GPS Trace Modality Classification

CS-229 Project Report

December 16 2016, Stanford University

Diana Juarez Madera, Matej Kosec, Yi Cao {djuarezm, mkosec, ycao4}@stanford.edu

1 Introduction

Open Street Map (OSM) is the most popular open-source map around the world. Users can download map vector data for free, and contribute back by uploading their edits. Recently, OSM has allowed users to upload their GPS traces to assist in the map digitization process. Inferring road attributes from trace behavior can extend the maps to include features such as the average speed of a road when traveling by car. Such data can be used to give users an accurate estimate of trip duration. The first step in this process is detecting whether users are walking, traveling by car, bicycle, train etc. This project consequently investigates the viability of using both unsupervised and supervised learning techniques to detect the transportation mode of GPS traces.

In section 2, an overview of previous work on GPS trace classification is presented. Section 3 provides information on the training set and an explanation of how the features were extracted and engineered. Additionally, it covers the use of t-SNE for evaluating different sets of features. Section 4 explains the overall machine learning methodology employed in the project. This includes both unsupervised labeling through k-means and mixture of Gaussians, and performance validation through supervised learning. The qualitative and quantitative results are presented in section 5, including the outcome of cross validation and feature selection routines. Finally, section 6 provides conclusions and ideas for potential future work.

(a) OSM network and GPS traces (red) in San Francisco downtown

(b) Sample Walking and Driving trips in San Francisco downtown.

Figure 1: Visualization of OSM map and GPS trips.

2 Related Work

Several digital mapping techniques based on massive GPS traces have been proposed and demonstrated by the OSM community [8]. The quality of these works depends on how well GPS noise is handled. Yuan et al. inferred road geometries by applying Gaussian Kernel to GPS point density distribution [9]. This approach can filter out outliers that fall outside of 3σ, but it heavily relies on the strong assumption that the density distribution is Gaussian or at least unimodal. This is usually not the case at intersections or places where GPS traces are sparse. Another approach is matching GPS trace to segment sequences and apply statistical analysis on map-matched results. This approach better fits the OSM data feed as data coverage is relatively poor due to its open-source nature (see Figure 1a). Shafique et al. adopted machine learning algorithms and classified GPS trace modality collected on smart-phones [4]. It is shown that SVM provides a higher prediction accuracy compared to other classifiers. For this reason, SVM was selected as the performance baseline against which unsupervised learning was compared.

3 Data and Feature Engineering

The approach combines two primary data-sets: GPS traces and OSM map data. These are freely available through the OSM GPX portal and OSM website [6, 7], and extracted through open-source parser [2]. The OSM map data includes road geometry and building footprints. In combination with the GPS traces, the topographical map information is used to generate geospatial features, including distance between GPS waypoints and nearest road network along the trip, and intersections with building polygons. These features are based on the intuition that if a trip's distance to road has large variation and frequently intercepts building polygons, it is likely a pedestrian trip (see Figure 1b).

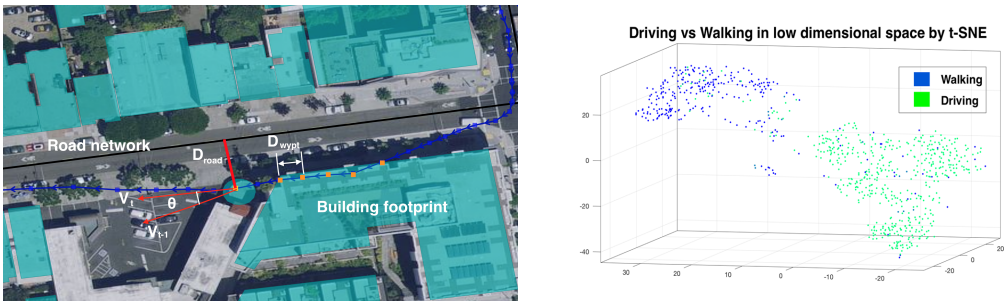
3.1 Feature Engineering

The raw GPS data contains spatial and temporal information for each GPS point, i.e. latitude, longitude and time-stamp. First order finite differences were used to obtain estimates of trip speed, distance covered, and heading changes. Each GPS trace is characterized by a set of minimum, maximum, and variance statistics. However, features such as maximum

speed and maximum acceleration were found to be too noisy and were later disregarded. Ultimately, only 16 features were used for clustering. The full list is given in table 1 and includes 11 GPS-based and 5 OSM features.

3.2 Data screening with t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) [1] is a dimension reduction tool that is widely used in visualizing high dimensional data-sets. It projects high dimensional data points into low dimensional space by minimizing Kullback-Leibler divergences between the original high dimensional distribution and its low-dimension projections. Due to the high dimension of the data the team used t-SNE to examine its structure. t-SNE performs PCA first to reduce dimension, and specifies an empirical value for perplexity. To determine the optimal value for these two parameters, t-SNE was run with a range of different values. Figure 2b visualizes the (clustered) structure of data in 3 dimensional space. Two major clusters can be visibly identified as corresponding to the walking (left) and driving mode (right). However, both clusters penetrate each other to some degree. This phenomenon can be explained by two facts. First, there are slow driving and running that are intrinsically similar, so it is difficult to separate them. Second, there are trips of hybrid mode and are not properly segmented in the preprocessing.



(a) Derived features based on GPS data.

(b) Visualization of clusters using t-SNE

Figure 2: Visualization of data cluster using t-SNE.

4 Methodology

Figure 3 gives an overview of the machine learning methodology that was implemented. The key aspect of the approach is the coupling of two distinct data-sets: the GPS traces, and the OSM topographical map data.

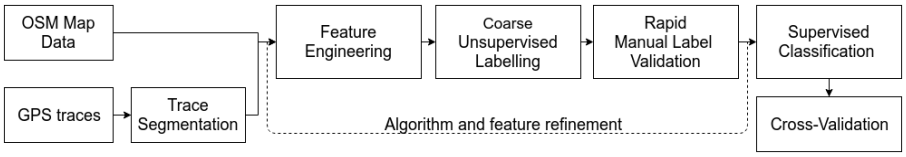


Figure 3: Project methodology overview.

To decrease the chance of hybrid mode, each GPS trip was segmented based on spatial and temporal distance. If two consecutive points are 50 meters apart in distance or 60 seconds apart in time, it is split.

4.1 Unsupervised labeling

In order to extract useful information (such as mean road speed) that can be used in improving digital maps, it is first necessary to classify trips according to the mode of transport being used. This means for instance, distinguishing between driving and walking trips. Approaching this problem through manual labeling is highly time-consuming. The 807 traces used in this project required over 6 hours of work to manually classify. In order to be able to analyze 10s and 100s of thousands of traces, it is necessary to automate the labeling procedure by using unsupervised learning techniques. It is hoped that these can either speed-up, prioritize, or replace manual labeling.

4.2 k-means clustering

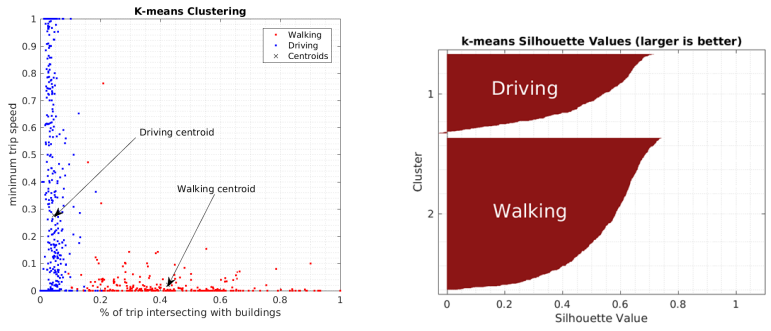
As the first unsupervised approach to be tested, k-means was used to create two separate clusters in the (normalized) 16-dimensional feature space. The k-means algorithm used was the standard MATLAB [4] implementation with squared Euclidean distancing. Since k-means can only be guaranteed to converge to a local optimum, the 'Repllicates' option in MATLAB was used to initialize the k-means at 50 different random starting points. The most optimal k-means used for labeling is summarized in Table 1.

To further evaluate the quality of the k-means clustering, the MATLAB silhouette plot [3] was used (shown in Figure 4b). For a point indexed 'i' in the walking cluster, the silhouette value is computed as:

$$S = \frac{avg\_distance\_driving(i) - avg\_distance\_walking(i)}{\max(avg\_distance\_driving(i), avg\_distance\_walking(i))} \quad (1)$$

Where the function 'avg_distance_driving' calculates the average distance between the current (walking) trace 'i' and all traces in the driving cluster. Similarly, 'avg_distance_walking' calculates the distance between point 'i' and all other points also in the walking cluster. A small or negative negative silhouette value therefore indicates that the point may be more similar to points in the other cluster than its own. On the other hand, a number close to positive 1 suggests that a point is very likely in the appropriate cluster. Silhouette values were included in this project primarily as a potential means to prioritize traces for manual labeling (instead of posterior probabilities).

Figure 4a further demonstrates the advantage of combining OSM and GPS features. On the vertical axis it is shown



(a) Potential effectiveness of the OSM data

(b) Silhouette plot for k-means clustering into 2 modes.

Figure 4: K-means and feature quality when using OSM and GPS data

that driving trips have a high variance in the mean trip speed and low variance in the percentage of the trip which intersects a building. Walking trips show the opposite pattern, and have very low variation in the mean trip speed, but a high variation in the percentage of the trip which intersects a building polygon. This indicates that the OSM data may be able to provide variance in directions almost parallel to the ones in which the GPS data has the least variance. Tt is found that the centroid of the mean speed for the driving cluster is 23 mph, while for walking it is only 2.45 mph. For OSM features, an average 27.8% of a walking trip and 2.13% of a driving trip lies withing a building polygon. These values are reasonable and give credibility to the validity of the clustering.

Table 1: Table summary of k-means centroids when using OSM and GPS data

	Speed [mph]		**Acceleration [mph^2]**		**Distance bw. waypts [m]**			OSM **Bldg. intersec.**
Feature	*mean*	*std.*	*mean*	*std.*	*mean*	*max*	*std.*	*percentage of trip*
Walking	2.45	1.63	0.561	0.724	4.88	16.6	2.75	27.8
Driving	23.0	11.9	1.28	1.31	15.26	38.9	7.88	2.13
	Δ Heading [deg/m]		**Δ Heading [deg/pt.]**		OSM **Distance to road [m]**			
Feature	*mean*	*std.*	*mean*	*std.*	*min*	*mean*	*max*	*std.*
Walking	67.2	150.6	25.8	31.1	133.4	192.7	361.4	71.8
Driving	32.4	152.4	8.70	18.7	29	171.4	1072	222

4.3 Labeling using a Mixture of Gaussians

While k-means appears to cluster the data quite well, it does not provide a probabilistic estimate of a trace being in a certain cluster. If unsupervised labeling is to replace manual labeling, it is desirable to use posterior probabilities to automatically prioritized traces for manual labeling. The project used the MATLAB implementation [10], with expectation maximization (EM), and a full covariance matrix. Since the EM algorithm can only be guaranteed to converge to a local optimum, the 'Replicates' option was used to rerun EM from 50 random initialization values.

Similarly to k-means, each cluster can be roughly characterized by the mean vector of its Gaussian distribution. This information is contained in table 2. Looking at mean speed, we find that the average value for the driving cluster is 18.8 mph, and 2.41 mph for walking. This means, that in terms of mean speeds the two clusters are closer together than in the k-means clustering approach. In practice this reduces the ability of the algorithm to distinguish between driving and walking on the

Table 2: Coordinates of the mean of each of the two Gaussians

Mode	**Speed [mph]**		**Acceleration [mph]**		**Dist. bw. waypts. [m]**			OSM **Bldg. intersec.**
Feature	*mean*	*std.*	*mean*	*std.*	*mean*	*max*	*std.*	*percentage of trip*
Walking	2.41	1.68	0.580	0.755	4.39	15.5	2.55	32.2
Driving	18.8	9.68	1.10	1.41	13.95	35.8	7.07	1.68

Mode	**Δ Heading [deg/m]**		**Δ Heading [deg/point]**		OSM **Distance to road [m]**			
Feature	*mean*	*std.*	*mean*	*std.*	*min*	*mean*	*max*	*std.*
Walking	76.1	172.2	26.5	31.7	130.2	179.7	271.2	40.9
Driving	27.9	123.7	11.3	20.5	24.6	193.1	1043	231

basis of mean speed. The trend of reduced separation between the two clusters is consistent across all GPS based features. However, for the OSM-based features the mean values of the two clusters are actually further separated. For instance, the building intersection percentage for walking is 32.2% (only 27.8% under k-means) and 1.68% for driving (2.13% under k-means). As such, the mixture of Gaussians approach appears to better utilize the OSM based features.
Lastly, the covariance matrices of the two distributions are shown in Figure 5. As was anticipated from the k-means analysis, the walking cluster shows much more variance in the percentage of the trace which intersects a building, while the driving traces have the highest variance in the mean speed and other GPS features. This insight can be used to engineer new features which target increasing the accuracy . That is, in order to improve walking classification, more OSM-based features should be added.

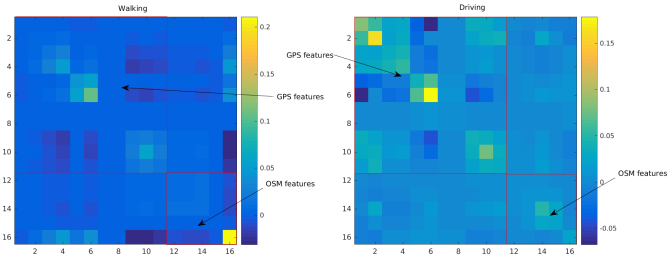


Figure 5: Magnitude of covariances between features in both clusters

4.4 Supervised Classification

For the reasons sated in section 2, SVM was chosen as the benchmark classifier. A SVM constructs the best hyper-plane in a high dimensional space that provides a separation frontier (also called decision boundary) between data points of two different classes, therefore classifying the data. The best hyper-plane is the one with the largest margin between the two classes. The data points that are closest to the separating hyper-plane are called support vectors.

The establishment of a comparison benchmark for the performance of unsupervised learning consisted on the training of a SVM on the 807 manually obtained labels with the standard "fitcsvm" MATLAB implementation [11]. The positively labeled examples (+1) were used to indicate data points of "Walking" mode and the negatively labeled ones (-1) corresponded to data points of "Driving" mode. To evaluate the SVM performance in this particular application to GPS trace classification, a 10-fold cross validation was implemented in MATLAB using "crossval" and "kfoldLoss" [12]. Since it was suspected that some of the features were superfluous, we decided to mitigate over-fitting by applying a feature selection algorithm to reduce the number of features and find the most relevant.

5 Results and Discussion

The clustering performance of the unsupervised algorithm is visualized in Figure 6. Here the clusters are visualized on 2D t-SNE plots. On a qualitative basis seems that the k-means follows the cluster patters set-out by manual labeling much better than the Gaussian mixture. However, the manual labeling indicates that some driving trips should be present 'inside' the walking cluster. While, k-means does not seem to be able to capture this very well, the mixture of Gaussians does quite well in this regard. Lastly, the plot on the very right demonstrates the probability of a given trace being associated with walking. Most traces have probabilities between 0 and 0.1 or 0.9 and 1. Thus, it may not be possible to select which traces should be manually labeled just on a probabilistic basis (which was the original motivation for using mixture of Gaussians over k-means).

Table 3 shows that k-means accurately clustered 95.4% of walking trips, and 88.2% of driving trips. Cumulatively

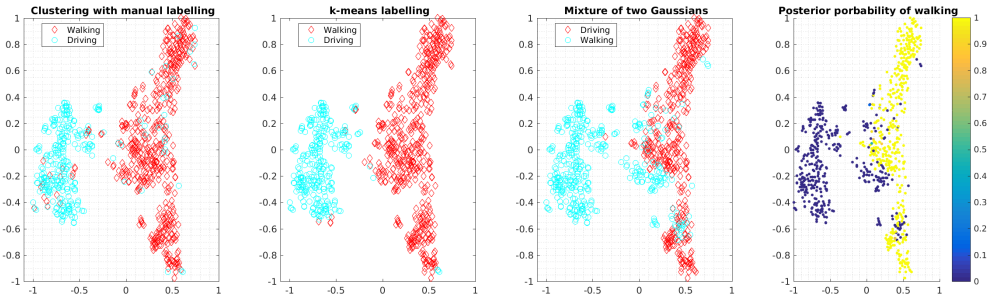


Figure 6: Comparison of manual, k-means, and Gaussian mixture clustering visualized on 2D t-SNE plot

this results in an error of just 7.2% in the classification of trips as either walking or driving. Note that the accuracy of the k-means decreases significantly if mixed modes of transport are present. It turns out these are quite common in reality. For instance it is common to walk to the train station, and then take the train to the city. The mixture of two Gaussians performed substantially worse and only achieved an accuracy of 86.6% overall.

Table 3: Unsupervised classification accuracy for both k-means and mixture of Gaussians

	k-means		**Gaussian mixture**	
	percentage	*trips*	*percentage*	*trips*
Correctly classified walking trips	95.4	496 of 520	83.2	433 of 520
Correctly classified driving trips	88.2	253 of 287	90.9	261 of 287
Total of correctly classified trips	**92.8**	**749 of 807**	**86.0**	**694 of 807**

Figure 7 shows the performance of the supervised learning application. The results of the 10-fold cross validation and feature selection on all 21 original features show that the smallest generalization error achieved was 5%. This corresponds to 4.9% training error with a subset of 6 GPS-derived and 2 OSM-based features. The final feature list includes GPS-based statistics on speed, acceleration, distance between way-points and heading changes. It also includes the OSM statistic on distance to road. However, surprisingly does not include the OSM feature of building intersection percentage. The error performance demonstrates the typical bias-variance trade-off.

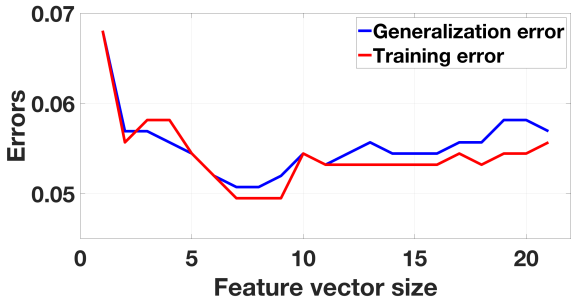


Figure 7: Generalization and training error as a function of number of features.

Overall these results show that k-means is nearly as effective at labeling data, as SVM is in training and generalization. As such, k-means can be valuable in speeding-up the labeling of GPS traces of trips. On the other hand, the mixture of Gaussians would require further refinement before it could be considered useful.

6 Conclusions and Future Work

The project demonstrated the effectiveness of unsupervised labeling for the classification of GPS trace modality based on a combination of GPS and OSM features. While both methods yielded high accuracy, k-means proved to be superior despite not providing posterior probabilities as a result. Prioritization of traces for manual labeling can still be achieved in k-means by using silhouette values. t-SNE was used as starting point for testing different sets of features and enabled a clustered visualization of the data. Further validation of the feature selection was carried out through supervised learning (SVM). Generalization and training errors were computed through 10-fold cross validation and a recursive feature selection routine. While this project serves as a good proof of concept, future work should extend the method to a larger set of transport modes including: rail, cycling, flying, and ferries. Additionally, upgraded and refined features to improve the accuracy should be investigated. This will require the acquisition of a more extensive data-set, perhaps of the entire United States.

References

[1] L.J.P. van der Maaten and G.E. Hinton, Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.

[2] S. John, S. Hahmann, A. Rousell, M. Loewner, A. Zipf, Deriving incline values for street networks from voluntarily collected GPS traces. Cartography and Geographic Information Science (CaGIS), 2016.

[3] Silhouette plot - MATLAB silhouette. (2016). Mathworks.com. Retrieved 20 November 2016, from https://www.mathworks.com/help/stats/silhouette.html

[4] M. A. Shafique, E. Hato, A Comparison among various Classification Algorithms for Travel Mode Detection using Sensors' data collected by Smartphones. CUPUM, 2015.

[5] k-means clustering - MATLAB kmeans. (2016). Mathworks.com. Retrieved 20 November 2016, from https://www.mathworks.com/help/stats/kmeans.html

[6] Zverev, I. (2016). Non-private GPS traces in OpenStreetMap. Zverik.osm.rambler.ru. Retrieved 20 November 2016, from http://zverik.osm.rambler.ru/gps/files/extracts/index.html

[7] Download OpenStreetMap data for this region: California. (2016). GeoFabrik GmbH. Retrieved 20 November 2016, from http://download.geofabrik.de/north-america/us/california.html

[8] 9 Years of OpenStreetMap GPS Tracks Available for Mapping. October 2013 https://www.mapbox.com/blog/openstreetmap-gps-layer/

[9] J. Yuan, A. M. Cheriyadat. Image feature based GPS trace filtering for road network generation and road segmentation. Machine Vision and Applications, 27:1–12, 2016.

[10] Fit Gaussian mixture distribution to data - MATLAB fitgmdist (2016). Mathworks.com. Retrieved 15 December 2016, from http://www.mathworks.com/help/stats/fitgmdist.html

[11] Train a SVM classifier - MATLAB fitcsvm (2016). Mathworks.com. Retrieved 9 December 2016, from https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5o09

[12] Cross-validated support vector machine classifier - MATLAB crossval (2016). Mathworks.com. Retrieved 9 December 2016, from https://www.mathworks.com/help/stats/classificationsvm.crossval.html