Deep Learning Based Origin Destination Matrix Estimate Through GPS DATA

AYOBAMI EPHRAIM ADEWALE*

University of Tartu
adewale@ut.ee

May 10, 2017

Abstract

Origin Destination matrix have been used over the years by transport agencies to understand and meet transportation demands but with the increase in world's population, previous method Gravity Model, Statistical model and Equilibrum have become inefficient in their OD matrix estimations. The aim of this paper is to describe and take advantage of the recent advancement in deep learning to estimate an efficient OD matrix through Neural Network by training with huge dataset such as a GPS data set.

I. Introduction

One of the main transportation problem is meeting the demand for good transportation system.With the increase in the world's population, it has been difficult to fully understand and fulfill the demand of transportation. Over the years, demand of transportation have been known by presenting it through an Origin-Destination matrix. The Origin-Destination (O-D) matrix is a matrix with rows as origin and column as Destinations that is used to represent the number of trips or volume of travel from one origin to another destination. With an accurate OD matrix, we can understand and predict the traffic flow of a particular region, draw road improvement plan, evaluate previous transportation investment performance and also predict the performance of a future investment.
Previous OD Matrix have been drawn by making use of data obtained from link counts or transportation surveys and estimated through different techniques such as Gravity Model , Statistical Models, Equilibrium Models and Neural Networks. Estimation through link counts is the most popular adopted method because of its near accurate estimation but with the recent increase in the world's population and increase demand for better transportation system, the technique has failed to make accurate estimation and meet current demands, thereby leading to research for more accurate techniques.

In this paper we propose a method for estimating OD matrix through the use of deep learning techniques such as Neural Network and from big data set obtained through Global Positioning System (GPS) to achieve more accurate and reliable OD matrix result.

The remainder of this paper is organized as follows, section 2 talks about the literature review, Deep Learning Techniques in section 3, section 4 discusses the case study and the last section covers the result and conclusion.

II. Literature Review

In [1], Remya et.al took advantage of the computing abilities of Artificial Neural Net-

*keywords: OD matrix, Deep learning, Neural Networks, ITS

work(ANN) which have been proven in various fields such as pattern recognition and fields related to prediction and estimation. ANN was used to develop a new technique that would result into a more accurate O-D matrix estimation. The authors divided the OD matrix estimation problem into three, shortest path estimation, selection of Links and Training of the Neural network. Dijkstra algorithm was used for shortest path selection and the selection of appropriate links was based on some few assumptions and constraint. This was done because the availability of multiple links with unequal cost between any pair zones often affects the computation and accuracy of the OD matrix estimation. To minimize the deviation between the model outputs and target values, neural network Levenberg- Marquardt algorithm was used to train the data set and the performance was measured using Mean Squared Error(MSE). The result showed that the neural network model fits good in the analyzed scenarios but this were subject to several assumptions and constraints which might not give an accurate OD matrix estimate in a different scenario.

In the second paper [2] , Daehyon Kim and Yohan Chang adopted a type of ANN called multi-layer feed-forward network on a backward propagation model to solve the O-D estimation problem using link traffic counts. The result discussed in the paper showed that the backward propagation model provided a better estimation accuracy than the popularly adopted equilibrium-based O-D estimation. The authors mentioned that the model discussed is also suitable for real-time dynamic O-D estimation problem and guessed that it might be more reliable when applied to large complicated road networks with missing and noisy link data. The issues with the method discussed in this paper can be divided into two: first, the assumptions made based on the result of the model must be validated and verified by doing the same test on a large complicated road. Lastly, the result of a neural network model is more accurate when trained with large data set and there is a limit to the size of data that can be acquired through link traffic count. This means that a more accurate OD matrix estimate would have been obtained if the test made use of a huge GPS dataset.

In [3], Gusri Yaldi et.al took an interesting approach by testing the performance of three training algorithms of the NN models when used in generating an OD matrix estimating and the aim was to know the algorithm that would generate the most accurate OD matrix estimate. The NN model was trained with Backward Propagation algorithm, Variable learning Rate Algorithm and Levenberg Marquardt algorithm. The already trained network is then used to predict an OD-matrix of new data set, which has not been used in the training process. The experiment was carried out 30 times and the performance of the three algorithms was compared by looking at the Root Mean Square Error (RMSE) between the observed and the estimated trip numbers. The result of the experiment showed that the NN model trained with LM is better than the other two algorithms. The authors also mentioned that, there are other factors that can affect the performance of the NN model, such as the type of normalization method used. The experiment made use of a work trip data that is based on an home interview survey in Padang City, West Sumatra , Indonesia and it would be interesting to see the output when the test is done with a big data set such as an GPS bus data set because NN models gives better output when trained it big data set.

III. Neural Network Model

Neural Network approach is one of the alternatives OD matrix estimation technique that could overcome the disadvantage of other techniques [3].Neural Network is a Deep Learning technique that based its computation on iterative process. It trains a set of inputs and estimates the output by minimizing the difference between model outputs and expected values. The former is referred to as the training or learning phase which can either be supervised or unsupervised while the later is referred to as

the testing phase, where the estimated output is compared to the expected output. Neural Network is designed to have multiple nodes divided into three layers, the Input layer, the Hidden Layer and the Output Layer. The input layer handles the task of reading the dataset into the model and connects it to the hidden layer, the hidden layer handles the task of actual processing and comparing with the trained data set, then computes the output which is the result.

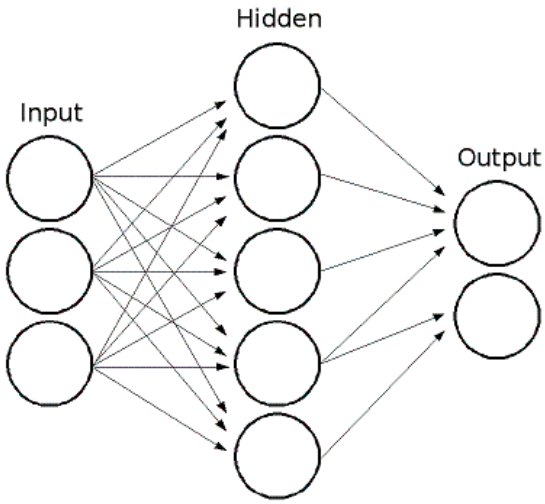

Figure 1: *Simple NN Architecture*

In this paper, the NN model will be trained using the LM algorithm because the result in [3] proved that training NN model with Levenberg-Marqardt (LM) algorithm yields better OD matrix estimate than when trained with other algorithms like Back Propagation (BP), Quickprop and Variable Learning RAte (VLR) algorithms.

The output of the model will be tested by computing the Root Mean Square Error (RMSE) between the expected value and the estimated value.

$$RMSE = \sqrt{\frac{\sum(t_{ij}^t - T_{ij}^t)^2}{z}} [3]$$

Where t_{ij}^t = the observed trip number from origin zone i to destination zone j for testing data.

T_{ij}^t = the estimated trip number from origin zone i to destination zone j for testing data.
z = the number of ij pairs.

IV. Case Study

The model discussed in the previous subsection was applied on a real GPS data set of Dublin buses obtained from the web page of Dublin City Councils traffic control. This data set was selected because of its popularity, availability and most importantly, its suitability for the discussed model.

i. Data Description

Two data sets were used in testing the proposed model, one contains mobility data and locations of context elements, in our case this are location of buses and mobility data are trajectories of buses that run on line 747 of the Dublin bus which is a subset of the Dublin bus GPS data set and the other contained the id and location of all bus stops in Dublin city. Each bus in the bus GPS data set produces data such as latitude and location, bus line Id, Vehicle Id, journey pattern, time stamp, an identifier of the nearest bus stop to the current location, an identifier that indicates if the bus is in a traffic congestion or bus stop, and the delay. This data are broadcast by the GPS attached to the buses at an average interval of 20 seconds.
Bus line 747 which was selected for the test of the proposed model, connects Dublin Airport with a number of popular Dublin locations and it includes 10 bus stops in one direction and 10 in the other direction. [2] shows an Open Street Map view of bus line 747 route for just a single.

ii. Data Cleaning and Preprocessing

As regard to steps required before data processing, obtained data set was cleaned and pre-processed. First step was to convert the time stamps to GMT which was previously in microseconds and an additional field for coordinate system transformations. Second step

was to correct the identifier for the closest bus stop for some fields because some irregularities were seen in this field. Since each recorded bus position was associated with an identifier of the closest bus stop, therefore we had to compute points of actual stops at bus stops for every journey made by a bus. This was done in three ways, first a clustering algorithm called Density-based clustering (DB-SCAN) was used to extract actual actual stops that is, stops at bus stop points for a single journey.

ii.1 DB-SCAN

In Density-Based Clustering, a cluster is defined as dense regions in the data space which is separated by areas of lower density [6]. DB-SCAN algorithm takes in two parameters: 1. Radius of neighborhoods around a data point parameters denoted by ϵ.
2. MinPts: Which is defined as the minimum number of data points needed in a neighborhood to define a cluster.
Using these inputs, DBSCAN then classifies the data points into 3 different categories namely, Core points, Border points and Outliers.
A point is classified as Core point if it as more than the number of points (Minpts) within ϵ [6]. This is defined mathematically as:

$$|Nbhd(p,\epsilon)| >= MinPts[5]$$

where $Nbhd(p,\epsilon)$ is the set of points that are at most distance ϵ away from point p given that $\epsilon > 0$.
A border point has fewer than MinPts within the specified ϵ, but it is in the neighborhood of a core point. Mathematically defined as:

$$q = |Nbhd(p,\epsilon)| < MinPts$$

. Where q is the border point.
Outliers are defined as noise that is points that are neither border points nor core points, they are denoted with $\circ$.

In our implementation, DB-SCAN algorithm is used such that any two or more GPS points located within 20 meters of each other constitute a candidate stop. Hence, the configuration

```
for each o ∈ D do
    if o not classified then
        if o is a core object then
            obtain points reachable from o;
            if points>= Minpts then
                form new cluster;
            end
        end
        else
            Assign Noise;
        end
    end
end
```

Algorithm 1: DB-SCAN Algorithm

was Eps=20 and Min_$Pts = 2$, where Eps represent epsilon distance and Min_Pts represents the minimal points to be considered a cluster. [3] shows the result of the DB scan for a single journey id 747001 with clusters colored in orange. Next was to filter out the invalid stops such as those caused by slow movement and traffic lights. To eliminate them, each cluster's center point was computed and we check for each points if it intersects a buffer of 20 meters around a known bus stop from the bus stop data sets. The 20 meter are selected based on the accuracy of GPS.

iii. Matrix Extraction

Before applying the proposed model, OD matrix for each day for 30 days between January 01 to January 30 will be extracted and making it a total of 30 OD matrix . The purpose of extracting this number of OD matrix is because deep learning techniques gives better estimation and prediction result when trained with more data and this is the main reason why the technique was adopted in this project.

iv. Matrix Estimation

As at the time of the writing of this subsection, the proposed model is yet to be implemented. The idea is to divide the matrix extracted in [iii] into two sets, one for the Training of

the proposed model and the other for the Testing accuracy of the estimate generated by the proposed model. The performance of the new model will be measured by looking at the Root Mean Square Error (RMSE) between the observed and the result of the OD matrix estimated by the new model.

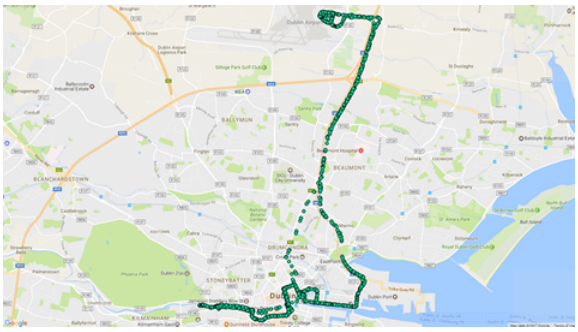

Figure 2: *A single day route for line id 747*

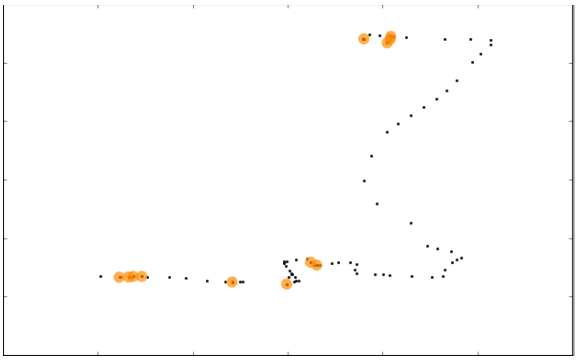

Figure 3: *DB scan result for a single Journey*

V. Conclusion

At the end of this project, a deep learning technique for efficient OD matrix estimation would have been implemented .

References

[1] Remya K P and Samson Mathew. "OD Matrix Estimation from Link Counts Using Artificial Neural Network (2013) " *International Journal of Scientific and Engineering Research*.

[2] Daehyon Kim and Yohan Chang (2011) " Neural Network-based O-D Matrix Estimation from Link Traffic Counts "

[3] Gusri Yaldi, Michael A P and Taylor Wen Long Yue " Forecasting origin-destination matrices by using neural network approach: A comparison of testing performance between back propagation, variable learning rate and levenberg-marquardt algorithms" `http://atrf.info/papers/2011/2011_Yaldi_Taylor_Yue.pdf`

[4] Jean DamascÃÍne Mazimpaka and Sabine Timpf"How They Move Reveals What Is Happening: Understanding the Dynamics of Big Events from Human Mobility Pattern " *International Journal of GeoInformation, January 2017*.

[5] Manojit Nandi "Density Based Clustering" `https://blog.dominodatalab.com/topology-and-density-based-clustering/`

[6] Jing Gao "Clustering: Density Based Method"`https://blog.dominodatalab.com/topology-and-density-based-clustering/` *Lecture Note for State University of New York College, Buffalo* .