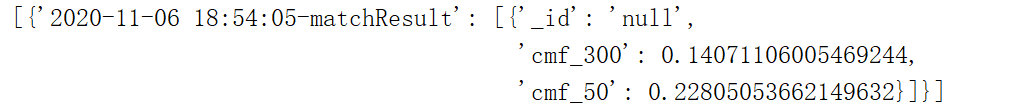
之前版本：

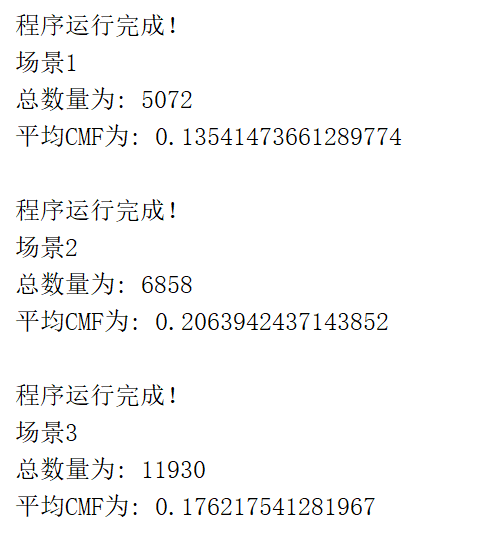


#----------------------------------------------------------------------

11月6日的结果，1w5的CMF50为0.22

新划分场景后1w2为0.17





#----------------------------------------------------------------------

12月5日，只添加了删除操作，得到的结果：

无论是综合CMF还是场景1的CMF，删除系数越大，CMF值越低，

但部分加了删除操作的好于不加删除的CMF值0.176217，加了删除的最低的综合可以达到0.170572

场景1的不加删除CMF值0.135414，加了删除的最低的场景1为0.1367101，略微变坏一点点

删除系数为AB+BC>删除系数倍的AC就删除B点

删除系数为：5 'cmf\_50': 0.17057215951530616,'cmf\_300': 0.08975642224999672,

场景1：'cmf\_50': 0.13671017815408004,

场景2：cmf:0.19751056526336663'

#-----------------------------------------------------------------------------

同时挑选测试了几组同时带惩罚与删除的操作

（此时全局惩罚为0，局部惩罚为50）

这里只测试了惩罚阈值和删除阈值的组合

可以看到，在惩罚阈值为1.15，删除阈值为2.0时，无论是全局CMF还是场景1的CMF，都比之前的最优要得到好，可以继续调试惩罚阈值与删除阈值、全局惩罚和局部惩罚的组合

不加惩罚与删除的CMF值0.176217，加了惩罚与删除的最低的综合可以达到0.165884

场景1的不加惩罚与删除CMF值0.135414，加了删除的最低的场景1为0.133931

12月5日

惩罚阈值:1.15 删除阈值:2.0 惩罚的轨迹数量:10537 删除的轨迹数量:6251

'cmf\_50': 0.16588420576367904, 'cmf\_300': 0.09058359020202239,

场景1：cmf\_50': 0.13393000757846532,

场景2：cmf:0.1913049024111849

--------------------------------------------

12月23日

调整了下界 ，循环保证下界数量满足要求

|  |  |  |  |
| --- | --- | --- | --- |
|  | 整体情况' | 场景一 | 场景二 |
| cmf\_50 | 0.160972 | 0.1364269 | 0.1815027 |
| cmf\_300 | 0.08232 | 0.0517 |  |
| count | 10798 | 4918 | 5880 |

整体情况'：

cmf\_50': 0.160972,

cmf\_300': 0.08232,

count: 10798

场景一：

cmf\_50': 0.1364269,

cmf\_300': 0.0517,

场景二：

cmf\_50':0.1815027

--------------------------------------------------------

12月24日

调整了将候选点按照垂直距离排序，选择前num个

|  |  |  |  |
| --- | --- | --- | --- |
|  | 整体情况' | 场景一 | 场景二 |
| cmf\_50 | 0.1573856403788031 | 0.13052147590861884 | 0.1798546813421307 |
| cmf\_300 | 0.08474301535635576 | 0.0551002227430759 | 0.1095360857767827 |
| count | 10798 | 4918 | 5880 |

整体情况'：

cmf\_50': 0.1573856403788031,

cmf\_300': 0.08474301535635576,

count: 10798

场景一：

cmf\_50': 0.13052147590861884,

cmf\_300': 0.0551002227430759,

count：4918

场景二：

cmf\_50': 0.1798546813421307

cmf300:0.1095360857767827

count：5880

---------------------------------------------------------------

12月26日

将参数分场景，场景一无改善，场景二略有改善，使用了删除阈值为4，删除阈值为3在测试中

|  |  |  |  |
| --- | --- | --- | --- |
|  | 整体情况' | 场景一 | 场景二 |
| cmf\_50 | 0.15660408533395076 | 0.13052147590861884 | 0.17842087055258662 |
| cmf\_300 | 0.08396557594417023 | 0.0551002227430759 | 0.10810925138493453 |
| count | 10798 | 4918 | 5880 |

整体情况'：

cmf\_50': 0.15660408533395076,

cmf\_300': 0.08396557594417023,

count: 10798

场景一：

cmf\_50': 0.13052147590861884,

cmf\_300': 0.0551002227430759,

count：4918

场景二：

cmf\_50': 0.17842087055258662

cmf300: 0.10810925138493453

count：5880

---------------------------------------------------------------

12月31日（变差变慢）

分出部分候选点随机分配给更远的路段，以避免集中在基站附近

分出10%的点

|  |  |  |  |
| --- | --- | --- | --- |
|  | 整体情况' | 场景一 | 场景二 |
| cmf\_50 | 0.16223291241686277 | 0.13856695523982393 | 0.1820269902054133 |
| cmf\_300 | 0.08766802206900441 | 0.06040085423758620 | 0.1104741328504525 |
| count | 10798 | 4918 | 5880 |

整体情况'：

cmf\_50': 0.16223291241686277

cmf\_300': 0.08766802206900441

count: 10798

场景一：

cmf\_50': 0.13856695523982393

cmf\_300': 0.06040085423758620

count：4918

场景二：

cmf\_50': 0.1820269902054133

cmf300: 0.1104741328504525

count：5880

---------------------------------------------------------------

1月2日（场景一变好，整体变差（很奇怪！），主要体现在数量变化，场景一变少了，但是总数量变多了）

对split中的Mee去重复

|  |  |  |  |
| --- | --- | --- | --- |
|  | 整体情况' | 场景一 | 场景二 |
| cmf\_50 | 0.1598630825841187 | 0.12733709063087917 | 0.18302212089369957 |
| cmf\_300 | 0.08662348497196878 | 0.05283010213185193 | 0.1106849263241564 |
| count | 10885 | 4527 | 6358 |

整体情况'：

cmf\_50': 0. 1598630825841187

cmf\_300': 0. 08662348497196878

count: 10885

场景一：

cmf\_50': 0.12733709063087917

cmf\_300': 0.06040085423758620

count：4527

场景二：

cmf\_50': 0.18302212089369957

cmf300: 0.1106849263241564

count：6358

----------------------------------------------------------------------------

1.12 不加绕路检测的结果：

综合：0.19574187680980948 'count': 10799

场景一：0. 1767170804589263 'count': 4919

场景2的cmf: 0.21165734845 'count': 5880

1.18 加绕路检测以及一些trick结果：

综合：0.15661962424806017 'count': 10798

场景一：0.13051804075308168 'count': 4917

场景2的cmf:0.1784426961822226 'count': 5881

------------------------------------------------------------

2.27 使用学姐的模型获取候选（过拟合）结果

综合：0. 1448676773168797 'count': 10749

场景一：0.12495533002945716 'count': 4909

场景2的cmf:0. 16160564184324222 'count': 5840

------------------------------------------------------------

3.4 更新维特比算法：

综合：0.15737649591457117 'count': 10619

场景一：0.12947657755593284 'count': 4813

场景2的cmf: 0.1805046921012963 'count': 5806

3.5 更新维特比算法（跑了参数：惩罚0.4删除0.6）

综合：0.15357405685306727 'count': 10798

场景一：0.12939627589484615 'count': 4917

场景2的cmf: 0.17378867153961264 'count': 5881

第二次测得：

综合：0. 15739257674682067 'count': 10798

场景一：0.1331176507034937 'count': 4917

场景2的cmf: 0.1776884127194509 'count': 5881

3.7 普通维特比（带trick的）

综合：0.15749668176646694 'count': 10798

场景一：0.12997372133878227 'count': 4917

场景2的cmf: 0.18050814179417066 'count': 5881

3.9 带拓展点维特比（1.增加了折扣因子，对距离进行折扣（不能太偏心，也不能吃亏）；2.更改了维特比相连为相加，更改了sigma与beta参数为600和0.3）

综合：0.14891861998330946 'count': 10798

场景一：0.12030987093654129 'count': 4917

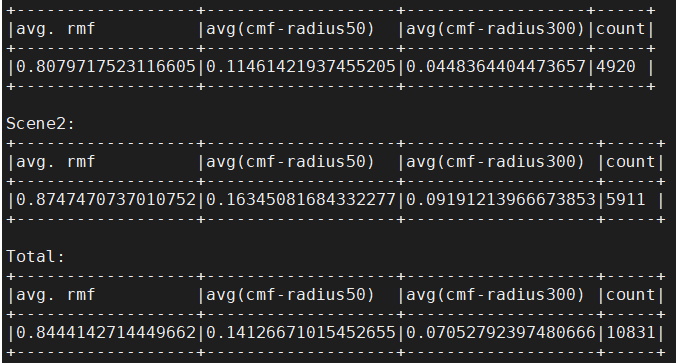
场景2的cmf: 0.17283788865580718 'count': 5881

4.9 使用历史概率+发射概率（距离）来获取候选，使用历史概率+发射概率（1:1相加）作为新的发射概率（历史概率与距离得到的发射概率没有做归一化！）

综合：0.14119429553833987 cmf\_300': 0.0704483844697904 'count': 10799

场景一：0.11461421937455205 cmf\_300':0.0448364404473656 'count': 4920

场景2的cmf: 0.16343855046704134 'count': 5881



发射概率部分 是否使用shortcuts 命中率

只使用距离 否 0.8726989283126676%

只使用历史概率（只选择历史中出现的边） 否 90.8%

结合历史概率与距离概率（只选择历史中出现的边） 否 89.8%

只使用距离 是 92.9669016526854%

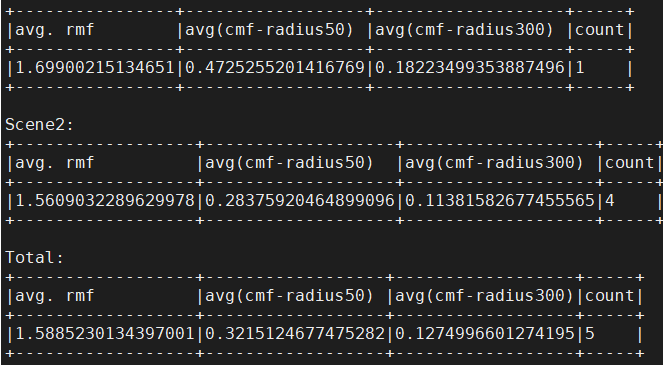
使用距离与历史概率 是 96.51846688938407%

使用GCN模型结合距离 是 95.6140923121294%

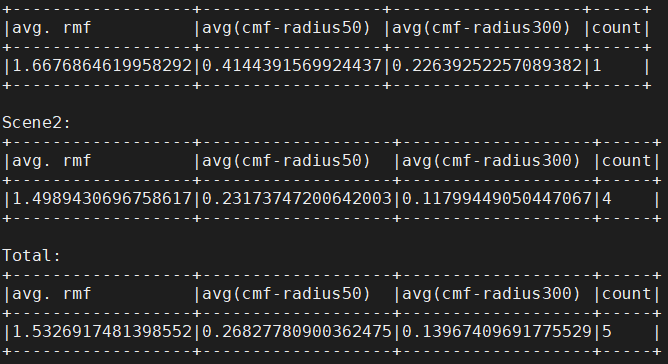
只使用GCN模型 否 77.4385703648548%

5.0 在5条数据上测试（这里写了bug，返回了大范围所有的候选）

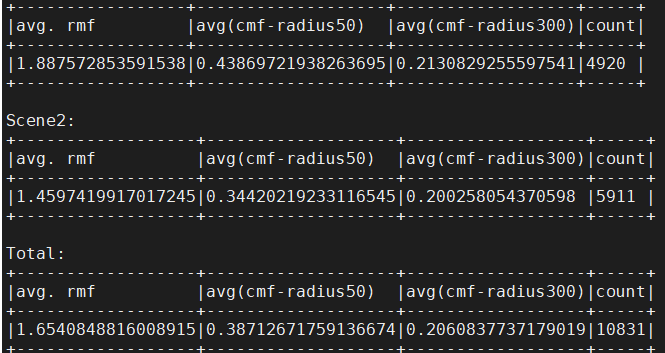
使用未经训练的模型



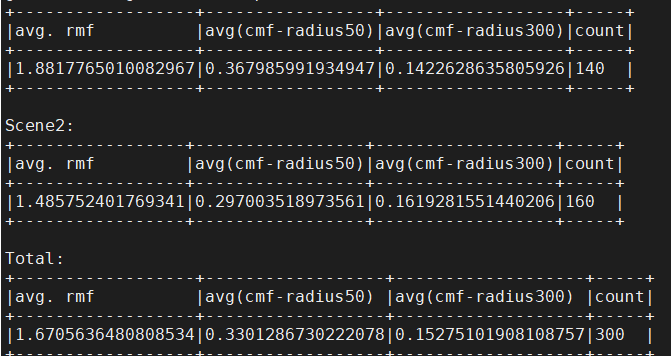
使用训练过后的模型



1w条训练后的模型，使用标准化+归一化，效果不佳

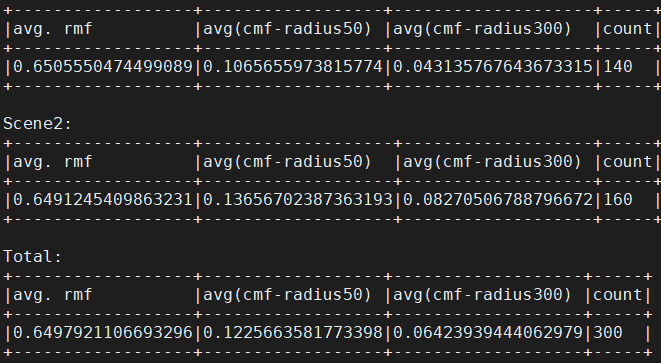


300条结果

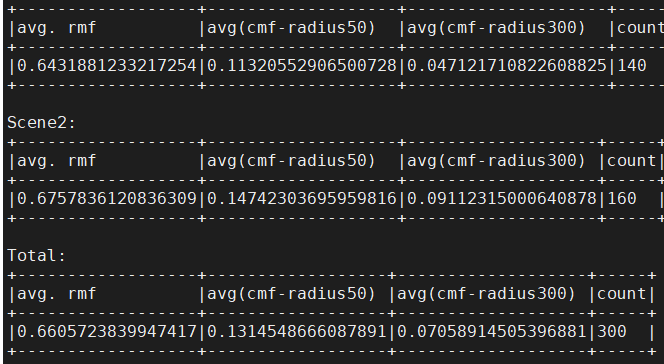


5.1 修复bug，使用模型+距离评分1:1，使用标准化统一量纲（效果更优）

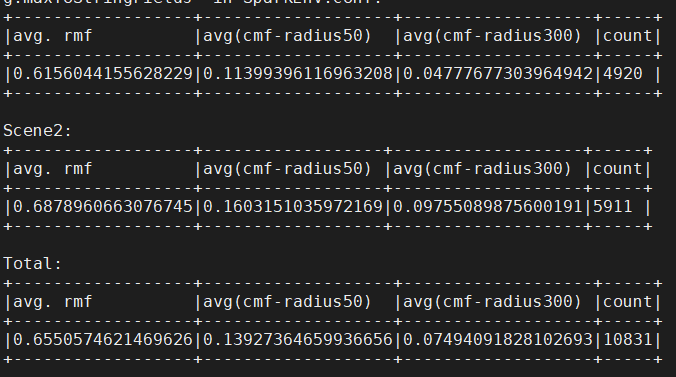
300条



使用模型+距离评分1:1，再测试标准化+归一化



5.2使用模型+距离评分1:1，再测试标准化，1w条数据



5.2 使用模型+距离评分，标准化，参数搜索比例

Model\_factor distance\_factor

0.5:0.5 cmf-50=0.4120

2:0 cmf-50= 0.4120

1:0 综合cmf-50=0.4118 场景一cmf-50= 0.4467

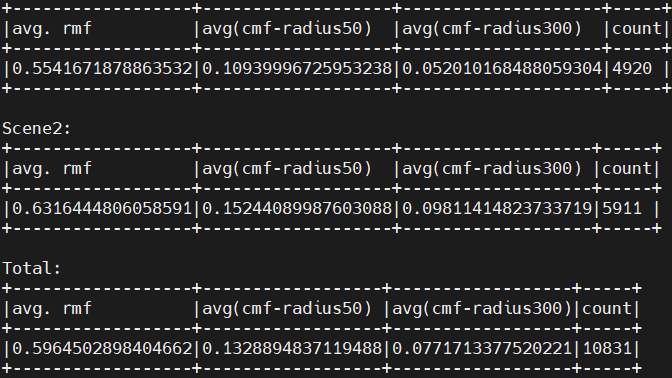
0.5:1 综合cmf-50=0.1258 场景一cmf-50= 0.1110

1:1 综合cmf-50=0.1225 场景一cmf-50= 0.1065

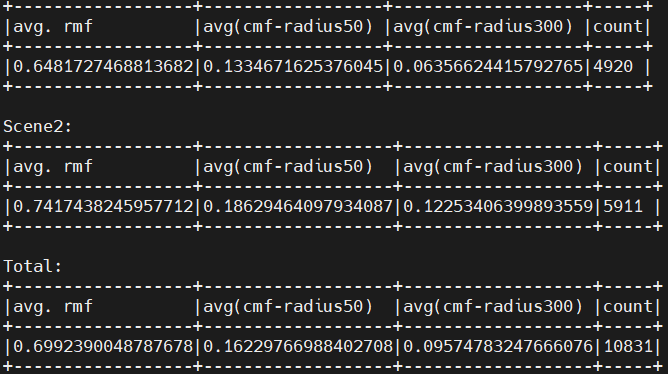
1.5:1 综合cmf-50= 0.1350 场景一cmf-50= 0.1169

1:1.5 综合cmf-50= 0.1246 场景一cmf-50= 0.1080

6.1 使用异构图+cross Attention的模型



7.1使用以基站为中心的attention的结果（变差，需要调参）



7.1.1

1. 300 条 只有distance，不进行标准化

场景一CMF50：0.11766902857119993

总的CMF50：0.13923929075436828

2. 300 条 有distance，不进行标准化 ，加上标准化后的模型评分 1:1

场景一CMF50：0.28919490520040214

总的CMF50：0.2869967044570972

3. 300 条 有distance，进行标准化 ，加上标准化后的模型评分1:1

场景一CMF50：0.12738634581395

总的CMF50：0.1463046147172148

4. 300 条 有distance，进行标准化 ，加上标准化后的模型评分0.5:0.5

场景一CMF50：0.12781828154981797

总的CMF50：0.14662973601684592

5. 300 条 有distance，进行归一化 ，加上归一化后的模型评分0.5:0.5

场景一CMF50：0.13817745683818763

总的CMF50：0.15030357448195966

6. 300 条 有distance，不变，加上放缩后的模型评分0.5:0.5

场景一CMF50：0.13612536192056263

总的CMF50：0.148355391040516

7. 300 条 无distance，加上模型评分

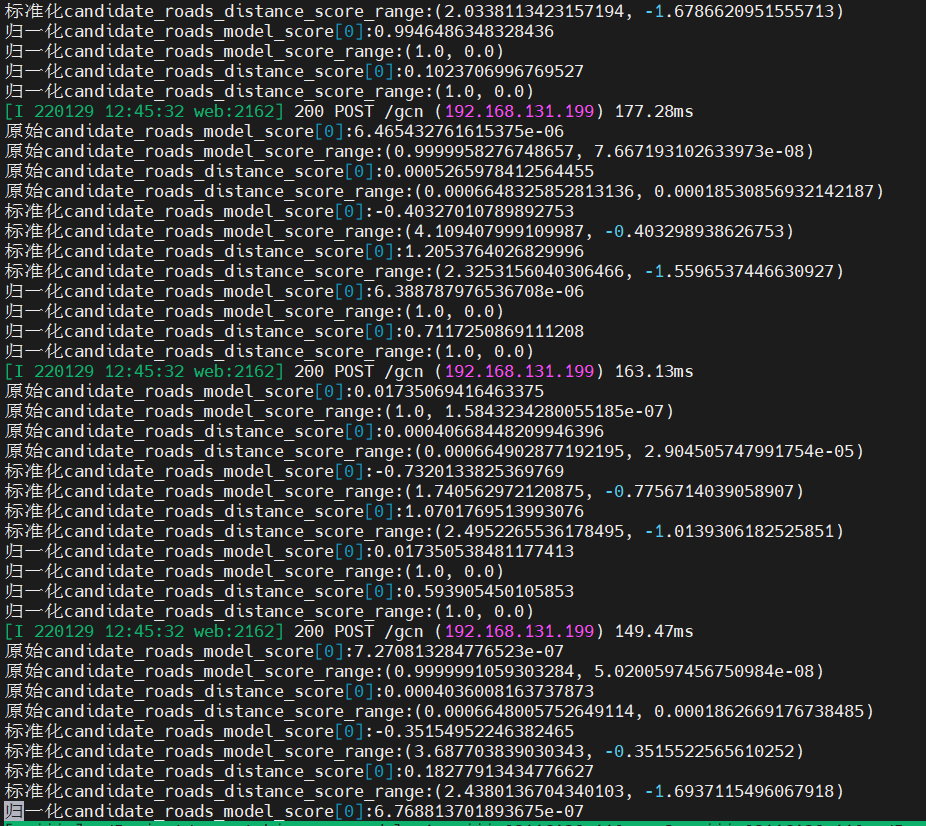
场景一CMF50：0.43321536260788723

总的CMF50：0.3921720165903526

8. 300 条 有distance，加上模型评分 0.5:0.5

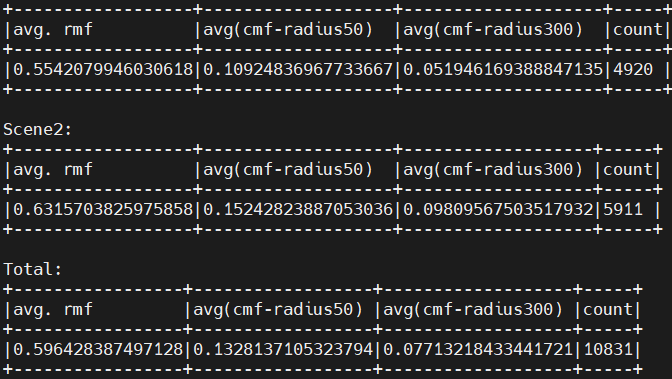
场景一CMF50：0.27276081572905336

总的CMF50：0.2786784146811471



8.1

1.使用第一个模型+距离评分0.5:0.5 原始shortcuts



2.使用第一个模型+距离评分0.5:0.5 无shortcuts