Modified Teacher-Student Learning Approach to Predicting Mode of Transportation

Marko Neskovic

University of Maryland, College Park

Abstract

Understanding and being able to predict human mobility is pivotal to many areas ranging from app development to urban planning. In this article I propose a modified teacher-student approach to predicting human mobility, specifically mode of transportation for an individual's trip trajectories. Using output data from a pre-trained "teacher" human mobility model, I train various "student" models to accurately classify modes of transportation in the same way the "teacher" does, achieving a maximum test accuracy of 81.70% on teacher data using a Random Forest model.

1 Introduction

Being able to predict human mobility and its features is crucial to solving various problems, however it takes a lot of research and experimentation to do so. Typically, in order to extract human mobility features, GPS and mobility data must be extracted for an individual or group and state-of-the-art algorithms can be used to make predictions about features like the user's location and mode of transport. Thankfully, research has shown that real human trajectories tend to be relatively temporally and spatially regular [3], so using information like a trajectory's speed, distance, and waypoints can be immensely helpful in extracting key information. In my approach, the teacher model has already used these and other features to extract mobility information, and provides both the features and the labels for our use.

2 Data

2.1 Overview

The data used in this experiment comes directly from Google in the form of Google Location History (GLH). A user's GLH consists of two parts, the actual location history, and the semantic data that Google predicts. The semantic data is the data I used for this experiment, as it summarizes Google's predictions of what locations a user visited, as well as the type of activity they were doing (mode of transportation), all the while including the same metrics it used to make its predictions. This data was chosen because not only does Google perform their own state-of-the-art mobility feature extraction, but it has been suggested that Google location data provides "unmatched individualized human movement information" [7] when compared to currently-available GPS tracker data in terms of accuracy, time-span capabilities, and avoidance of compliance issues.

GLH semantic data consists of two types of observations: a "Place Visit" and an "Activity Segment". We focus on the activity segments as those provide us with Google's transportation prediction. The prediction itself consists of a confidence level ranging from "LOW" to "HIGH" and a list of potential activities which have a key for probability, however Google does not provide the probability and lists everything as 0 probability. For the purposes of this experiment, I aimed to predict the most likely activity in the list instead of all three. In addition to the predicted activity, the segment also includes start and end locations, distance, duration, and the simplified raw path that the user took (locations).

2.2 Tidying and Features

After all of the semantic data was collected, it was cleaned by first extracting the primary activity from the list of predicted activities. This allows us to focus on and predict the singular most likely mode of transportation for a user's trip. Additionally, the duration of the trip was calculated using the start and end timestamps provided in the semantic data. Then, the average speed of the trip was calculated by dividing the distance of the trip by its duration. Finally, entries that had missing data for the necessary features like speed, distance, and duration were removed, as leaving them would be difficult to use and could skew weights in the wrong direction.

For this experiment the features we used were the duration, distance, and average speed of the trip. These are not only the easiest ways to distinguish between different modes of transportation (a train trip is typically further, longer, and faster than a walking trip), but they also are features that could be manually extracted from any location data, and do not limit our model to just Google's data. If we included something like confidence which Google calculates themselves, it would likely make our model less effective when used on non-semantic location data.

3 Methodologies

3.1 Modified Teacher Student Approach

The traditional teacher student method of machine learning involves an agent, the "teacher", advising another, the "student", by suggesting actions for the student as it's learning in a sequential decision problem [8]. Additionally, a primary goal of the teacher student approach is to compress the original teacher model through such knowledge distillation [4].

In this experiment I employ a simple, modified version of the original teacher-student approach. Using output from Google's already trained teacher network (GLH), I train students using that output, thereby training them to mimic the teacher network in their classification, with the end goal being good performance on Google's own data (like a student taking a teacher's test) and then application to other semantic or location data. This is also done in order to compress Google's network into a much simpler model that is easier to understand while still predicting the mode of transportation in a similar manner.

3.2 Models

3.2.1 Support Vector Machine

A support vector machine (SVM) is a machine learning algorithm that classifies by separating classes using a hyperplane, with the goal being to maximize the margin between points in either class and the hyperplane [5]. SVM's also include support for "soft margins", a parameter to determine how many points can "bleed over" onto the other side of the hyperplane, as well as a kernel function, which changes how the hyperplane is determined hence allowing for data that are linearly separable in higher dimensions as well as non-linearly separable data. Research has demonstrated that Support Vector Machines have outperformed other models such as multinomial logit in travel mode prediction [6], so they are a natural candidate for this experiment.

3.2.2 Random Forest

Popular for its ease of use and great results without the need for much hyper-parameter tuning, random forest (RF) [1] is an ensemble machine learning method that works by training many decision trees on subsets of the training data, and combining their results to solve the problem, typically either by averaging them in a regression task, or majority vote in a classification task. Research has suggested that RF performs similarly to or better than SVM and other models in terms of both accuracy and efficiency when it comes to travel mode prediction [2]. In particular, RF's superior advantages come in its robustness, particularly its capability of handling different variables and modeling non-linear relationships such as the ones in our problem.

3.2.3 Artificial Neural Network: Multi-layer Perceptron

The multi-layer perceptron (MLP), sometimes used to refer to any kind of Artificial Neural Network, is a type of neural network that consists of an input layer, one or more hidden layers, and an output layer. Inputs start with weights, and these weighted inputs are passed through an activation function which can be linear or non-linear, finally reaching the output layer which is responsible for outputting a vector in the format needed to classify.

Research has suggested that, like SVM's, Aritificial Neural Networks (ANN), specifically MLP's outperform various other models in travel mode prediction tasks. Additionally, MLP tends to outperform SVM, MNL, and RBF ANN overall, specifically when predicting car trips and public transportation trips [6], which make up the bulk of our dataset.

4 Implementation and Results

4.1 Model Accuracy and Tuning

First, baselines for each model were implemented to get a sense of how the models generally perform against each other, and which one is the best for our problem overall. All training and testing was done on 80/20 splits, with accuracy being the mean accuracy given the test data and labels. Additionally, the Support Vector Machine and Multi-Layer Perceptron Models had the training data normalized beforehand as both are susceptible to differing scales.

- **Support Vector Machine:** a radial basis function kernel is used as the data is not linearly separable, the regularization parameter is 3 for some increased regularization and less support vectors, and gamma (influence of each training example) is set to 0.1 to prevent overfitting of the data.

- **Random Forest:** the baseline RF uses 100 trees and Gini coefficient to measure the quality of each split. There is also no limit to the depth of each tree, and trees are expanded until all leaves are pure or contain less than two samples. Finally, there are no limits on the number of nodes or the impurity needed to stop tree growth early.

- **Multi-layer Perceptron:** The baseline of the multi-layer perceptron contains two hidden layers, each of size 100. The activation function for these hidden layers is the rectified linear unit function (RELU), and the solver for the weight optimization is the limited-memory BFGS algorithm, which tends to work well on smaller datasets. The maximum iterations was set to 500 so that it can properly converge. Finally, the learning rate is adaptive, changing from constant when epochs don't decrease training loss. The network is trained using back-propagation.

After comparing the baseline models, it was evident that Random Forest was the most appropriate for this dataset. With no tuning, it achieved a test accuracy of around 79.15%, compared to around 71.28% with the multi-layer perceptron, and just 48.72% with SVM. As such, I chose to continue working with and tuning the Random Forest model to achieve the highest test accuracy possible.

Two crucial parameters of the Random Forest that I chose to focus on were the number of trees in the forest and maximum depth of each tree. The number of trees was an arbitrary 100 for the baseline, and max depth was set to limit when all leaves are pure or the leaves contain less than two samples. Iterating

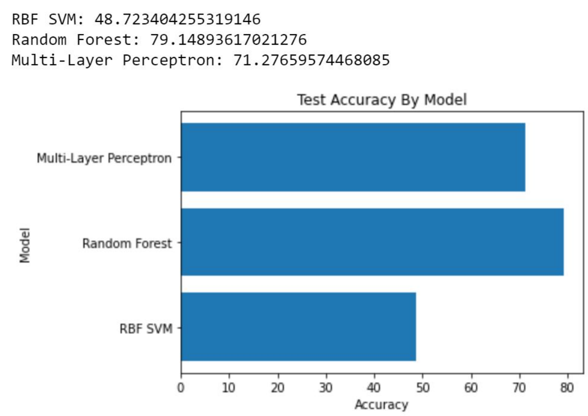

Figure 1: Performance of our three models, clearly showing that Random Forest achieved greater test accuracy than the other two

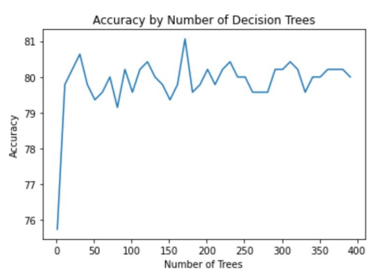

Figure 2: Graph showing accuracy vs. number of trees, indicating initial growth, a small spike around 171 trees, and slight fluctuation after

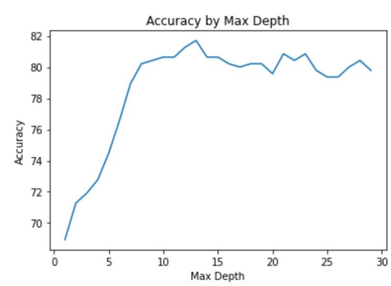

Figure 3: Graph showing accuracy vs. maximum tree depth, showing initial growth, a peak around 13 nodes deep, and tapering off at the end

through the number of trees from 1 to 400 (in intervals of 10) achieved a maximum test accuracy of around 81.06% at 171 trees. Following this and using the optimal 171 trees, iterating from 1 to 30 nodes deep for the maximum depth achieved a best combined test accuracy of 81.70% at 13 nodes deep: a 2.55% improvement over our baseline random forest model.

4.2 Classification Analysis

In addition to maximizing the accuracy of our model, learning how well it classifies certain labels is crucial to understanding how it works and what improvements can be made in the future. Out of the labels CYCLING, IN_BUS, IN_PASSENGER_VEHICLE, IN_SUBWAY, IN_TRAIN, IN_VEHICLE, MOTOR-

Label	Precision	Recall	F1	Support
CYCLING	0.00	0.00	0.00	6
IN_BUS	0.00	0.00	0.00	3
IN_PASSENGER_VEHICLE	0.81	0.78	0.8	146
IN_SUBWAY	0.00	0.00	0.00	1
IN_TRAIN	0.56	0.60	0.58	15
IN_VEHICLE	0.95	0.79	0.87	97
MOTORCYCLING	0.00	0.00	0.00	2
UNKNOWN_ACTIVITY_TYPE	0.00	0.00	0.00	7
WALKING	0.75	0.91	0.82	193

Table 1: Table showing the precision, recall, f1 score, and support of our model on each label found in the testing subset

CYCLING, UNKNOWN_ACTIVITY_TYPE, and WALKING, this model either classifies relatively well or terribly, which seems to be directly related to that label's frequency in the dataset.

The poorest performing labels were CYCLING, IN_BUS, IN_SUBWAY, MOTORCYCLING, and UNKNOWN_ACTIVITY_TYPE, which were all predicted with 0 accuracy. The commonality between these labels is how rare they are. The support (number of samples of the true response that lie in that class/label) for all of these was incredibly low (under 10). This means that not only are they not prevalent in the test set, but they are also rare in the dataset overall. With such rarity our model does not see enough examples to properly train and label instances of these classes. Also while not as bad, the IN_TRAIN class is the worst performing label out of those with above-0 accuracy, and out of those it also has by far the lowest support. Again, while our model may have seen some examples to train on, it was simply not enough to make an accurate enough prediction.

For the three most prevalent labels in the dataset, IN_PASSENGER_VEHICLE, WALKING, and IN_VEHICLE, the model was able to classify relatively well. These are obviously the largest contributors to model's overall accuracy. WALKING had lower precision and higher recall, indicating that the model labels many trips as walking, even when they are not walking. Even so, it's accuracy was still good. As for IN_VEHICLE and IN_PASSENGER_VEHICLE, both had similar recalls, however IN_VEHICLE had incredibly high precision, meaning that while it missed some instances of IN_VEHICLE, almost all of the trips our model labeled as IN_VEHICLE were truly IN_VEHICLE. This is presumably because of the ambiguity that comes with the label, as IN_VEHICLE refers to any vehicle that is not a passenger vehicle. Since they both travel in similar manners, it is likely that our model often misclassified IN_VEHICLE as IN_PASSENGER_VEHICLE.

Overall, the model classifies well. It correctly predicts the most prevalent labels in the dataset with good accuracy, albeit struggling slightly due to the ambiguity between passenger and non-passenger vehicles. There are a number of labels that it completely misclassifies, but this is largely due to their rarity within the dataset, and simply using a larger set of data with more trips with such labels should improve the model's performance.

5 Conclusion and Further Work

This modified teacher student approach shows promise in accurately predicting the mode of transportation of an individual, while simplifying and condensing the original model. I demonstrate that by using data from a state-of-the-art, pre-trained model, a random forest student model can be trained to mimic the original with very high accuracy, making it nearly as effective without the original's complexity. Additionally, not relying on teacher-specific features like Google's correlation feature suggests that, assuming Google's model performs well on real location data, the student will as well once the necessary features like distance, duration, and speed have been extracted.

Future work related to this would first be to work with larger, and more diverse datasets. The dataset used in this experiment was relatively small, and the accuracy on much larger (but similar) datasets may change, especially when more modes of transportation are added as labels, and modes which are not as prevalent in this data become more prevalent. Additionally, this experiment did not account for the true mode of transportation of the user, as the data was sampled during random periods over years of travels. It would be extremely beneficial to see how accurately Google's original model is able to predict the mode of transportation of a trip using the true label that the user would have to keep track of, and then compare it to random forest (student) model's performance on the same data. Finally, a crucial feature of mobility data is the actual geolocations forming a user's trip. In the context of transportation prediction, a geolocation along a certain road (like a highway), near a park, or in the middle of the ocean, can all be indicative of different modes of transport. Due to time constraints and the difficulty of determining such information from latitude-longitude waypoints, this feature was not incorporated in the experiment, but should be an integral piece to improving the model going forward.

References

[1] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[2] Long Cheng et al. "Applying a random forest method approach to model travel mode choice behavior". In: *Travel behaviour and society* 14 (2019), pp. 1–10.

[3] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. "Understanding individual human mobility patterns". In: *nature* 453.7196 (2008), pp. 779–782.

[4] Seyed Iman Mirzadeh et al. "Improved knowledge distillation via teacher assistant". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 5191–5198.

[5] William S Noble. "What is a support vector machine?" In: *Nature biotechnology* 24.12 (2006), pp. 1565–1567.

[6] Hichem Omrani. "Predicting travel mode of individuals by machine learning". In: *Transportation Research Procedia* 10 (2015), pp. 840–849.

[7] Nick Warren Ruktanonchai et al. "Using Google Location History data to quantify fine-scale human mobility". In: *International journal of health geographics* 17.1 (2018), pp. 1–13.

[8] Matthieu Zimmer, Paolo Viappiani, and Paul Weng. "Teacher-student framework: a reinforcement learning approach". In: *AAMAS Workshop Autonomous Robots and Multirobot Systems*. 2014.