PROJECT REPORT

Methods Overview in taxi trajectory prediction

Authors:
Ettore Randazzo
Vittorio Selo
Paolo Simone
Benedetto Vitale
Professor:
PierLuca Lanzi
Course:
Data Mining And Text Mining

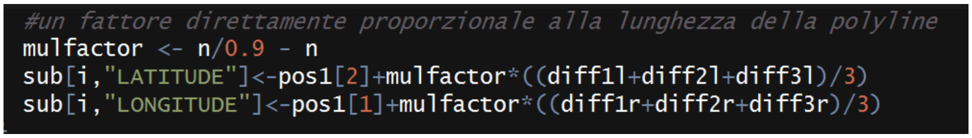
June 15, 2015

1 Introduction

In this report we are going to present an overview of the methods used in achieving the score of 2.57(30th position with 10398415 on Sunday,June 14th at 23:59:59) in the relative Kaggle competition.

2 Attempt I: Interpolation

As first attempt we have just tried a simple interpolation based on the n last positions available on the test trajectories and on the distance already covered from the starting position.

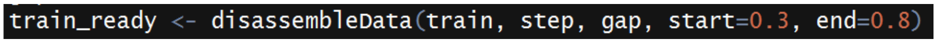


```
#un fattore direttamente proporzionale alla lunghezza della polyline
mulfactor <- n/0.9 - n
sub[i,"LATITUDE"]<-pos1[2]+mulfactor*((diff1l+diff2l+diff3l)/3)
sub[i,"LONGITUDE"]<-pos1[1]+mulfactor*((diff1r+diff2r+diff3r)/3)
```

The code snippet above shows an example of interpolation considering the last four points.

3 Attempt II: Disassemble with two digit approximation

Our second attempt, since the various polyline in the train set have different length, we have thought of a function, called Disassemble:



```
train_ready <- disassembleData(train, step, gap, start=0.3, end=0.8)
```

The aim of this function is to generate, for each train trajectory, a series of sub trajectories of a certain length(parameter ”step”); after the first sub trajectory is considered, it then passes to the next one by going ahead of a certain number of points(parameter ”gap”) and so on until a sub trajectory of length step is still available. The parameters ”start” and ”end” represents together the percentage of the points of the polyline which are considered(each point of the output has been approximated to two decimal-digits). Then we have performed the prediction by disassembling each test trip always with the same function and then by using mainly four classifiers(once at a time):

- Decision Trees
- Regression trees(with the numeric version of the disassemble)
- Naive Bayes
- KNN(K Nearest Neighbours)

The final prediction for each test trip is then obtained by a weighted average(weighting more the predictions related to the final sub trajectories obtained by disassembling the test trip). However, the best result obtained with this method has been 3.03 with 10k data, because increasing the number of considered data worsened performance.

4 Attempt III: Snake trajectories with random forest

In the third attempt we have improved the Disassemble function by adding three new preprocessing features:

- **Clean Proximity:** points too close to each other are merged in a unique one.
- **Clean Error:**points too far from the precedent one are eliminated.
- **Snake 250m:** All the points are clustered in the square of edge 250m, which center is the nearest one to the considered point.

This approach greatly decrease the length of trajectories but standardize them. Then the approach is just as in the previous attempt, with the unique difference that we have used Random Forest as classifier. This time, by increasing the number of data we have obtained better results, but it was too computationally and memory expensive(something like teras of RAM). With the adoption of these new preprocessing techniques, we cover just the 75% of the test data. With these approach and just 25k data, also by setting for the remaining 25% their last position as destination, we have obtained 2.95.

5 The Chosen One

In order to avoid RAM limitations, we chose to perform bagging with a certain number of chunks(of dimension 15k); for each chunk we did a Random Forest and then, once collected every chunk prediction, for each line of the test, we picked the one with the least average distance from all the others. Since this covers just the 75% of the data, for the remaining 25% we tried with:

- **Last Position:**with an average error of 3.60 Km
- **Centroid of the closest cluster:**with an average 3.45 Km
- **Interpolation 2.0:**with an average error 3.0 Km, this time we have still considered the last four points(if available)but the multiplicative factor was inversely proportional to the covered distance until the last available point and estimating the average distance covered by a trip, also putting an minimum and a maximum threshold.

5.1 Extra: Knowledge Domain

As a post processing, all the trips longer than 20 Km are likely to return to their initial position.

Fun Fact: this is an illegal behaviour of many taxi drivers that make people pay for the whole roundtrip, if they want to go outside Porto. So, many of them, for very long trips, to avoid being suspected of tax evasion, keep their GPS on until they get back to Porto.