

Big data learning and suggestions in modern apps

Sharma G, Nadesh R K and ArivuSelvan K

School of Information Technology and Engineering, VIT University, Vellore-632014,
Tamil Nadu, India.

E-mail:rknadesh@vit.ac.in

Abstract Among many other tasks involved for emergent location-based applications such as those involved in prescribing touring places and those focused on publicizing based on destination, destination prediction is vital. Dealing with destination prediction involves determining the probability of a location (destination) depending on historical trajectories. In this paper, a destination prediction based on probabilistic model (Machine Learning Model) feed-forward neural networks will be presented, which will work by making the observation of driver's habits. Some individuals drive to same locations such as work involving same route every day of the working week. Here, streaming of real-time driving data will be sent through Kafka queue in apache storm for real-time processing and finally storing the data in MongoDB.

1. Introduction

Life, in the advanced social order, exists in an environment which is inactive and content and usually a pattern is followed when a person relocates within a geographical region. Most relocations involve home, relatives' house, workplace, a preferred cinema hall or a shopping center which are the referred spots. After studying the activity pattern, a prediction system is designed that involves the storage of all the places where a driver could go in order to anticipate arrival.

Plentiful of our driving is routine; in that, we go to similar destinations repeatedly and also follows same routes at the same time (day or a week). Despite better routes being available, which are shorter or faster, we tend to follow the routes used in the past. This was the basis for the idea in this paper.

The aim of this paper involves developing a system involving collection of data based on the driver's destination and the routes taken by them to reach destination. Using the data, the driver's route and destination can be predicted by based on driver's previous routes. These predictions can be used for the following:

- a) In cases of navigation systems, providing better route without involving the driver
- b) By integrating real-time traffic estimates, provision of smarter route guidance could be achieved.
- c) Generate Service Envelope around Potential Driver Trajectory.
- d) Provide POI (Places of Interest) in Service Envelope-
 - Fuel
 - Parking
 - Coffee shops
 - Service stations
- e) Indirect Saving to End User, e.g., Fuel Price during fuel stop V/S Avg. fuel price in country.



To analyze driver's intended destination, we need concepts and tools such as data mining, big data, Apache Storm, Apache Spark, MongoDB etc. Big data information has to be analyzed in certain way to draw conclusions. For stream processing, Apache Storm provides efficient scalable and flexible architecture.

1.1. Unstructured Data

Information that is unorganized in a pre-defined manner is referred to as unstructured data. Unstructured information contains a lot of text and also information such as facts, dates and numbers. So unstructured data is difficult to understand using traditional program.

1.2. Big Data

Big data contains datasets that are very large or complex. So, advance methods are used to extract value from this data. Big data is difficult to share, store and transfer. Hence, advance methods are used to extract valuable data. Accuracy in big data will give better decision making and hence high efficiency reduction in cost and risk. Relational database management systems face difficulties to handle big data. Massively parallel processors are required to handle big data. Big data has properties such as volume, variety, velocity, variability veracity and complexity.

1.3. Apache Storm

Apache Storm is an open-source distributed real-time working out system. Similar to Hadoop's batch processing, Apache Storm makes processing of unrestrained streams of data easy. Storm employs online machine learning, real-time analytics, ETL, continuous computation, distributed RPC. It is fast (above a million tuples processed per second per node). It is easy to set up and function and is also fault-tolerant.

Streams of information are consumed by Storm topology and those streams are processed in subjectively complex customs. However, between each phase of computation, the stream's repartitioning is required.

1.4. MongoDB

High availability, high performance, and automatic scaling is offered by NoSQL database. Being an open-source document, MongoDB documents are similar to JSON objects. MongoDB is a document composed of data structure with field and value pairs.

1.5. Apache Kafka

Being an open-source stream processing platform, Apache Kafka offers high-throughput, unified, low-latency platform for handling real-time information feeds. Its storage layer is an enormously scalable publisher/subscriber message queue architected as a distributed transaction log, makes it extremely valuable for enterprise infrastructures in processing streaming information.

Where X_1 , X_2 , X_3 and X_n are the number of features given as input to the input layer and $Y(X)$ is the output function.

Input:

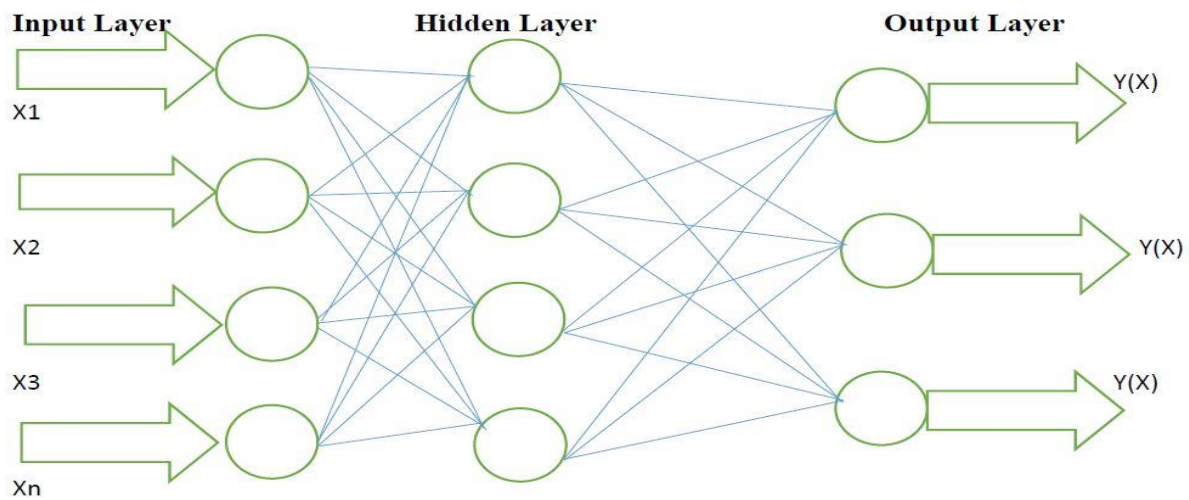
X_1 = Stay-Duration

X_2 = No. of Occurrence

X_3 = Day of the Week

X_4 = Time of a Day

X_n = More Features etc.



Output:

$Y(X) = \text{Home/Office/Unknown}$

In feed-forward neural network, which is an artificial neural network, the connections between the units do not form a cycle. It was the first and simplest type artificial neural network developed. In this, the data moves in only forward direction, (input nodes to the output nodes through the hidden nodes [if any]). No loops or cycles are involved in this network

Figure 1. Feed-forward Neural Network with Back Propagation (For MLPC)

Feed-forward networks characteristics:

1. Perceptrons are organized in layers. The first layer takes the input and the last layer creates outputs. The middle layers are called hidden layers because they have no connection with the external world.

2. Every perceptron from one layer is linked to every perceptron in the next layer. And this results in the data being constantly "fed forward" from one layer to the next, and hence are called feed-forward networks.

3. No connections exist among perceptrons in the same layer.

Multilayer Perceptron Classifier (MLPC) is based on the feed-forward artificial neural network. It consists of multiple layers of nodes and each layer is completely linked to the next layer in the network. Nodes in the input layer represent the input data. Inputs to the outputs are mapped by nodes by carrying out linear combination of the inputs with the node's weights W and bias b that involves applying an activation function. It can be written in matrix form for MLPC with $K+1$ layers as follows:

Nodes in intermediate layers use sigmoid (logistic) function:

Nodes in the output layer use soft-max function:

N (number of nodes) in the output layer relates to the number of classes.

MLPC employs back-propagation for learning the model.

2. Related Work

The Predicting short-term human behavior is a fast-growing area of research. Most of the work involved in the field of intelligent transport systems, has focused on predicting or identifying short

behavior [1], or regarding the construction of intellectual models that involve the reasoning behind how humans make these types of decisions.

[2] Researchers are trying to improve the functionality of the vehicles in order to offer a safer and more beneficial application to driver. The benefits to drivers are that application can automatically support drivers deal with certain unsafe situation.

In the previous studies, researchers had made efforts on the area of destination prediction and behavior study. These studies about destination prediction turn out to be popular since GPS device-equipped smartphone had occurred.

Besides the study above, most of the destination prediction studies are driver oriented and are based on driver's driving history such as in paper [3]. Driver's destination is necessary for the purpose of delivering useful information. For instance, drivers want to know the traffic situation in their route and also want to know services (e.g., petrol station, parking lot, restaurant and etc.) near their destination. By knowing the destination, there is no need to distract driver's attention away from the road by manually inputting destination address. We can still provide this kind of useful information for drivers without driver's manual inputting if the car can predict the destination.

Relevant studies have been made in papers [4, 5]. In both papers, they studied about the real-time destination prediction based on historical GPS logs. The GPS logs that were used in paper [4] were from each individual driver. The researchers built a polygon strip for each trip logs, and then they predicted the trip by matching the driver's current GPS location with those strips. In paper [5], Johan Krumm did not use the GPS logs from a single driver. Instead, he used logs from 118 driving volunteers. He proposed an intuitive predictor to predict the destination based on the most efficient route. He believed that drivers would prefer the most efficient route to the destination rather than the one that costs more time.

The paper [6] made the prediction similar to what had been done in paper [4], but also added map matching to increase the accuracy. By observing driver's driving behavior, it shows that car navigation system is useful but drivers rarely use it. There are several reasons, for example, inputting a destination to the system takes too much time and it will distract the driver's attention during driving.

Another group researchers in paper [7] built a new navigation system that could automatically predict the destination by using probability model. This new navigation system could provide useful traffic information for drivers in real-time without distracting driver on inputting.

Similar to the predicting method that had been used in paper [7], papers [7, 9] also used Bayesian inference to predict the destination. The differences were that they split trips into small pieces, which were called sub-trajectories. The purpose of their studies was to push suggestions to the driver if there was a possibility that the driver would drive into specific areas.

Besides the methods that had been used in the papers above, two main approaches have also been studied: Hidden Markov Model that was used in papers [10, 11] and Decision Tree-based classification in papers [8, 9, 12, 13, and 14]. Both of those two approaches achieved high accuracy of prediction.

By using Hidden Markov Model, researches in paper [10] achieved approximately 98% accuracy in prediction. On the other side, Christian and Raja showed the result of prediction with 96% mean accuracy by using Decision Tree model in the paper [15], which was published in 2013.

3. System Architecture

The flow of the overall system is:

a) Getting data from the GPS devices.

b) Make a REST call :-

1) Sending raw data in JSON format through Kafka Queue into storm for processing and finally storing in MongoDB

2) Getting data from MongoDB

3) Finally sending end result to an application

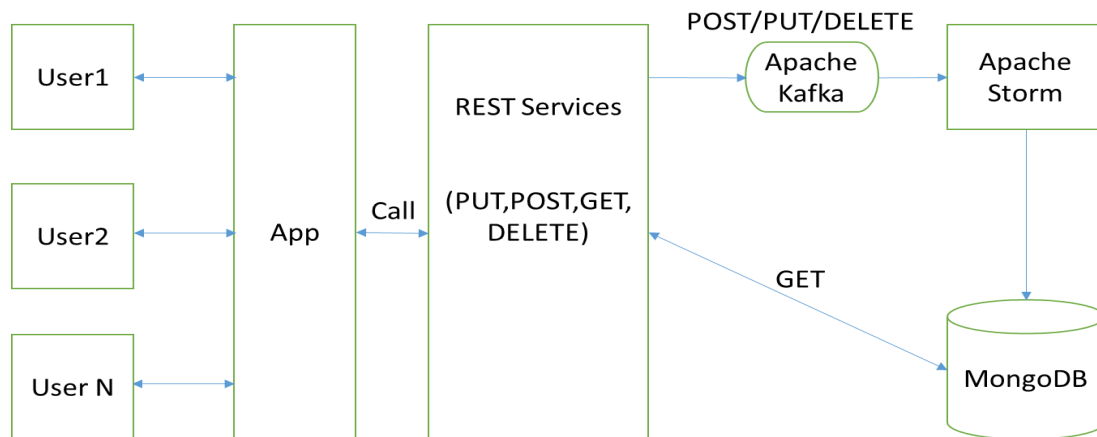


Figure 2.System Architecture

4. Methodology

In this section, the method for predicting driver's intended destination is proposed using Apache Storm Streaming through Kafka queue which is implement in Java. For storage purpose, MongoDB (NoSQL) database is used. This approach uses the feed-forward artificial neural network i.e. Multilayer Perceptron Classifier (MLPC) and is implemented using Apache Spark in java. The algorithm which is used in this method is described as follows:

4.1. Data Set

For each trip in our database, we have the following information:

tripId: which is unique for each trip made by driver
 driverId: Id of the driver who made that trip
 vehicleId: Id of the vehicle using which the trip was done
 start-address: starting address of the trip (latitude, longitude, altitude)
 end-address: end address of the trip (latitude, longitude, altitude)
 start-time: start time of the trip (in UTC)
 end-time: end time of the trip (in UTC)
 status: ENDED, UPDATED

(A trip with ended and updated status can be taken as a completed trip)

Clustering geo-points to refer as a driver location: As the location recorded is in numerical scale, for a small change of location (say 1 mm) the latitude, longitude changes. Here there is a need to group these geo-points to refer some driver's location. We decided that if the distance between two geo-points is less than 200 meters, we will cluster them together as a driver location and refer that location by the centroid of the clustered geo-points. Next step is to identify some features/characteristic of that location so that we can classify the location as Home or Office or some Unknown (not frequent).

Preprocessing raw data: Trip's start and end address is enriched with unique location (clustered geo-point) Id. Start and end time is in UTC and is converted into local time, accordingly. (In Java, I made it dynamic by extracting the time zone based on the start-end geo-points and then, based on that, converted the time to local time). Extract the day information of the trip (Weekday/Weekend). Extract the time information from the trip.

4.2. Generating features for learning

For each driver, get all the locations (given by clustered geo-points) he travelled from his trip records. Count the number of times he started from and also ended to a particular location based on the day of a week and time of a day.

Count the total number of trips by the driver.

Calculate stay-duration.

Stay-Duration: This is the time period (in minutes) a user stays in a particular location. If the end location and start location of two consecutive trip is same, then the difference between end time of first trip and start time of second trip is stay duration of that location.

4.3. *Getting Labelled data for supervised learning*

Now we have driver's travelled locations, and also characteristic features of that location. But we don't have label as "Home" or "Office" which is confirmed by the driver. Created a "Rule-based Algorithm" based on the number of times a driver travelled to a location and also time he spent at that location by basic data analysis and also considering some usual common behavior. Predict/Classify the location using the "Rule-based Algorithm", and ask the driver for approval. If the driver accepts/confirms a location as his "Home" or "Office", then we are using that location information in supervised learning to learn about driver's behavior.

Supervised learning algorithm: We are using the information captured using "Rule-based Algorithm" to feed into a supervised learning model. Now we get more than 98% accuracy using Multilayer Perceptron Classifier (MLPC) model.

5. Results and Analysis

Clustering geo-points within a threshold distance: While clustering the geo-points, We are not using the usual clustering approach as most of the clustering algorithm works on Euclidean distance measure. For geo-points, we have used Haversine distance formula to get the distance between two points. We also find the centroid of each geo-clusters, within 200 meters cluster radius.

Removing outliers for stay duration: For stay duration, we perform outlier analysis so that aggregation measure of stay duration is not influenced by some unusual observations. (Value $> Q3 + 1.5IQR$ and value $< Q1 - 1.5IQR$). Getting aggregate measures of stay duration as features (minimum, maximum, mean, median).

Observing each feature characteristics using summary statistics and their relationship (influence) to the output class.

Looking for multi-collinearity between the features.

Performing mean normalization of the input features.

Predicted/output classes are considered as "Home", "Office" and "Unknown". "Unknown" is a location used by user not frequently, which can be random restaurant, fuel station, friend's place or can be some parking area etc. Our motive is to increase the predicted classes' in future using driver's inputs.

For training, we consider multiple drivers' "Home", "Office" and "Unknown" location's characteristics so that we can get a common pattern to distinguish between Home/Office/Unknown.

Our main aim is to choose a learning algorithm with learning parameters so that we can get highest accuracy of prediction.

We divide the data set into training, cross validation and test set. Training and cross validation set are used to decide the model taking into consideration of over-fitting and under-fitting issues and changing the parameters (changing features, learning rate, regularization parameter) accordingly to get the best model. And afterwards, we chose model where cross validation set is working better. Model is tested again with the test data set.

Multiclass classification using different algorithms such as Random Forest, Decision Tree, Logistic regression, and MLPC are tried to select the best model.

Random Forest gave the features that are important, which helped use to find out the best model.

Models are evaluated by looking at the various criteria like Accuracy, Precision, Recall, Area Under Curve (AUC in ROC curve), F1 score and Log Loss.

Multilayer Perceptron Classifier (MLPC) with 16 features gave us the best model with:

- Accuracy 0.985
- Precision 0.983361
- Recall 0.985000
- F1 score 0.983900
- Log Loss 0.2735
- AUC score 0.9954

We used regularization with penalty 0.7 to get the best Neural Net with gradient descent optimization technique for multiclass classification. The learning rate chosen to get the optimal result is 0.5

6. Future Work

At this moment, the algorithm can perform very well for small set of real-driving data using Spark single node cluster with good accuracy but still it can be extended using Spark multi node clusters to handle huge amount of real-driving data.

7. Conclusion

Efficient solution for predicting driver's intent has been proposed with real-time driving data which will reduce the time as Apache Spark is 100 times faster than Apache Hadoop. In this paper we have presented Multilayer Perceptron Classifier (MLPC: based on feed-forward neural network with back propagation) machine learning model using Spark MLlib which demonstrate an accuracy over 98% of prediction in most cases.

References

- [1] McCall A, Joel C, Wipf D P, Trivedi M M and Rao B D 2007 Lane change intent analysis using robust operators and sparse bayesian learning *IEEE Transactions on Intelligent Transportation Systems* **8**(3) 431-440
- [2] Salvucci D D 2004 Inferring driver intent: A case study in lane-change detection *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting* **48**(9) 2228-2231
- [3] Kostov V, Ozawa J, Yoshioka M and Kudoh T 2005 Travel destination prediction using frequent crossing pattern from driving history *In Intelligent Transportation Systems Proceedings* 343-350
- [4] Ashbrook D and Starner T 2003 Utilizing GPS to study noteworthy locations and predict drive across numerous users *Personal and Ubiquitous computing* **7**(5) 275-286
- [5] Krumm J 2006 Real time destination prediction based on efficient routes *SAE International* 2006-01-0811
- [6] Tiwari V S, Chaturvedi S and Arya A 2013 Route prediction using trip observations and map matching *Advance Computing Conference IEEE DOI: 10.1109/IAdCC.2013.6514292*
- [7] Terada T, Miyamae M, Kishino Y, Tanaka K, Nishio S, Nakagama T and Yamaguchi Y 2006 Design of a car navigation system that predicts user destination *Mobile Data Management 7th International Conference DOI: 10.1109/MDM.2006.67*
- [8] Xue A Y, Zhang R, Zheng Y, Xie X, Huang J and Xu Z 2013 Destination prediction by sub-direction synthesis and security protection against such prediction *Proceedings of The 29th IEEE International Conference on Data Engineering* 254-265
- [9] Xue A Y, Zhang R, Zheng Y, Xie X, Yu J and Tang Y 2013 Desteller: A system for destination prediction based on trajectories with privacy protection *Proceedings of the VLDB Endowment* **6**(12) 1198-1201
- [10] Simmons R, Brett B, Zhang Y and Sadekar V 2006 Learning to predict driver route and destination intent *Intelligent Transportation Systems Conference* 127-132
- [11] Alvarez-Garcia JA, Ortega J A, Gonzalez-Abril L and Velasco F 2010 Trip destination prediction based on past GPS log using a Hidden Markov Model *Expert Systems with*

Applications **37**(12) 8166-8171

- [12] Nguyen L, Cheng H -T, Wu P, Buthpitiya S and Zhang Y 2012 Pnlum: System for prediction of subsequent location for users with mobility *In Mobile Data Challenge 2012 Workshop Dedicated challenge* **2**(2)
- [13] Morzy M 2007 Mining common routes of moving objects for location prediction *International Workshop on Machine Learning and Data Mining in Pattern Recognition* 667-680.
- [14] Monreale A, Pinelli F, Trasarti R and Giannotti F 2009 Wherenext: a location predictor on trajectory pattern mining *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* 637-646
- [15] Manasseh C and Sengupta R 2013 Predicting driver destination using machine learning techniques *In Intelligent Transportation Systems-(ITSC) 16th International IEEE Conference* 142-147