4.2轨迹的距离/近似度

在回答KNN查询或聚类轨迹时，我们需要计算一个轨迹和几个点之间的距离（或者我们可以说近似度），或者两个轨迹之间的距离。

点和轨迹之间的距离通常用从点到轨迹上的最近点的距离来度量，记为

例如： 和 如图所示。

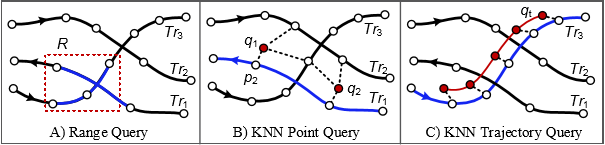


图9

从单点 延伸到多个查询点 的时候，公式变为

在这里采用了与上式相同的构成方式，如图 所示

在这里我们使用指数函数的目的，是给那些更接近待匹配轨迹的点，分配一个更大的贡献值（contribution），同时降低那些相对遥远的点的价值。

陈等人定义了最佳连接距离（Best Connect Distance），它可以测量轨迹与若干点之间的距离，无论这些点是否有顺序。

两个轨迹之间的距离通常通过轨迹点之间的某种距离的聚合来测量。

最近对距离（Closest-Pair Distance, CPD）使用两个轨迹（𝐴，𝐵）之间的最小距离点来表示轨迹的相似性，即

假设两个轨迹具有相同的长度，对数和距离(Sum-of-Pairs Distance, SPD)使用来自两个轨迹的对应点的和来表示距离，即.

由于这个假设在现实中可能不成立，为了得到最好的对齐，我们提出了动态时间调整（Dynamic Time Wrapping, DTW）距离以允许多次重复某些必要的点数。

由于轨迹中的某些噪声点可能会在轨迹之间产生很大的距离，所以使用最长公共子序列（Longest Common Sub-Sequence, LCSS）的概念来解决这个问题。基于LCSS的距离允许在计算轨迹距离时跳过一些噪声点，使用阈值δ来控制我们可以走多远的时间，以便匹配从轨迹到另一轨迹中的点的一个点。另一个阈值ε用于确定两个点（来自两个不同的轨迹）是否匹配。陈等人。 [18]提出的EDR距离，这是类似于LCSS使用阈值ε来确定一个匹配，而罚款分配两个匹配的子轨迹之间的差距。在[19]中，陈等人。也提出了结合DTW和EDR的优点的ERP距离，通过使用一个恒定的参考点来计算距离。请注意，DTW，LCSS和EDR都不是度量距离函数，因为它们不满足三角不等式。 ERP是度量的，因此可以用来修剪不必要的轨迹[18]。

基本上，LCSS和编辑距离提出了匹配的字符串。当用于匹配两个轨迹时，需要设置一个阈值ε;这并不容易。 K-BCT [21]是一个轨迹的无参数相似性度量，结合了DTW和LCSS的优点。在匹配过程中，K-BCT可以重复一些轨迹点并跳过包括异常值在内的不匹配的轨迹点。

两个轨迹段之间的距离：轨迹段的距离度量基于段的最小边界矩形（MBR）[43]。如图12（a）所示，两个分段（𝐿1，𝐿2）的MBR分别为（𝐵1，𝐵2），每个分段由低界点（𝑥𝑙，𝑦𝑙）和上界点（</s>， 𝑦𝑢）。基于MBR的距离𝐷𝑚𝑖𝑛（𝐵1，𝐵2）定义为从（𝐵1，𝐵2）开始的任意两点之间的最小距离，计算公式为：

√（Δ（[XL，许]，[x'l，x'u]））2+（Δ（[基，宇]，[Y'L，y'u]））2;

两个区间之间的距离定义为：

Δ（[x1，xu]，[x'l，x'u]）= {0 x'l-xu xl-x'u [xl，xu]∩[x'l，x'u]≠x' l> xu xl> x'u;

在图12A所示的两个例子中，𝐿1和𝐿2之间的距离分别为0和𝑦'𝑙𝑙𝑦</s>。

如图12B）所示，Lee等人[51]提出了一个距离函数，名为轨迹 - 豪斯多夫距离（𝐷𝐻𝑎𝑢𝑠），它是三个项的加权和：1）测量两个轨迹之间间隔的聚合垂直距离（𝑑⊥）; 2）捕获两个轨迹之间长度差异的聚合平行距离（𝑑//）; 3）反映两个轨迹之间的方位差的角距（𝑑θ）。从形式上看，

𝐷𝐻𝑎𝑢𝑠=𝑤1𝑑⊥+𝑤2𝑑// +𝑤3𝑑θ，

其中𝑑⊥=𝑑⊥，𝑎2+𝑑⊥，𝑏2𝑑⊥，𝑎+𝑑⊥，𝑏，𝑑// = min（𝑑//，𝑎，𝑑//，𝑏），𝑑θ= ||𝐿2||∙∙ 𝑠𝑖𝑛θ，𝑤1，𝑤2，</s>3

重量取决于应用程序。

图12.轨迹段的距离度量