# GPS 定位实验报告

邱能 21307130310

## 一、PJ 介绍：

**任务：**给定高采样的轨迹序列（采样频率在 10s~15s 间）和路网的空间位置和拓扑结构，求得每个轨迹点实际所落在的路段号。

**准确率：**答案返回的路段序列记为$\hat{R}$，官方的结果序列为$R$. 那么准确率为$\sum_{i=1}^{|R|}|\hat{r_i}=r_i|/|R|$。

**评分方式：**本次 PJ 主要考察匹配的效率。在达到平均准确率为$k$的前提下，速度越快的程序分数越高。(暂定$k\in\{97, 93, 90, 70\}$，即如果平均准确率没有达到80，则可能无法得到分数)

## 二、重要内容的实现：

本次 PJ 使用 Hidden Markov map matching through noise and sparseness 论文中所给的隐马克洛夫模型方法实现，重要部分在于计算点到一个路段的最近点，路网上两点间的距离，已经状态转移等。分别就重要部分的实现进行阐释。

### 1. 计算轨迹点 p 到道路 r 上的最近点

道路 r 可以看做是由一系列相连的线段组成的折线，我们可以对每一个线段求轨迹点到线段上的最近点，再保留所有线段最近点上最近的一个。在求到线段上最近点时，主要思想是利用数学上的公式进行求解。

我们设线段两端为（$x_1$,$y_1$）,($x_2$,$y_2$)，轨迹点 p（$x_p$,$y_p$）。首先考虑特殊情况，即$x_1=x_2$或$y_1=y_2$时，线段为平行于坐标轴的直线，我们过 p 点做线段所在直线的垂线即可。$x_1=x_2$时，最近点 q($x_1$,max($y_1$, $y_2$))（$y_p>max(y_1,y_2)$）或($x_1$,min($y_1$，$y_2$))（$y_p<min(y_1,y_2)$）或（$x_1$，$y_p$）（$min(y_1,y_2)<y_p<$

$max(y_1, y_2)$）;$y_1 = y_2$时同理。

当$x_1 \neq x_2$且$y_1 \neq y_2$时，我们设线段所在直线的方程为$y = kx + d$，其中$k = \frac{(y_1 - y_2)}{(x_1 - x_2)}, d = y_1 - kx_1$，所以通过 p 点到$y = kx + d$的直线方程为$y = -\frac{1}{k}x + \frac{m}{k} + n$,将两式联立

$$\begin{cases} y = kx + d \\ y = -\dfrac{1}{k}x + \dfrac{m}{k} + n \end{cases}$$

解得$x_q = \frac{\frac{m}{k} + n - b}{k + \frac{1}{k}}$，我们比较该点是否在线段上，如果在就取该点值，如果不在取线段离$x_q$最近的端点值，$x_q = max(x_1, x_2)(x_q > max(x_1, x_2))$或$x_q = min(x_1, x_2)(x_q < min(x_1, x_2))$或$x_q = x_q(min(x_1, x_2) < x_q < max(x_1, x_2))$, $y_q = kx + d$。

如此操作后我们得到了 p 到线段上最近的点 q，接下来我们根据 p,q 之间的距离选择所有线段上距离 p 最近的点 q 即可。

**2. 计算路网上两点间的距离**

设$x_{t,i}$表示轨迹点$z_t$在路$r_i$上的最近点，要计算的是点$x_{t,i}$到点$x_{t+1,j}$之间的距离 dis[i][j]，我们利用 BFS（广度优先搜索）的方法搜索在起始线段$r_i$周围的路段。

我们利用 trans[]队列来储存 BFS 的遍历顺序，初始时将路段$r_i$的终点与$x_{t,i}$到该终点的路径长度作为结构体加入 trans 中，每次 trans 头部元素 intersection（路口）出列，考虑以 intersection 作为起始点的所有路段，将这些路段的终点以及到达时的路径长度作为一个结构体都压入 trans 队列中。如果遍历深度大于3 时，我们通常认为不会在短时间内经过 3 个及以上的路口，因此可以不必考虑这些情况，在算法中就是遍历深度大于 3 的点不必考虑。在考虑以 intersection 作为起始点的路段时，我们同时遍历所有$z_{t+1}$可能匹配的路段$r_j$,如果两者相同，我们就计算此时点$x_{t,i}$到点$x_{t+1,j}$的距离$dis = intersection_len +$

$distance(r_{j.start}, x_{t+1,j})$,若 dis[i][j]之前不存在或$dis[i][j] < dis$,则更新$dis[i][j] = dis$。

这里要注意一个细节，即计算$x_{t,i}$与$x_{t+1,i}$的距离时，由于都是在同一条路段$r_i$上，我们计算距离时应该用$dis[i][j] = distance(r_{i.end}, x_{t,i}) - distance(r_{i.end}, x_{t+1,i})$计算，而不是在 BFS 算法中的先从$r_{i.end}$出去，再从$r_{i.start}$返回的回路距离。但如果$distance(r_{i.end}, x_{t,i}) - distance(r_{i.end}, x_{t+1,i}) < 0$时，因为道路是单行道，不能掉头，所以我们可以假设是通过其他路段绕了一圈回来，此时依旧适用于 BFS 算法。

以上的算法可以计算出从点$x_{t,i}$出发的所有路径长度 dis[i][j]，对于不同的 i，我们利用一个 for 循环全部重复一次上述的计算便可以得到$dis[i][j], \forall i, j$。上述的方法其实和 bellman-ford 类似，每次进行松弛。

**3. 计算测量概率（measured possibility）**

我们假设测量概率和$z_t$到$x_{t,i}$的距离$\|z_t - x_{t,i}\|$之间成正态分布关系，由正态分布的概率公式建立模型，$z_t$到$x_{t,i}$的测量概率$p(z_t|r_i) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-0.5\left(\frac{\|z_t - x_{t,i}\|}{\sigma_z}\right)^2}$,其中$\sigma_z = 1.4826 median_t(\|z_t - x_{t,i}\|)$。

**4. 计算转移概率（transition possibility）**

基于论文的阐释，我们可以发现在大量真实数据的测量下，$\|z_t - z_{\{t+1\}}\|$与$\|x_{t,i} - x_{t+1,j}\|_{route}$间有着数据关系，其分布与$\left|\|z_t - z_{\{t+1\}}\| - \|x_{t,i} - x_{t+1,j}\|_{route}\right|$的值成指数函数关系，具体如下图 1 所示，基于这个性质我们可以构造出关于路径上距离差的转移概率：

$$p(d_t) = \frac{1}{\beta} e^{-d_t/\beta}$$

其中$d_t = \left|\|z_t - z_{t+1}\| - \|x_{t,i} - x_{t+1,j}\|_{route}\right|$,

$\beta = \frac{1}{ln2} mediant_t(\left| \|z_t - z_{t+1}\| - \|x_{t,i} - x_{t+1,j}\|_{route} \right|)$

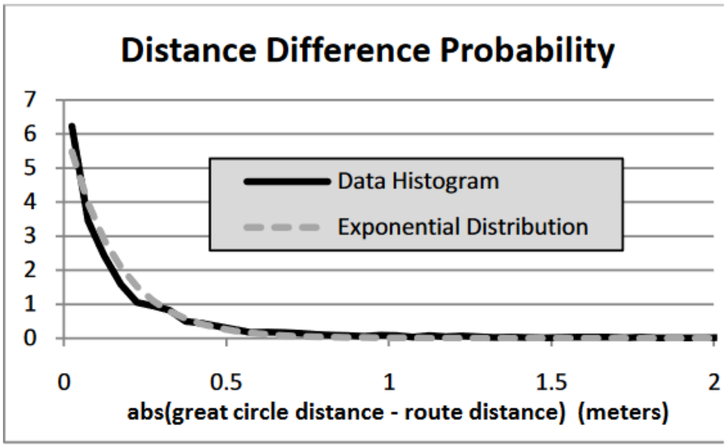


图 1

5.**Viterbi 算法的动态转移**

Viterbi 算法使用动态规划来快速找到 HMM 模型网格中的路径，使这条路径上测量概率和转换概率的乘积最大化。设乘积$P_{t+1}[k]$是轨迹点$z_{t+1}$在路段$r_k$上的投影点对应乘积之和，我们在进行动态转移时只需要考虑前面一个状态的乘积，转移方程为：

$$P_{t+1}[k] = max_{\forall i}(P_t[i] * p(d_t)) * p(z_t|r_i)$$

在转移后要注意一点，即所有的概率都是小于 1 的数字，如果过多的概率相乘，最后得到的概率积是一个很小的数，可能不能够在计算机中存储，于是我们可以在每一层 t 的转换后给所有$P_{t+1}[k]$乘上一个放大因子，使数据保持在合理的范围内。这里我选择每次乘以$\frac{1}{max_{\forall k(P_{t+1}[k])}}$,这样最大的$P_{t+1}[k]$就被还原为 1，所有步骤中的概率也在 1 附近的小数部分，便于计算。

## 三、创新点

### 记忆化存储

利用 map<pair<int,int>,double>实现记忆化存储路口间距离,当第一次计算后，后续使用都不必重新计算。通过实际检验，我们发现这样计算后的时间复杂度大大降低。同时为了更好使用这一方法，当需要两个路口间

距离时，我们选取局部做 Dijkstra 算法求两个路口之间的距离，随后把产生的所有结果都给压入 map 中。

## 四、实验结果与分析

```
1000
99226 99226 99226 99227 99227 99228 99228 25 61020 61021 1439 1198 61015 61015 61015 61016 1191 1192 1192 1192 1192 1192 1192 1192 1192 1192
61451 61451 61451 61451 61453 61453 61453 61453 61455 61455 61455 61455 61457 61457 61457 61459 61459 61459 61459 61461 61461 61461 61461
11989 11988 11988 11988 120105 120105 120106 120106 120106 120107 120107 120107 12101 12101 12101 12101 12101 12101 12099 12097 12097 12097
45014 43612 43612 43612 43612 43612 43612 43612 8247 8247 43614 43614 43615 43616 43618 43618 43618 43619 43619 51279 51279 51279 51279
63460 63458 63458 4363 4363 4363 4363 4361 4361 4361 4361 4361 4357 4355 4355 4355 4355 4355 4355 4355 4355 4353 4353 4353 4353 4353 4353
2298 2299 2299 2299 2299 2299 2297 10245 2449 43629 531 2449 1361 1361 43629 1361 1361 1361 1361 1361 1361 54 54 54 55 17539 17539 43612
5071 5071 5071 5072 5072 12997 12997 12998 12998 58217 58217 58217 58217 58217 58218 58218 40735 40735 40735 11202 40738 40739 40739 40734
38081 38080 38080 38083 38083 38083 38093 38085 38085 38087 38210 38210 26535 26535 26535 26535 26535 26535 38197
10202 10202 1185 1186 1187 1187 1187 1187 1187 1187 1187 1187 1187 1187 1187 1189 1189 1190 1190 1190 1190 1195 1197 1197 1197 1197 1200
5125 6028 6028 9695 5125 6029 6029 6029 5125 5129 5129 5130 19006 19006 19006 19001 5082 9695 9695 9695 9695 9695 9695 9695 9695 9695 9695
5254 5243 5243 5243 5243 5243 5243 5243 5243 5228 5228 5228 5228 5228 5228 5228 50554 5229 5229 5229 5229 5229 5230 5229 50553 50553 50553
1961 1961 873 706 706 706 706 61306 2079 894 2079 1962 1962 1962 2501 2501 2501 2502 11667 11667 11667 2293 11651 11668 11668 11668 11668
2100 2100 2101 2101 2101 2101 2101 2101 2101 1337 1338 1338
99175 99176 99174 1928 1939 1939 1940 1940 1940 6700 6700 6700 6700 1942 6723 6723 6723 6723 6723 6723 6725 10330 10330 10331 10332 10329
512 512 512 513 513 8224 8224 8226 17549 17542
35389 35389 35407 35407 6032 3279 3279 3279 3279 3279 35408 6033 6033 35408 35386 35409 35383 35383 35383 35383 35383 35383 35383 35383
3316 3276 3276 3276 3276 3276 3276 3277 3277 3277 3277 3277 3279 3279 35387 35385
```

```
1000
99226 99226 99226 99227 99227 99228 99228 25 61020 61022 61022 12527 61015 61015 61015 61017 1191 1192 1192 1192 1192 1192 1192 1192 1192
61451 61451 61451 61451 61453 61453 61453 61453 61455 61455 61455 61455 61457 61457 61457 61459 61459 61459 61459 61461 61461 61461 61462
11990 11988 11988 11988 120105 120105 120106 120106 120106 120107 120107 120107 12101 12101 12101 12101 12101 12101 12099 12097 12097 12097
45015 8247 8224 8224 8224 8224 8224 8224 8224 8224 43613 43614 43615 43617 43618 43618 43618 43619 43619 51279 51279 51279 51279 51279
63460 63458 63458 4363 4363 4363 4363 4361 4361 4361 4361 4361 4357 4355 4355 4355 4355 4355 4355 4355 4355 4353 4353 4353 4353 4353 4353
2299 2299 2299 2299 2297 2297 2297 10245 1360 1360 1360 1361 1361 1361 1361 1361 1361 1361 1361 1361 1361 54 54 54 8254 45014 45014 45014
5071 5071 5071 5072 5072 12997 12997 12998 12998 58217 58217 58217 58217 58217 13372 40735 40735 40735 40735 5029 40738 40739 40739 40734
38082 38080 38080 38083 38083 38083 38093 38085 38085 38085 26535 26535 26535 26535 26535 26535 26535 26535 38197
10202 24 1185 1186 1187 1187 1187 1187 1187 1187 1187 1187 1187 1187 1187 1189 1189 1190 1190 1190 1190 1195 1197 1197 1197 1197 1200 12519
5125 6029 6029 6029 6029 6029 6029 5125 5125 5125 19006 19006 19006 19006 19001 19001 5079 5080 5081 5081 5081 5081 5081 5081 5081 5081
5242 5243 5243 5243 5243 5243 5243 5243 5228 5228 5228 5228 5228 5228 5228 5228 5228 5228 5228 5228 5228 5228 5228 5230 50553 50553 50553
1961 1961 873 706 706 706 706 706 706 706 706 1962 1962 1962 2501 2501 2501 11667 11667 11667 11667 11667 11667 11667 11668 11668 11668
1369 2100 2101 2101 2101 2101 2101 2101 2101 1337 1338 1338
99176 99176 99174 1939 1939 1939 1940 1940 1940 1940 1940 1940 1940 1941 6723 6723 6723 6723 6723 6723 6725 10330 10330 10332 10332 1779
512 512 512 8224 8224 8224 8224 8226 17549 17542
35389 35389 35407 35407 35407 35407 35407 35407 35407 35407 35407 35407 35407 35407 35409 35409 35409 35410 35410 35410 35410 35410 35410
3314 3276 3276 3276 3276 3276 3276 3277 3277 3277 3277 3277 3279 3279 35387 35385
```

```
sum=57433 correct=54530
0.949454
PS E:\VS project\pj>
```

程序最后的准确度接近于 95%，通过打印出每一时刻的转移概率表格与测量概率表格，发现在发生分歧的点上，几乎都是由于产生了另一条道路，其与给出的标准解答离两个测量点间的距离更近，于是这一条道路就取代了原本的道路，成为了新的分叉点。

但如果使用打印当前最佳点的方法，我们又可以在一定程度上弥补这些遗漏的点，可以在将来试图寻找一种权衡方法使得能够同时考虑到乘积最大的路径和当前乘积最大的点，得到的算法能够在最后有更高的准确度。