Neural-network categorization of unlabelled transit trajectories from MTL Trajet

By Tyler Sloan

Near the end of September of 2017, the City of Montréal's Open Data Portal released the data that was collected in 2016 using the MTL Trajet smartphone application. The purpose of this project was to collect GPS coordinates with related metadata, in order to better understand how citizens navigate through the City of Montréal and by what mode, in part in order to see the influence of events such as street closures from construction.

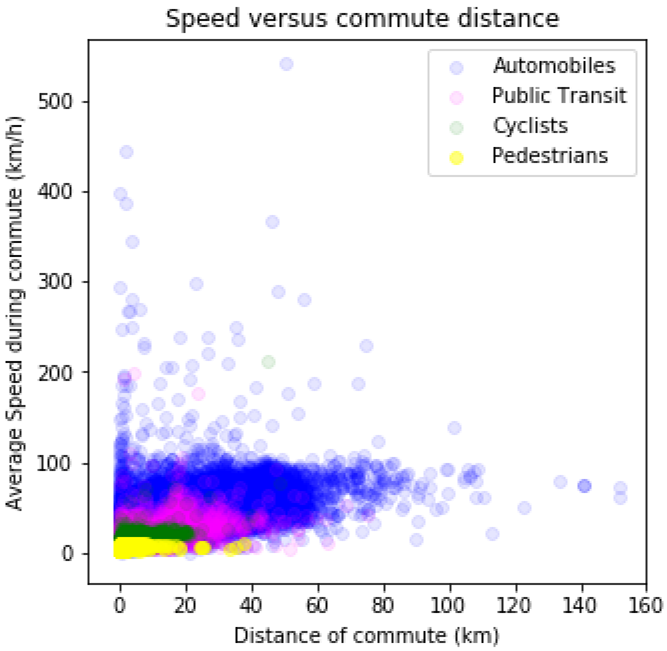
This rather expansive data set captured > 320,000 individual trajectories, which were anonymized and hosted online through the portal. Within the dataset, the trajectories also included information about the mode of transit, and the purpose of the journey. Unfortunately however, a relatively small proportion of the recorded trajectories were labelled to indicate the mode of transit, leaving >80% of this metadata missing.

The point of this article is to use some entry-level machine-learning techniques to try and guess the missing labels, to see if the unlabelled data can be used. Throughout the article, I will refer to the trajectories which contained user-defined labels as the ***'labelled'*** data or trajectories, whereas the labels guessed using the neural network will be referred to as ***'categorized'***.Although I had originally planned to use this as a way of getting an introduction to machine learning in Tensorflow and Keras, I decided to take baby steps and start with SciKit-Learn in Python. The Jupyter Notebook containing the full exploratory analysis can be found on my Github page tsloan1377/montreal_open_data.

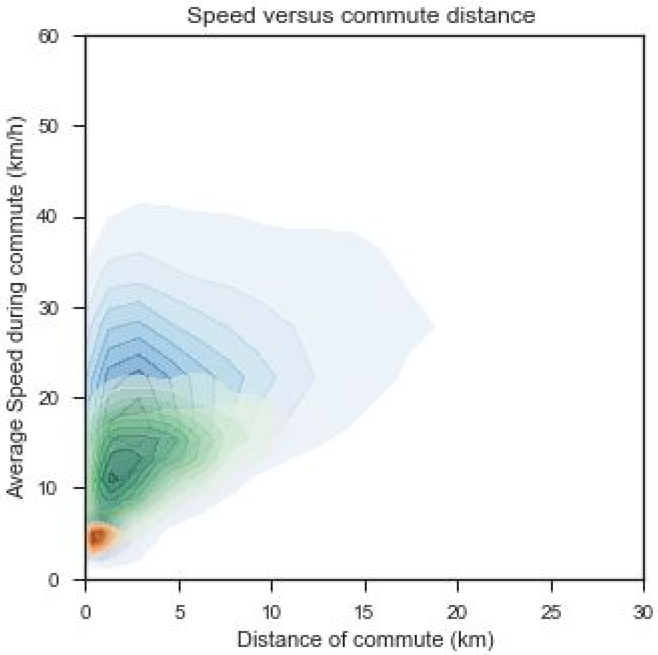
As with all of my analyses of this dataset, we begin by running the **load_mtlTrajet_data.py** script to load the data from json, into python dictionaries, and clean up some of the text. Importantly for this analysis, the data from the json **trip_final** is loaded into a dictionary of the same name, and the a list of the unique ID numbers of each recorded trajectory are stored as a separate vector **ids**.

Although I originally envisioned performing complex shape analyses of the individual commute trajectories to able train a model on high-dimensional data, for this article I consider only two straightforward measurements that are easily accessible in this dataset. They are chosen because intuitively we can expect them to differ between modes of transit: the average speed of the commute, and the distance across which commuters travel.

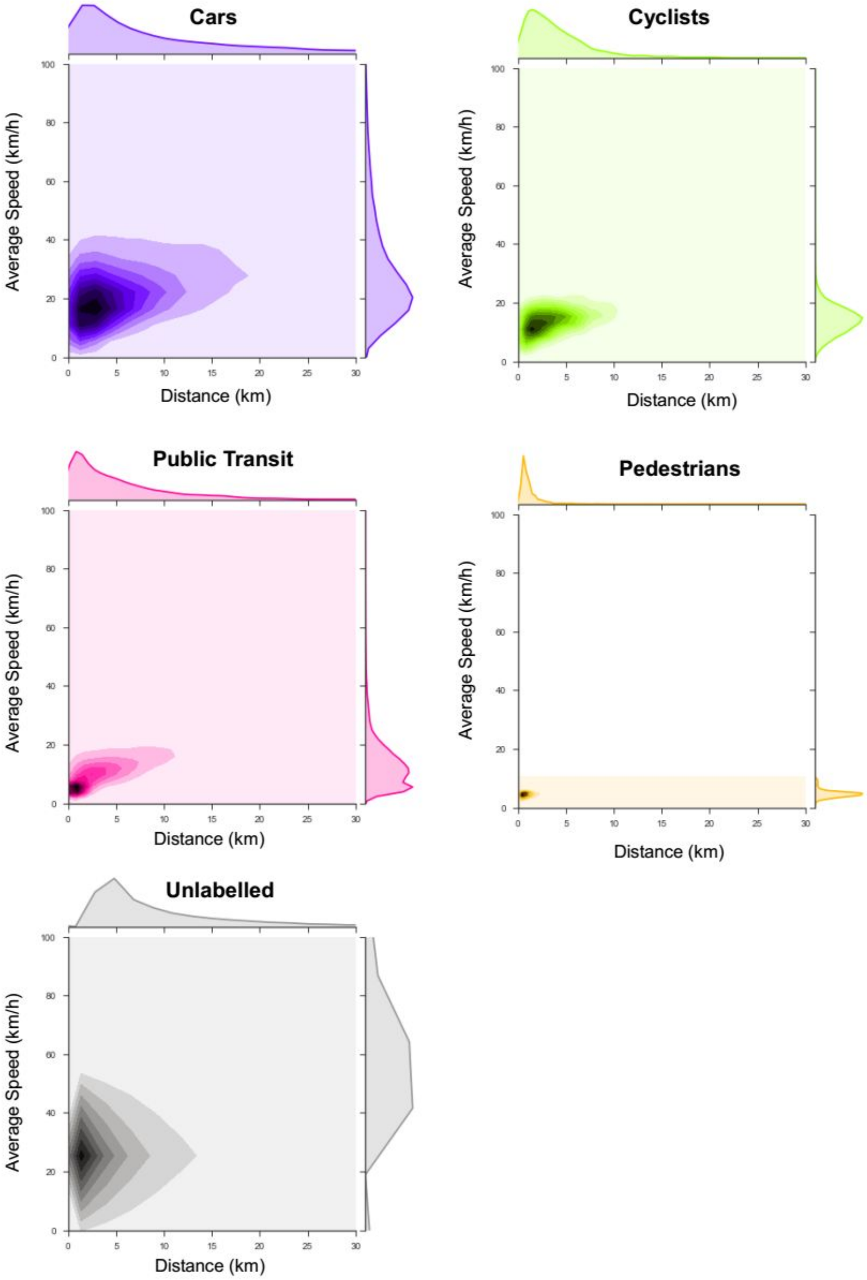
We begin by visualizing all of the labelled trajectories in a scatter plot with a color dependent on the mode of transit, in order to see what differences stand out. The x axis plots the Euclidean distance ('as the crow flies') while the y axis plots the average velocity.



Unsurprisingly, the pedestrians cluster together at the lowest speed clustered around 5 km/h - consistent with the average walking speed quoted on Wikipedia. The cyclists are notably faster with larger variance, and cars and public transit share a larger variational space with respect to their speed. We can also see that the further a distance someone has to travel, the more likely they are to use their car, possibly the most obvious discovery ever. I also show below an alternative view using Seaborn's Kernel Density Estimate (KDE) plots, because they nicely visualize how the distributions with respect to both speed and distance change with respect to mode of transit.

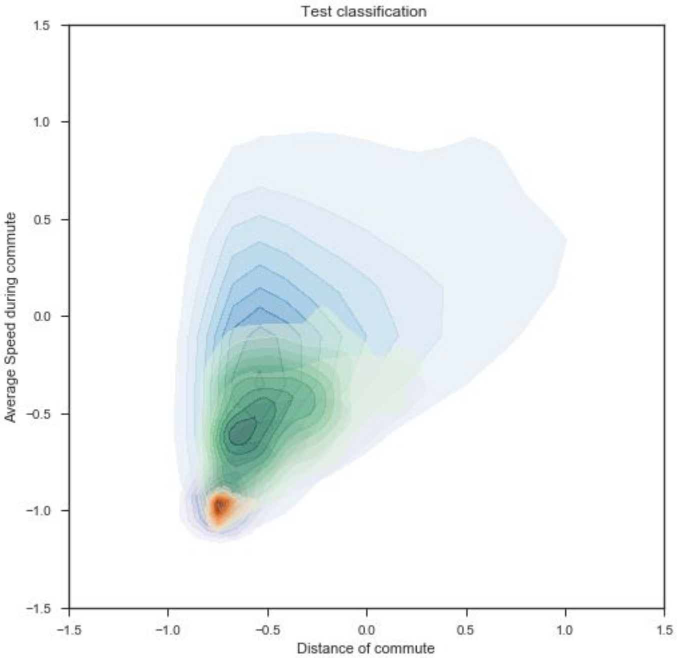


Because the overlap can make it difficult to see each of the contour plots individually, here is the same data on separate axes with the same scale. These are made with Seaborn’s jointplot function, including a density of the distribution for each variable individually in the margins. In the bottom-most axes, the 2D distribution of the unlabelled data is shown in black.

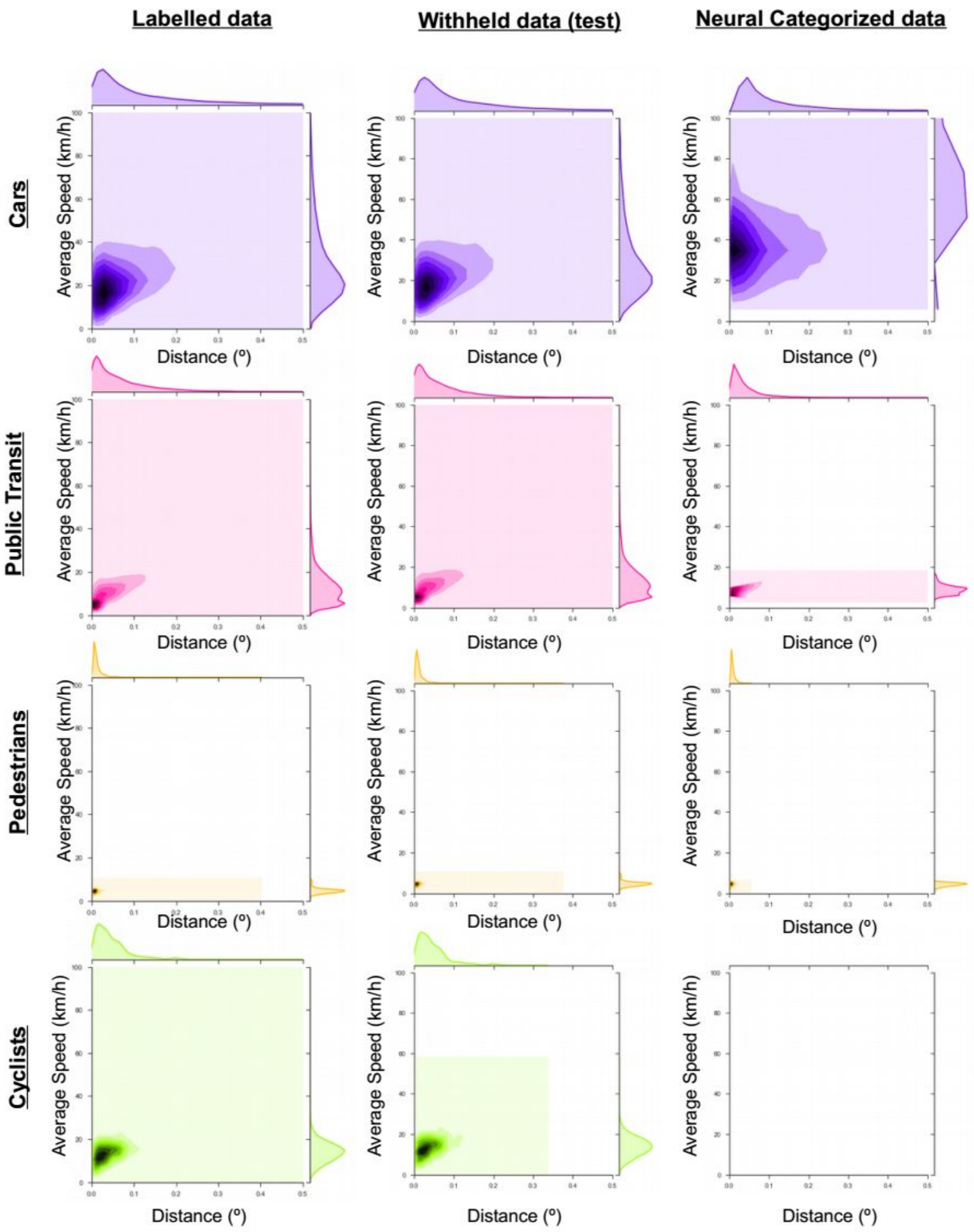


Using Scikit-Learn, I trained a neural network discriminator on the labelled trajectory data using the speed and distance. I chose the neural network because it was a general purpose classifier, and because it seemed to work quite well to fit the circular test set in the classifier comparison example of the SciKit-Learn documentation. 'Circular' seemed to be the most approprappeared most appropriate considering how the different modes of transit clustered together in the scatter plots. An interesting exercise would have been to systematically test each of the classifiers to see the resulting match on the test data, however this has been shown previously in the SciKit-Learn documentation: Classifier comparison**.** While this would have been the ideal way of optimizing the model selection, it has been covered in depth already, so I would offer little of novelty by repeating the same analysis here.

The labelled data set was split into training and testing subsets (40-to-60% respectively), and the result of the testing was a success rate of 0.68, which isn't great. At this point I would have ideally compared other models in a systematic fashion to maximize the success rate on the test set of data, which I don't have the computational resources for at this time. Instead, I accept that the data is modelled less than perfectly, considering that it is real data with lots of variance, and frequency distributions that are almost entirely overlapping one another. I did not expect a perfectly clean segmentation as seen in the example data, but nonetheless this is an area of the analysis which could be further optimized.



I applied the trained model to the data that lacked any labels, and obtained an array that lets me associate each trajectory ID from the list with the model's best guess of what mode of transit it made. To compare directly the influence of categorization on the distributions, I compared the groups using small multiples of the Kernel Density Estimate plots with marginal histograms from Seaborn.

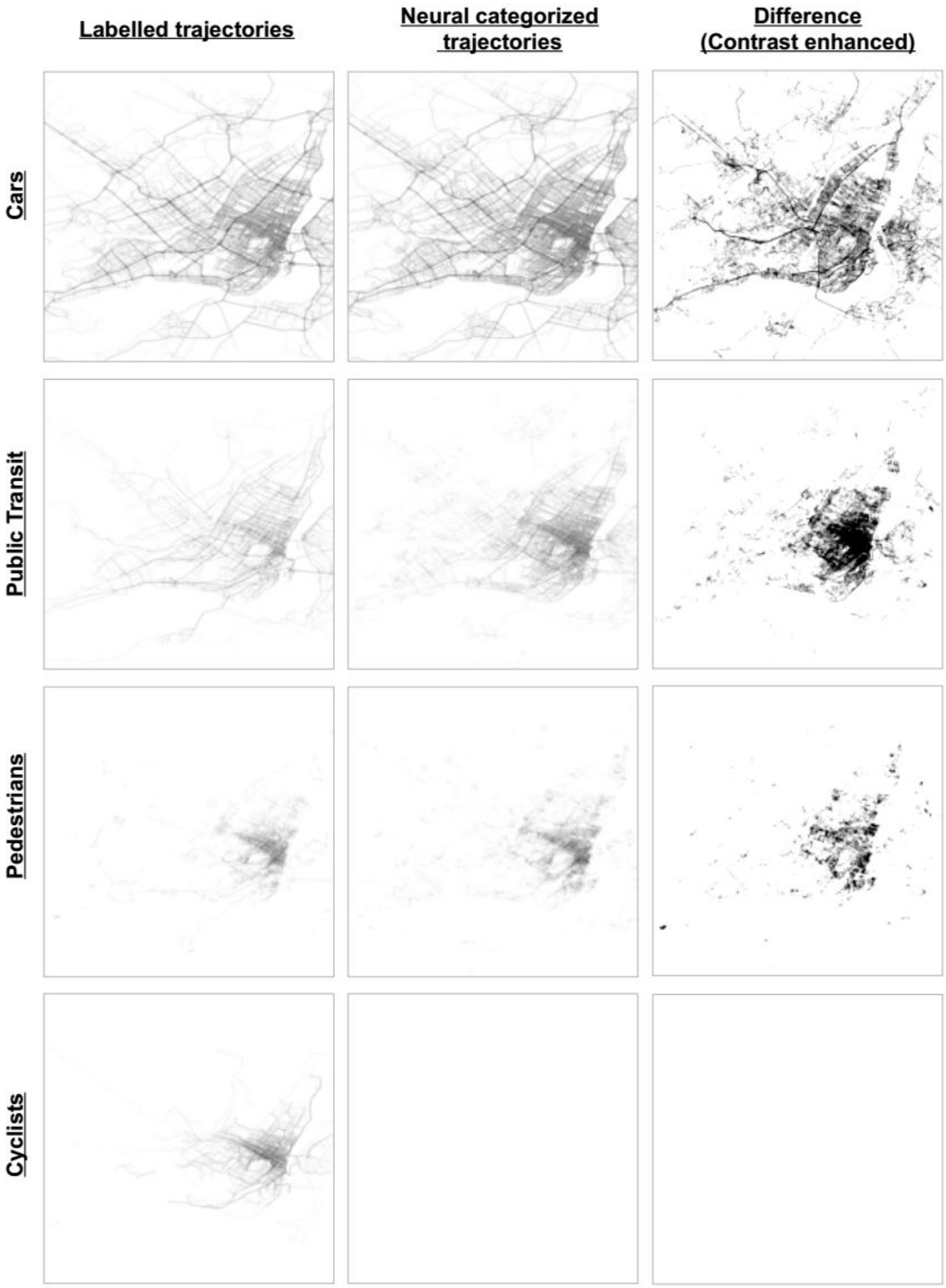


Note: unlike the previous figures, here the distance is in degrees (latitude / longitude), the original units in which I did the calculation, before converting to km) Unsurprisingly, the vast majority of the uncategorized data is associated with automobile traffic. However, the high proportion is somewhat strange, as the categorized data has ~60% of traffic classified as automobile traffic when considering the entire area of Montreal. Therefore it seems as though there is a tendency for automobile traffic to be less likely to classify their mode of transit, or more prone to errors the remove the labels or cause problems with the recording.

One thing that is particularly strange with the categorization is that none of the unlabelled trajectories were categorized as cyclists. It could be the fact that the recorded trajectories erroneously lost their labels when the person came to a complete stop, as is often the case in a car, bus or as a pedestrian - whereas cyclists are able to get from point A to B without every fully stopping (this is why some bikes don't even have brakes, leading to a recent crackdown). That's just one hypothesis, but it could alternatively be an issue with the similarity of the data between public transit and cyclists. If we look at their 2D distribution, we can see that the two groups occupy approximately the same space, which could have fooled the learning algorithm to think that all cyclists were in one of the other groups. If that were the case, then adding more variables could help to properly distinguish the cyclists from the other groups.

Next I wanted to find an independent method to validate the trustworthiness of the categorized data.
Exactly how we can validate how well the model classified the unlabelled trajectories it is not trivial, so we'll have to get creative to find a way of measuring this. To avoid any sort of circular logic, and potentially proving our own assumptions, ideally we would be able to draw a comparison between the labelled and categorized data that is independent of the variables used for categorizing the data. A good candidate is the spatial localization and density of the various modes of transit - with respect to the map of Montréal. Since the only metrics that went in to teaching the model were the average speed and distance, the learning model had no access to the geospatial location of the points. However, if we look at density plots of the labelled data showing the location of various modes of transit, we tend to see patterns specific to each mode of transit. For example, we tend to see more pedestrians in the downtown/plateau area, and the surrounding boroughs have a higher amount of bike traffic.

Therefore, comparing density plots for each mode of transit between labelled and categorized data sets should give us a good sense, visually, of how well the trajectories were matched. Such an analysis would also allow us to make more quantitative measurements (such as measures of overlap and center of mass), if need be.


Labelled trajectories
Neural categorized trajectories
Difference (Contrast enhanced)
Cars
Public Transit
Pedestrians
Cyclists

The geospatial distribution of the trajectories on the map tend to be fairly similar between the labelled and neural-categorized trajectories, which is encouraging; the trajectories for each mode of transit tend to pass through the same areas of town in roughly the same proportions.

I also wanted to see whether there were particular areas where there was a strong difference between labelled and categorized trajectories. For example, if passing through the tunnel

caused the app to stop recording and erroneously clip the label from a trajectory, then we’d expect a far higher density of points there in the categorized image versus the original labelled data.
Therefore, the intensity difference between the two images is shown in the third column. Note: the intensity has been enhanced consistently between each mode of transit, such that the patterns can be seen. It appears that rather than occurring systematically in one area, the area of higher density of unlabelled trajectories are pretty similar to the high density of original labelled data.