# Master Thesis

# Applying Machine Learning For Generating Synthetic Geospatial Trajectories


## BY
## Yannick Patschke

*Prof. Benoît Garbinato*
*PhD Vaibhav Kulkarni*



**Abstract**

Applying machine learning for generating synthetic geospatial trajectories. Learning mobility behaviors of entities by applying recurrent neural networks (RNN) utilizing the learn models to generate synthetic traffic.

# Contents

# 1 Introduction

From the begins of the humanity, humans have ever needed and try to adapt of his environment and new environment. We have also rapidly understanding that we were lazy. So since that we have all our best to find way to reduce our effort and maximize our results. Firstly the first men have try to use tool to simply their life and survive; we can think of the use of mace, knife, spear, torch and so one. But rapidley not only just for protection, but also to cook, light up or other. Then when the survey among other spices was not more our main problem, we have beginning the thinking larger, and try really to increase our results in every part of our life; by beginning to construct first village and city, to regroup the production, to use tool to plow our fields. Then after that we even replace men by machine when it was possible. We have try to automatize the most possible task of our society. Finally with the arrival of computers and the internet, we have increase even more this automation of thing. We have begun not only replace but really trust the computer and its power, to do stuff that we will not able to archive or only very slowly. So we are at a point where the machine can begin to not only just do what we have explicitly says to do, but begin to do stuff or learn without being explicitly programmed. And that it is Machine Learning.

## 1.1 Machine Learning

Machine learning: a big term, but what is it exactely ? What can we archive with it ? Where it come from ? Why so widespread ? Machine learning is basically, to teach to the machine to act in situations that differ from what we show her or train her. It's a process where we train the machine with examples and then give it new but similar data and let the machine try to work with this new data based on what it has learn previous fromtraining example. Without going into details, now days we distinguish two major types of machine learning; suppervised and unsupervised.

Suppervised machine learning allow us to train the machine based on example data and their results. So basicallythe during supervised machine learning, we show to the machine our example and their correct results a number of times. Until the machine have learn some pattern to understand that if we give her a input A the correct ansver will B. For instance, we can make a connection with how a human kid learn to read. The kid will try to read some simple books a number of time, until he correctly reads and articulates all words of these books. So when he have correctly read all these books entierely, we can assum that he understand how read different words, and we can assume that if we give him another book, he will managed to read this new book even if it's the first time he read this new one. Fot the machine, it's similar at each time the machine see the data example, she will try too remember what she do good or bad and try to make better the next time. So we can that supervised machine learning, it manly use to understand the way to go from a point A (our input data) and a point B (our output data, the correct answer).

Unsuppervised machine learning works differently, this time the machine will just train on example without feedback of what is true or false, so without correct answer of the example data. So we can imagine, a child again who is given a pile of marble to tidy up separately, but no other instruction about criterias for separte them. Maybe the child will sperate them by color, size, weights or otheres, maybe mixed criteria. But he will try to found a way, so do the machine. The machine will for examples try to identify some attributs about the data that are similar between some examples and try to found a way to group them together. So here it's an example about clustering and we can see that unsupervised mahcine learning is more to found some structure in the data given.

So unspervised and supervised machine leanring, but where and when we can use it ? Nearly everywhere

and any times, if we are attentive. It's in a lot of part of our society, we can found traces of machine learning everywhere. For instance in our email with the spam classification, with some social network with facial detection on pictures, with speech recognition for example from Siri of Apple., and so many other examples can be found, if we search a little bit.

## 1.2 Motivation

### 1.2.1 Mobility Traffic

Why working on mobility traffic data ? What are the utilities ? Simply because of the increases amount movement, not only humain, but also materals or informations. For more than a century, we have not stopped seeing advances in fields; the mechanic, the doctor, computer science and many other sciences. Every time, this advances are reserved for a small number of people, but in what time quickly popularized and accessible for all, especially since globalization. So a lot of things that could be one century ago reseved for rich or a particulary secteur, for instance a car, are now common and widely used. So our technology and also our way to live, now days have particulary influence movements in our society. What we can see that all movement fast now: people, informations, materials,... So we can imagine the amount of data that we could collecte just about a movement of goods in one day in city, and how about the number of letter ships or the people moving to go working. And get all this information can be considered at the technological level, now that some technologie as gps and connected device are so widely use and accessible. All this different data about mobility can be usefull in some ways. If we think only about humain mobility,ew can already think about a lot fields where this data can be useful:
- Traffic management where we can analys the movement a people to better now the most affected areas and think of a way to improve the fuidity or prevent
- Urban planing with beahavior of movement of peaople we can predict the best areas to construct, or maybe think how to increase existing area level
- Consumer profiling: where we can try to learn paterns of consumer to better understand theirs needs or found a better ways to reach them
-

### 1.2.2 Generating Synthetic mobility traffic

So now that we have see that Mobility Traffic Data can be useful in different purpose, why does we need generating synthetic mobility traffic ? Why not just collect directly real data in the real world ? For understand that, we need to understand that for all the the aforementioned fields on mobility traffic, to obtain results that makes sens, we really need a lot of data. Not just go and ask to 100 people to give use their mobitlit data. But so why not just take all what we need ? Here is one of the main problem collect all this data. As saids before at technolgoicly level we could obtain nearly all this data, but in practice we are stopped be the amount of data. It's reallyy difficult for anyone, a city, or an organisation to obtain enough data. Because you need to ask at people directly, so to a lot of people... The second main problem will results of the will of people. Less people will aggres to give freely their mobility data. So they will be always the barrier of privacy. Rare are those who accept to give them all their movement in life. And thinking of pay people to have them mobility traffic data is just not imaginable. So by paying you ou will have people who will particip1, but the city or the organisations need to have a lot of funds. So gereating sythetic mobility traffic can be an interesting thing finally, with it we need less data, so less problem about collecting data.

#### 1.2.3 Random walk model

Geneting data, can be usefull, but why speaking about machine learning for that task, why not use a easier model as the Random Walk Model to generate data ? Yes, a random walk model can be imagine and be apply in our case. It's is a process where the current value of a variable is composed of the past value plus an error term. The implication of this model is that the change of y is absolutely random. So use this model for generate movement, and what will be the results ? Movement totaly random, but people and our society works and thinks not in random way, so this model will not capture reality. It will not capture the real behaviour of users. It's will not seems as a real human movement. And having humain movement data is stricly needed for projects working on mobiltiy traffic of people... So we can rapidely disaprouve and forget the random walk model for our task.

# 2 Littérature

# 3 Approach

All project need to have plan to follow or to try to follow. So the archive my project of create a generating synthetic geospatial trajectories, I divided it in three phases: Prediction, Optimize, Generate.

## 3.1 Predictions

The first and important part of the project is to find a model that predict the trajectories of users based on the previous position(s) of the user. For this we need to check and find a model that perform well and better than classic model as logistic, linear model or random walk model. So we need to try, execute and compare different models. See how they perform on datasets we have available, check where and maybe why they perform better than other. When we are confident in a model and its performances, we can go to the next step to go further.

## 3.2 Optimize

After having selected a model on which work, the next phase is now begin to optimize and deepen the model selected. Begin to switch and compare parameters. So need to find adequate learning rate, number of epochs during the training (number of time that our train data will pass in our model), number of neurones or the number of hidden layer for a neural network, and so one. We need also to check the best way to give him data, and try to go to the limit of the model with the datasets in our disposition. So a big part of testing, switching some littles thing, compare them, and try to leave conclusions and results.

## 3.3 Generate

The next part of the project is about finally to start generating data based on our model. By for example give him a starting position and let the model give us the next position based on how it's predicted it. So

it will be a phase of generating and evaluate the data generated to see if they fit with what we expect and if they seem near the human comportment observed in our datasets.

# 4 Data

For the project, we have access at two datasets: one from Lyon and one from Nokia. Each of them represent position in the time from some users in their everyday life. A lot of other information were collected in this datasets and are accessible for us, as for instance message sends, ...

## 4.1 Nokia dataset

So in this dataset 150 users received smartphones to use and keep all time during ??? Smartphones were collecting all data possible, as sms, phone call, network and many more data. Always in preserving the confidentiality of the users. But data that concern us in priority was geolocation position of the user in time. So basically the longitude, latitude ant time per user. And with this data we can begin to use them and work on our subject.

Starting directly position (logitude, latiude) and time with machine learning is not necessary the best way, it can be really complex and have bad results. Especially, if like me this is the first time that we work on machine learning. So I have first simply the data, by taking only what we call point of interest. This only some area on a map where users stay more than a certain time. So the data which I work now is simply movement between these points of interests. So all this data are put in sequences. We have then a sequences of index, where each index represent on of this zone where user stay more than a certain time. For instance: ???

In futher work we will maybe reuse the first data (longitude, latitude, and time, and also maybe more feature if we think our see that help our case) and all that to perform maybe better or representing our results.

## 4.2 Lyon dataset

# 5 Prediction

## 5.1 Data preparation

As said in the section "Data", I work on a simplification of the problem with sequences of point of interest (that we called POIs). So the first thing was to extract all transitions from users in shape of sequences of POIs. So having all data as:.... Then to rework them to have shape easy and better to work on machine learning, so according with some readings, I choose to represent all position by a vector binary which represent all POIs possible. We obtain a vector of a shape (1,238) because we identify 238 different areas where are useres stay more than a certain time.

## 5.2 Models

Now let speaking really about machine learning, this last decades have allowed really to see the power of computer increase. So it allow to process a large amount of data faster than before. All that made it more interesting to use machine learning, indeed machine learning needs often to work with large amount of train data to be efficient. So this last years have been really good for this fields. Seen more and more people using machine learning, not only in big organisation but also personal. Indeed we don't need any more a computer that we can't afford to make machine learning, but most of personal computer can now handle so machine learning in great way. Moreover a lot courses have been created on the subject and a lot of services allow help us on machine learning, for instance: Tensorflow, Keras, ...

So we can now easy find what we need the work with machine learning. Some models are basicly mainly know and use. We choose some of them for try to predict the next position of the users. Here they are

### 5.2.1 Linear

The linear model is clearly the most know model in statistic and machine learning. It's generally the first model which we begun to test and learn basic stuff in machine learning. So a linear model is basically a model which assumes a linear relationship between the input variables and the single output variables. So, that the output can be calculated from a linear combination of the input variables. So basically the representation mathematic for a simple model is $y = B0 + B1 * x$ ! Where x represent the input, y the output, BO and B1 coefficients.

So learning a linear regression model means estimating the values of the coefficients used in the representation with the data that we have available. We use usually the Ordinary Least Squares procedure to seek to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize.

So we optimize the values of the coefficients by iteratively minimizing the error of the model on our training data. This operation is called Gradient Descent and works by starting normally with random values for each coefficient. The sum of the squared errors are calculated for each pair of input and output values. We also use in this a learning as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error seems to be obtain. Basically we use the model with the coefficient updated and test it on test data and train data to see how it work. And compare training accuracy and test accuracy of the model.

### 5.2.2 Logisitic

Logistic regression algorithm is similar to the previous linear regression model. It is also one of the most famous model. It based mainly on the logistic function, also called the sigmoid function $1/(1 + e^{-}value)$. This function was firstly used to represent relation of the population growth, with the first stage of growth is exponential, then the growth slows, and finish to stop.

So basicly the function in our case can be represent as this $y = e^{(}b0 + b1 * x)/(1 + e^{(}b0 + b1 * x))$!
Where y is the predicted output, b0 is the bias and b1 is the coefficient for the input value. So each column of our input data have coefficient that we need to determine with the train step as I explain before for the

linear model. So we try to optimize cost function at each iteration and update the coefficients values.

### 5.2.3 Neural Network

The basic idea behind a neural network (NN) is to simulate interconnected brain cells inside a computer so we can get it to learn things, recognize patterns, and make decisions in a humanlike way. The amazing thing about a neural network is that you don't have to program it to learn explicitly: it learns by itself, similar to a brain!

The different cells that simulate neurones are classify in 3 different classes: input units (information from the outside world that the network will attempt to learn about), output layer (on the opposite side of the network, is how the NN responds to the information it's learned) and in the middle it is hidden layer (which form the majority of the artificial brain). The connections between one cell and another are represented by a number called a weight. Weights represent influence of one unit has on another. So more the weights is high more the selected cell influence the cell linked.

So how the NN learn really. First it simply us the basic method called feedforward. So each unit receives inputs from the units to its left, and the inputs are multiplied by the weights of the connections in which they go through. Every cell adds all the inputs it receives and triggers the units it's connected to. Secondly, the neural network need to do backpropagation, so a feedback process. In order to compare the output produced with the real output: The NN use the difference between them to modify the weights of the connections between the cells in the NN.

### 5.2.4 Requrent Neural Network

A Requrent Neural Network (RNN) works similar to a simple NN. But when we need some time to have persistence. Need to reasoning about previous events, previous data. It's where RNN goes in the game. We can see a RNN as multiple copies of the same neural network, each passing a message to his successor. So we have a persistence of some past events.

### 5.2.5 Long short-term memory

Long short-term memory (LSTM) is a special kind of RNN, which it's capable of learning long-term dependencies. The LSTM does have the ability to remove or selected and add information pertinent information from past event. It's the main model of RNN used in practice.

### 5.3 Results

#### 5.3.1 Linear

#### 5.3.2 Logisitic

#### 5.3.3 Neural Network

#### 5.3.4 Long short-term memory

### 5.4 Discussion

## 6 Generate

### 6.1 Results

### 6.2 Discussion

## 7 Conclusion