Similarity between line strings

Asked 16 years, 3 months ago Modified 8 years, 1 month ago

Viewed 3k times

13

I have a number of tracks recorded by a GPS, which more formally can be described as a number of line strings.

Now, some of the recorded tracks might be recordings of the same route, but because of inaccurasies in the GPS system, the fact that the recordings were made on separate occasions and that they might have been recorded travelling at different speeds, they won't match up perfectly, but still look close enough when viewed on a map by a human to determine that it's actually the same route that has been recorded.

I want to find an algorithm that calculates the similarity between two line strings. I have come up with some home grown methods to do this, but would like to know if this is a problem that's already has good algorithms to solve it.

How would you calculate the similarity, given that similar means represents the same path on a map?

Edit: For those unsure of what I'm talking about, please look at this link for a definition of what a line string is:

http://msdn.microsoft.com/en-us/library/bb895372.aspx - I'm *not* asking about character strings.

sql-server algorithm gis

Share

Improve this question

Follow

edited Jul 10, 2014 at 18:09

Alessandro Jacopson
18.6k ● 16 ● 104 ● 156

asked Sep 15, 2008 at 12:49

Liedman
10.3k ● 4 ● 35 ● 36

6 Answers

Sorted by: Highest score (default)

12

Compute the Fréchet distance on each pair of tracks. The distance can be used to gauge the similarity of your tracks.

Math alert: Fréchet was a pioneer in the field of metric space which is relevant to your problem.

Share Improve this answer

Follow

edited Nov 19, 2016 at 16:27

answered Sep 17, 2008 at 15:08

erickson
269k ● 59 ● 401 ● 497

2 As a mathematician, +1 just for citing Fréchet! – Edwin Jarvis Feb 11, 2009 at 4:08

3

I would add a buffer around the first line based on the estimated probable error, and then determine if the second line fits entirely within the buffer.

Share Improve this answer Follow

answered Sep 15, 2008 at 12:53

Paul Tomblin
183k ● 59 ● 323 ● 410

2

To determine "same route," create the minimal set of normalized path vectors, calculate the total power differences and compare the total to a quality measure.

1. Normalize the GPS waypoints on total path length,
2. walk the vectors of the paths together, creating a new set of path vectors for each path based upon the shortest vector at each waypoint,
3. calculate the total power differences between endpoints of each vector in the normalized paths weighting for vector length, and
4. compare against a quality measure.

Tune the power of the differences (start with, say, squared differences) and the quality measure (say as a percent of the total power differences) visually. This algorithm produces a continuous quality measure of the path match as well as a binary result (Are the paths the same?)

> Paul Tomblin said: I would add a buffer around the first line based on the estimated probable error, and then determine if the second line fits entirely within the buffer.

You could modify the algorithm as the normalized vector endpoints are compared. You could determine if any endpoint difference was above a certain size (implementing Paul's buffer idea) or perhaps, if the endpoints were outside the "buffer," use that fact to ignore that endpoint difference, allowing a comparison *ignoring side trips*.

Share Improve this answer Follow

edited Sep 15, 2008 at 14:05

answered Sep 15, 2008 at 13:13

Stephen Marney
23 ● 4

1

You could walk along each point (Pa) of LineString A and measure the distance from Pa to the nearest line-segment of LineString B, averaging each of these distances.

This is not a quick or perfect method, but should be able to give use a useful number and is pretty quick to implement.

Do the line strings start and finish at similar points, or are they of very different extents?

Share Improve this answer Follow

edited Sep 16, 2008 at 5:26

answered Sep 15, 2008 at 23:03

P Kinney
33 ● 5

1

If you consider a single line string to be a sequence of [x,y] points (or [x,y,z] points), then you could compute the similarity between each pair of line strings using the Needleman-Wunsch algorithm. As described in the referenced Wikipedia article, the Needleman-Wunsch algorithm requires a "similarity matrix" which defines the distance between a pair of points. However, it would be easy to use a function instead of a matrix. In your case you could simply use the 2D Euclidean distance function (or a 3D Euclidean function if your points have elevation) to provide the distance between each pair of points.

Share Improve this answer Follow

answered Sep 18, 2008 at 4:01

Chuck Wooters
1,494 ● 1 ● 9 ● 7

-2

I actually side with the person (Aaron F) who said that you might be interested in the Levenshtein distance problem (and cited this). His answer seems to me to be the best so far.

More specifically, Levenshtein distance (also called edit distance), does not measure strictly the character-by-character distance, but also allows you to perform insertions and deletions. The best algorithm for this distance measure can be computed in quadratic time (pretty slow if your strings are long), but the computational biologists have pretty good heuristics for this, that might be of interest to you on their own. Check out BLAST and FASTA.

In your problem, it seems that you are dealing with differences between strings of numbers, and you care about the numbers. If you give more information, I might be able to direct you to the right variant of BLAST/FASTA/etc for your purposes. In any case, you might consider adapting BLAST and FASTA for your needs. They're quite simple.

1: http://en.wikipedia.org/wiki/Levenshtein_distance, http://www.nist.gov/dads/HTML/Levenshtein.html

Share Improve this answer Follow

answered Sep 15, 2008 at 18:02

eladv

2,001 ● 1 ● 12 ● 3

I have a hard time getting my head around how to transform my problem, that to me seems to be in the domain of computational geometry, to something dealing with character strings (whether it's DNA sequences or character strings). A line string is a list of coordinates, connected by lines.
– Liedman Sep 15, 2008 at 19:29

Ah, I see. My mistake. I thought that by "line string" you meant a string of characters. I was wondering about how you transform one to the other as well. Aaron F probably had the same misunderstanding. So, disregard my answer. – eladv Sep 16, 2008 at 11:22

(I'll keep my answer posted rather than delete it, because maybe it is somewhat relevant. I'll think more about that).
– eladv Sep 16, 2008 at 11:24