Vessel Prediction from Automatic Identification System (AIS) Data

Mason Leung, Wen Tao Lin

Introduction:

Our goal for this project was to implement machine learning algorithms to track various moving vessels using Automatic Identification System (AIS) data including a vessel's latitude, longitude, speed over ground, course over ground labeled with time stamps. Although we were given two data sets as training, we still opted to use unsupervised learning methods. The models are going to be tested using two metrics: one given K (the number of clusters is given) and one without being given K. The final models we settled on were two clustering methods: spectral clustering and density-based spatial clustering (DBSCAN).

Final Choice:

Spectral Clustering:	Value:	DBSCAN:	Value:
n_clusters:	Given K	eps:	0.30
assign_labels:	cluster_qr	min_samples:	8
n_jobs:	-1	n_jobs:	-1
gamma:	1		

Pre-processing:

We started preprocessing the data by attempting to transform the features into shapes that will separate different true clusters. One feature that we looked into was the course over ground, in the form of degrees. Because degrees have a range between [0, 3599], we noticed that there could be a potential problem where our unsupervised method may misinterpret a jump from 0 -> 3599 or 3599 -> 0 as two separate vessels. In order to fix this problem, we converted the "course over ground" into a sinusoidal function. That way, even when a vessel's position might change between 3599 and 0, the values do not actually change drastically.

Next, we tried different combinations of transforming/combining the features with no visible improvements to the ARI score of our algorithm. We tried combining longitude, latitude, and speed over ground all into one feature and scaling it. However, as shown below in **Figure 1**,

there was no clear distinction between each of the vessels using the course over ground and the combined features.

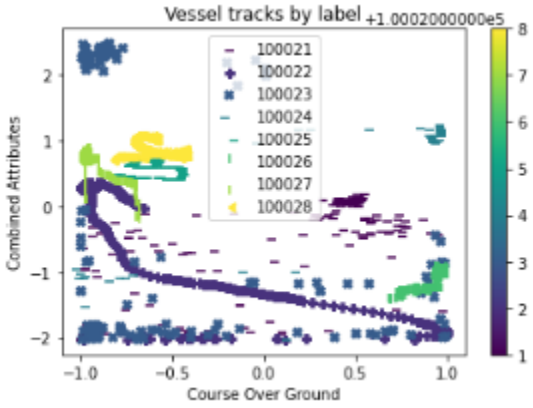

Figure 1: Combined Feature vs. Course Over Ground

We also attempted feature selection by removing both time and speed over ground one at a time. Again, this only proved to decrease the ARI score of our clustering methods. Lastly, we tried to standardize only some features, leaving other features unscaled. In the end, all these attempts to modify the features did not improve the reliability of our algorithm.

Classification Algorithm Selection:

When we first began thinking about the optimal machine learning model, the first question we asked ourselves was: unsupervised or supervised learning? Because we were given two extra data sets to train our model on, we thought that supervised learning could have potential. However, we came across a problem where the number of clusters differed in each dataset, so the standard supervised learning methods that we learned in class didn't seem to be applicable. We ran some brief experiments using K nearest neighbors and SVM, but both achieved low ARIs for the training set 1 and 2.

For the unsupervised algorithms, we experimented with 6 different clustering models: kmeans, spectral, hierarchical, birch, gaussian, and DBSCAN. We eliminated kmeans early on as the dataset did not fit the assumptions of the algorithm. Each cluster was not spherically blobbed and had roughly the same radius as the other clusters. Next, we tried agglomerative hierarchical clustering because the linkage feature seemed like a good way to test out different metrics for separating the vessels. Although the idea made sense, we realized that the dendrograms were far too complicated to be able to interpret the T value for a specified number of clusters. **Figure 2** is an example of one of the dendrograms for complete linkage:

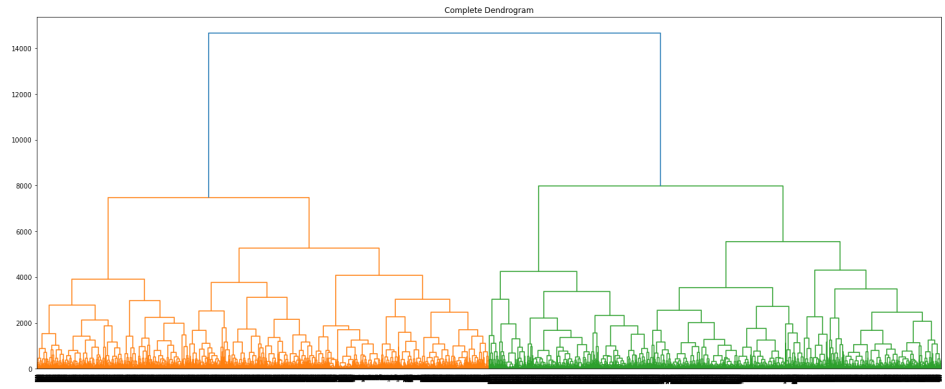

Figure 2: Hierarchical Clustering Dendrogram

We considered spectral clustering next because it does seemingly well with the idea of "connected graphs." If the two vessels travel along the same path at different times, they can be represented as two separate graphs because of the different time stamps. Vice versa, if two vessels are traveling along two different paths, then it is even clearer that the vessel tracks belong to two different vessels as a vessel cannot be in two places at once. Because the vessels are traveling along a path provided by the longitude and latitude, the position coordinates will continuously change in increments determined by the speed and course over ground meaning that the behavior of these two features can be similarly represented in terms of nodes of a graph. When we performed experiments on spectral clustering, it continuously outperformed both birch and gaussian clustering.

We also attempted clustering using DBSCAN because the advantage of this model is that it is optimal in separating clusters of high density compared to clusters of low density. Because each vessel has a different number of data points recording the position, it can also be interpreted that each "cluster" or vessel have vastly different densities of data points, meaning that DBSCAN would probably perform well as it is a density-based clustering algorithm. Furthermore, DBSCAN does optimally when the clusters have various different cluster shapes, which is an advantage that it has over the standard kmeans algorithm. Because the number of clusters does not need to be specified for DBSCAN, we opted to use this model for the metric measuring the ARI without being given K.

Spectral Clustering: Given K

For Spectral Clustering, the main parameters we experimented with were "affinity" and "assign_labels." When looking at affinity, we went with the default radial basis function (RBF) kernel because nearest_neighbors would not perform optimally when vessel paths overlap with each other at different times, whereas RBF's mapping technique will help differentiate those vessels. As for the "assign_labels" parameter, we were deciding between kmeans and cluster_qr methods. We felt that kmeans was not a good choice for other data shapes because the method

can be sensitive to initialization of centroid location, and we expect there to be some overlaps and crossovers in the vessel path. Moreover, we tried both methods for set 1 and set 2 because they could be representative of the data in set 3. Unsurprisingly, cluster_qr performed better than the nearest_neighbor for both sets, so we decided to select this method for assigning labels in the embedding space. **Figure 3** below shows sample vessels' tracks with randomly selected 15 clusters, as we do not know the true number of clusters in set 3.

DBSCAN: Withheld K

After selecting DBSCAN, we decided to test out the parameters "eps" and "min_samples". We began by considering all the reasonable "min_samples" values, where the rule of thumb is typically 2*numberOfFeatures or 0.1% of the sample size: 10 and 8 respectively for set 3. We ultimately chose min_sample = 8 because of its clustering result around our desired number of clusters (12 to 18). We then tried out different values for "eps" to find a cluster map that looks visually reasonable with its path of travel. We selected the value 0.3 because the number of resulting vessels was near our predicted number of vessels. The trajectory of the vessels also looked reasonable in **Figure 4** when being compared to the trajectory pattern of the vessels in set 1 and set 2.

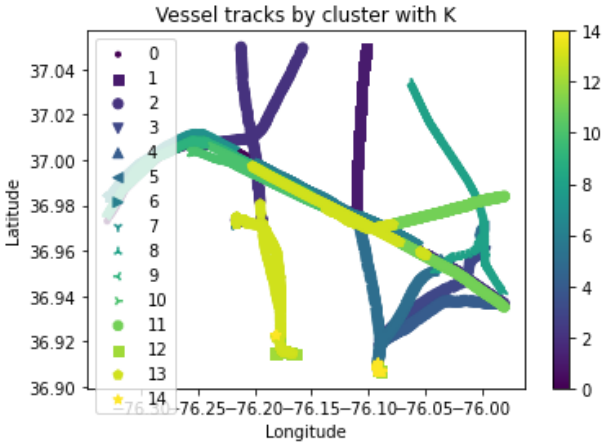

Figure 3: Spectral Clustering Vessel Tracks Set 3

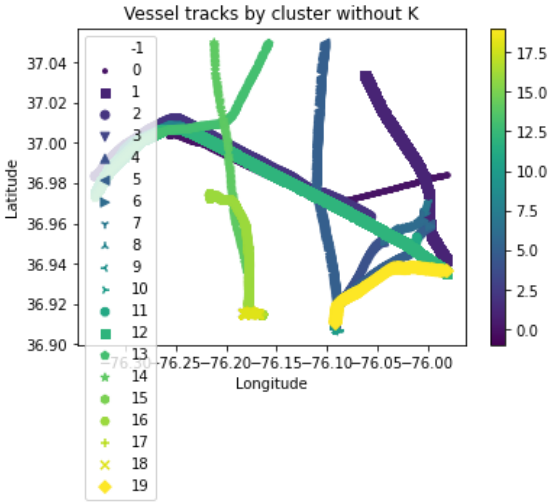

Figure 4: DBSCAN Vessel Tracks Set 3

Application:

To evaluate the algorithm's effectiveness, we first have to consider what are the consequences of "misclustering" the objects. For example, if there are a different number of vessels out in the same area but they are misclassified to be the same vessel, the local traffic might be assumed to

be light and therefore, we might recommend other vessels to routes in this area, thus further increasing the traffic. Therefore, we can assess the effectiveness of the algorithm by keeping track of the traffic report in the various areas over time. If the model is effective, then traffic should be greatly reduced because vessels can receive real-life updates about other vessels' path, therefore giving them time to plan ahead and take other potential routes instead of joining the traffic jam. On the other hand, if the model is not effective, meaning we falsely clustered vessels, then we would be sending false information to vessels out in the water; therefore, there is a higher chance for them to take the less optimal path. For example, a vessel could be informed that there is a heavy traffic ahead when in reality there is very minimal traffic, which will likely cause them to take an unnecessary detour. On the other hand, a vessel could also be informed that there is no traffic ahead when in reality, there is a traffic jam, which will delay their trip.

In the context of this algorithm working with the AIS vessel data, by being able to classify the vessels, we can combine our algorithm with an automated alert system that would alert the authorities if a certain type of vessel, such as a shipping vessel, is going off a predetermined course or is behaving abnormally. In addition, this algorithm will be able to identify real-time local sea traffic and notify other vessels in the area to prevent unnecessary interception of the path. In order to deal with practical issues such as gaps in data, we could use a hidden markov chain to compute the probability of the future positions of the vessels. From there, we can generate synthetic data of the vessel's positions during that gap to maximize the probability of the vessel having taken that path to the current location.

Work Cited:

Lutins, Evan. "DBSCAN: What Is It? When to Use It? How to Use It." *Medium*, 4 Dec. 2020, https://elutins.medium.com/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818.