Midterm report

Title: Utilize machine learning methods to improve the movement measurement accuracy according to GPS data

1. What have been done

(1) Establish testing system.

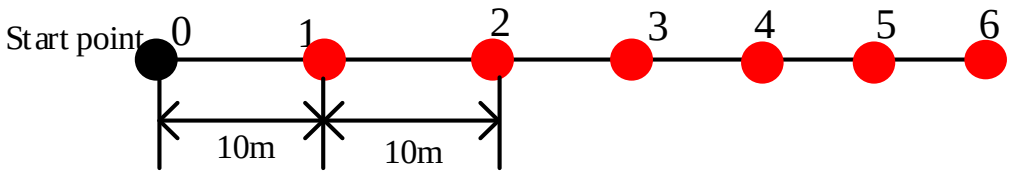


Figure 1 testing point diagram

The whole testing system is consists of 7 points in a line (1 start point and 6 test points). Each point is 10 meters away from its neighbors. The start point is a place with known GPS data (test with more expensive GPS for a very long time). The accurate position for other 6 test points is unknown. We want to measure their distances from start point using the GPS data we collected in a short time.

(2) GPS data collection.

We collect 3000 GPS data for each point, including the start point. Every data consists of one latitude and one longitude.

The excel file to store GPS data has been uploaded to github.

(3) Weight calculation

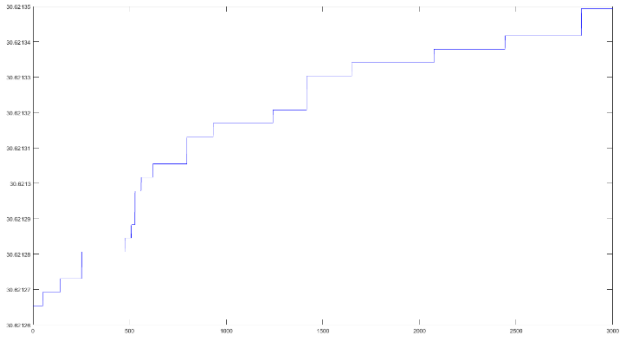


Figure 2 3000 latitudes sorted in ascending

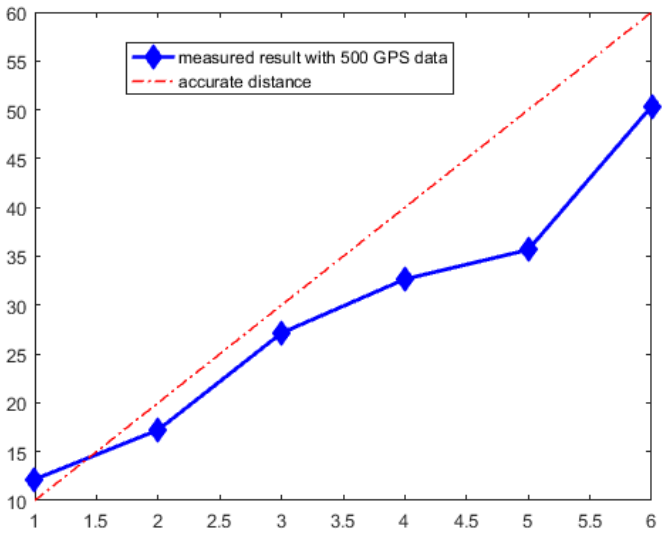
Value	Count	Percent
30.6211547851562	10	0.84%
30.6211624145507	10	0.84%
30.621166229248	23	1.92%
30.6211795806884	47	3.93%
30.6211833953857	17	1.42%
30.621187210083	26	2.17%
30.6211948394775	11	0.92%
30.6211986541748	123	10.28%
30.6212043762207	85	7.10%
30.6212120056152	68	5.68%
30.6212158203125	91	7.60%
30.6212196350097	66	5.51%
30.621223449707	39	3.26%
30.6212329864501	18	1.50%
30.6212368011474	61	5.10%
30.6212406158447	81	6.77%
30.6212482452392	152	12.70%
30.6212520599365	62	5.18%
30.6212558746337	39	3.26%
30.6212654113769	40	3.34%
30.6212692260742	38	3.17%
30.6212730407714	70	5.85%
30.621280670166	20	1.67%

Figure 3 discrete data frequency treated as weight

The GPS data for each point (latitude and longitude) are discrete in several possibilities. We calculate each possible value frequency in Matlab. These frequency are treated as weight in later algorithm.

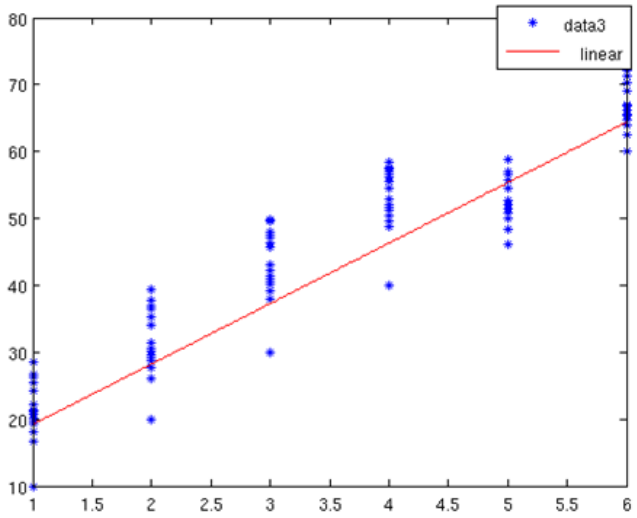
(3)Simple Linear regression implementation.

If we use the average value of random 500 GPS data to represent the location of each point. Then we use Haversine formula https://en.wikipedia.org/wiki/Haversine_formula to calculate its distance to start point. The initial result is:



Where the error rate for test point 1 to 6 is 21.6239%, 13.88%, 9.51%, 18.35%, 28.66%, 16.06%.

If we calculate the distance to start point for each GPS data then do linear regression to $\min_n (d_i - \theta x_i)^2$, where d_i is the distance and x_i is the index for test point.



We find the model:

d = 9.0417x+6.137. So the error rate for each test point is 51.8%, 21.1%, 11.2%, 6.2%, 2.69%, 0.64533%. We can see with the linear regression, the error rate is smaller than average calculation when we try to measure a long distance.

(4).weighted Quadratic regression.

The linear regression above is a predictor. I want to use quadratic regression to implement a classification to discriminate the 'bad' GPS data from relatively accurate one. We use the data for start point as training data and the GPS data for other points as testing data.

Data normalization: we normalize each GPS data (x, x^1), where x is latitude and x^1 is longitude, to $(x - \overline{x}, x^1 - \overline{x^1})$.

The weight w has been calculated using the frequency.

We classify one GPS data as 'good' data and labeled 1 when its error to accurate position is less than 2 meters. Classify GPS data as 'bad' data labeled -1 when its error to accurate position is less than 2 meters.

Our goal is to find a parameter set β that:

$$\min_n (y_i - \beta_1 {x_i}^2 - \beta_2 {x_i^1}^2 - \beta_3 x_i - \beta_4 x_i^1)^2$$

I implemented 50 times iteration and get the parameter result: -5.59682464541870e-07

-8.40214567666491e-07, 0.00326637596554780, 0.000419273918630419.

However, the training data error rate still reaches 28.16%.

This is my current bottom-neck. I am trying to fix it.

2. What we need to do for final result?

Continue to improve our regression method and try to implement other machine learning methods to compare the result. The quadratic regression does not works very well, I need to find the problem and solve them.

3. Existing challenges and the ways to solve them

The GPS data has few features and I did not find the suitable parameters for quadratic model. I will try other ways instead of gradient descent to find the parameters. Or I will try to increase the step size in gradient descent.

Another challenge is I find for the same point, longitude is relatively more stable than latitude. The longitude has less potential values. I am thinking about how to use this information to further improve the accuracy of result.

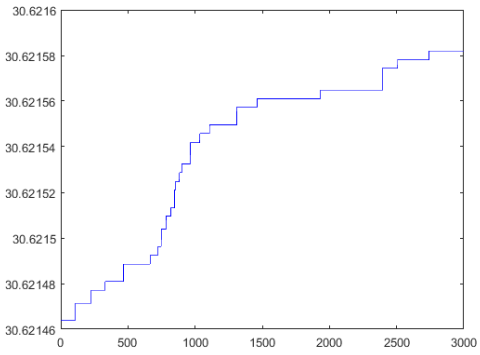


Figure 4 latitude distribution for point 4

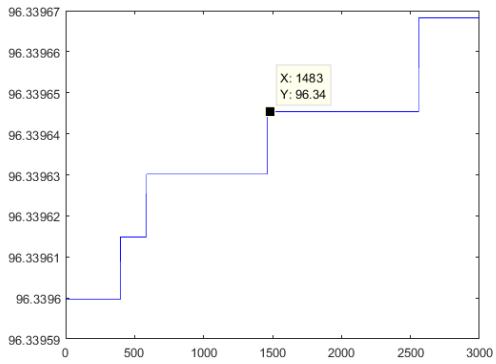


Figure 5 longitude distribution for point 4

I think about the uncertainty difference for latitude and longitude may also be considered as another weight.

4. Changes from the proposal.

I change the purpose for this project. The initial purpose is to improve the absolute location we get from GPS. However, I found it is hard to verify the accuracy of absolute position. So I change the purpose into improve the movement measurement using GPS. In proposal, I think of reinforcement learning. However, I don't come up an idea to define the cost function and actions to make the algorithm really work.