Snap2Route Breadcrumb Deviation Algorithm

For CS510 Data Explorations, Summer 2020

July 10, 2020

Deviation Algorithm Given an input set of timed breadcrumb records B along single shape and a single set of coordinates S defining that shape, what is the projection of B onto S?

Note: The set B may contain breadcrumbs from multiple trips.

Principle 1 The projections onto S should be in chronological order.

Principle 2 If some point p has a large error but its neighbors do not, then p should be assigned a large deviation value and its neighbors should not be assigned large values. Especially, if p is out of order but its neighbors are not, then p should be the only point assigned a large deviation value.

```
Compute-Deviations($S, B$)
Sort $S$ by pointOrder
for every record $r_i \in S$
    if $r_i.pointOrder \neq 0$
        Set $r_i.distTraveled = dist(r_i, r_{i-1})$
    else
        Set $r_i.distTraveled = 0$
for every record $r \in B$
    $r.projection$ = nearest point on all segments in $S$
    $r.distTraveled = r.projection$'s distance along $S$
Sort $B$ by tripID, time // Note we don't compare crumb records to each other until this point
for Each unique tripID $t \in B$
    for Each record $r_i \in B$ such that $r.tripID = t$
        if $r_i.distTraveled < r_{i-1}.distTraveled$
            Look backward to find correct projection halfway between new neighbors
        elseif $r_i.distTraveled > r_{i+1}.distTraveled$
            Look forward to find correct projection halfway between new neighbors
return $B$
```

Run Time In Python, time complexity is hard to determine due to the obfuscated nature of the abstract data structures. At this point, it appears that efficient operations do not loop through large record sets within the python code; instead, such operations are delegated to libraries. Thus, I anticipate that this or any algorithm will be implemented with function calls that do not syntactically resemble the algorithm's psuedocode.