Assignment 3: Trajectory Compression

2022 Fall EECS205002 Linear Algebra

Due: 2022/12/14

Nowadays most mobile devices have GPS for driving navigation or trajectory recording. A trajectory is a list of tuple (x_i, y_i, t_i), where x_i is the longitude, y_i is the latitude, and t_i is the time. Because the data size can grow fast when the recording time is long, people usually compress the GPS data. One of commonly used methods is Ramer–Douglas–Peucker algorithm. But in this project, we will use the least square method to compress the data.

Let $(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)$ be a trajectory to be compressed. We are going to find a line segment in the 3D space, spanned by x, y, and t, where x and y are linear functions of t, $(x, y) = (at + b, ct + d)$. We need to find variables a, b, c, d such that

$$\sum_{i=1}^{n} (at_i + b - x_i)^2 + (ct_i + d - y_i)^2$$

is minimized. Figure 1 shows an example, in which $n = 4$, and the goal is to minimize the sum of the squared distances of the red line segment.

Let's express the above summaztion using a matrix form. Let

$$A = \begin{bmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_n & 1 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \text{ and } w = \begin{bmatrix} a \\ b \end{bmatrix}, z = \begin{bmatrix} c \\ d \end{bmatrix}.$$

The above problem becomes

$$\min_{w,z} \|Aw - X\|^2 + \|Az - Y\|^2.$$

Since w and z are not separated in two terms, we can split the above equation into the sum of two least square problems,

$$\min_{w} \|Aw - X\|^2 + \min_{z} \|Az - Y\|^2.$$

However, a trajectory may not be a straight line. We can approximate it using several line segments. Figure 2 shows an example, in which there are 253 GPS points. The left subplot is the original trajectory, and the right subfigure

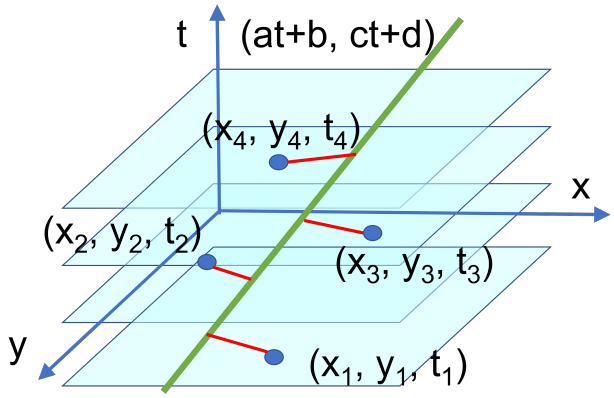

Figure 1: An example of compressing a trajectory using least square method.

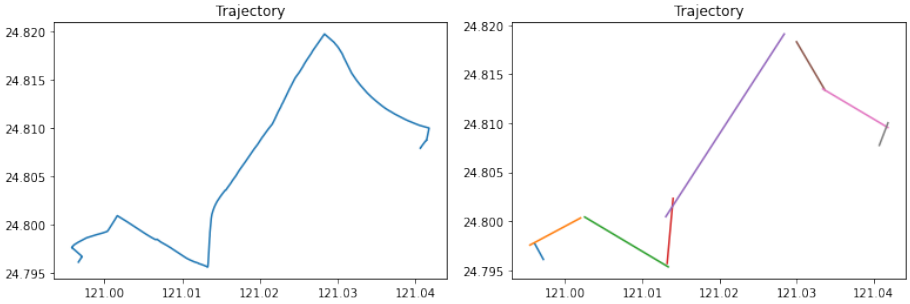

Figure 2: An example of compressing a trajectory using least square method.

is the approximated trajectory using several line segments. Each line segment is computed using the least square method. The break points of line segments are selected by hands. Since there are 8 segments, which equals to 16 points, the compression ratio is 16/253=0.063.

1 Assignments

1. (20%) Using Google Map (`https://www.google.com.tw/maps`) to find one of your favorite route, and convert it to the GPX file `https://www.groovypost.com/howto/export-google-maps-route-data/`. You can also get the GPX file from your sport APP. Your GPX should contain at least 100 points, but less than 1000 points. The trajectory should have at least 3 turns. Try to break the trajectory into line segments, and use the given program to run the results.

2. (10%) Study the linear least square function `numpy.linalg.lstsq` and answer the following question.

   - How to know the errors $\min_w \|Aw - X\|^2 + \min_z \|Az - Y\|^2$?
   - What is the meaning of `rcond`? Try different values and see its effect. Explain your result.

3. (30%) The given program has some drawbacks. First, you need to select the break points. Second, the line segments are not connected. Third, the errors of the line segments are not controllable. Design and implement an algorithm that uses linear least square method to compress the GPS data. Figure 3 gives an example that contains four subfigures. Each shows the break points for specific errors. Your algorithm/code should be verified on the given GPX file and your own GPX file in question 1. The quality of your algorithm/code will be judged by the following criteria.

   - Correctness (20%): Your algorithm should fit all the requirements: (1) it can automatically break a trajectory into line segments, (2) all the line segments are connected, and (3) the error of each line segment is no more than the given error bound ϵ, which means

     $$\sum_{i=1}^{n} (at_i + b - x_i)^2 + (ct_i + d - y_i)^2 \leq \epsilon$$

     for each line segment. In addition, you should explain the correctness of your algorithm.
   - Computing efficiency (5%): What is the time complexity of your algorithm? Suppose a function call to lstsq takes $F(k)$ time, where k is the number of input points, and there are m line segments, and the total number of input point for a trajectory is n.
   - Compression efficiency (5%): How many break points are required for a given trajectory under a specific error bound ϵ? For example, in my implementation, for $\epsilon = 10^{-4}$, it only needs 9 points. So the compression ratio is $9/253 = 3.6\%$.

4. (20%) Derive the formula of using the quadratic curves to fit the trajectory. The form of the quadratic curve is

   $$x = a_1 t^2 + a_2 t + a_3,$$
   $$y = b_1 t^2 + b_2 t + b_3.$$

   Explain your derivation. Implement your algorithm and evaluate it using the given GPX file and your trajectory. You can use the given function `drawCurve` to show the curves.

5. (20%) Look up the *total least squares* method. Explain its algorithm, and the difference with the linear least square method, and use it to compress the trajectory data. Compare the results with that of least square method.

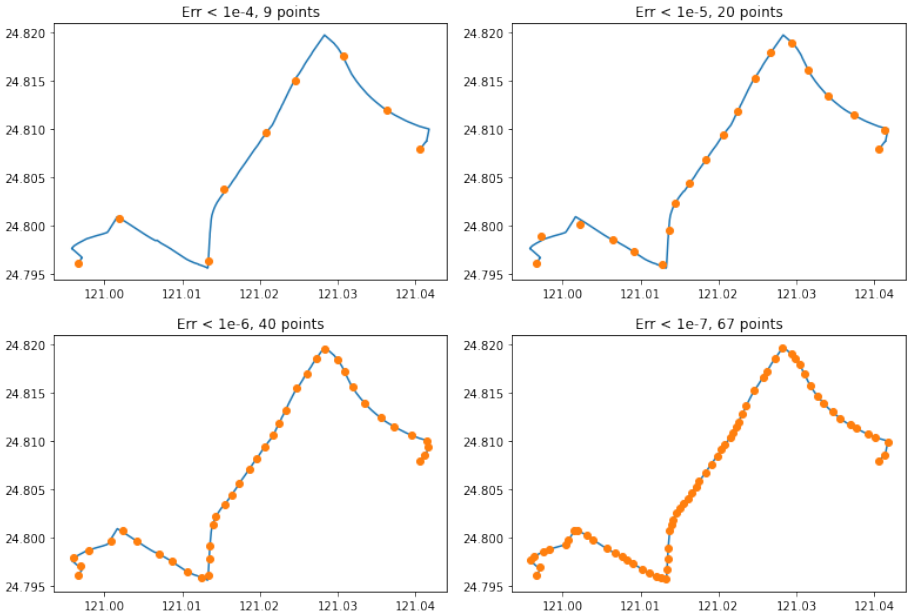

Figure 3: An example of compressing a trajectory using least square method.

2 Submission

1. Write a report in PDF file that includes the answers of question (1), (2), (3), (4), and (5).
2. The code of (1), (3), (4), (5).
3. Zip them and submit the zip file to the eeclass.