Evaluating Unsupervised Clustering Accuracy

Introduction

One of the most consistent guidelines to succeed with a data science project is to tap into subject matter expertise. As a former Surface Warfare Officer in the US Navy, I spent over four years operating on ships and am familiar with ship navigation and operations. Additionally, I am always interested in Geospatial Information Systems (GIS) so I choose a project that would combine my love of all things maritime and GIS.

This project analyzes shipboard Automatic Identification System (AIS) activity as collected and shared by the US Coast Guard. AIS is an automatic tracking system that uses transponders on ships and is used by ships and port authorities to avoid collisions and manage ship traffic. Because of the tremendous size of the data, I have limited my analysis to a section of data from January 2017 that includes 500 million individual position reports from 20,731 unique ship identifiers, known as an Maritime Mobile Service Identity (MMSI). The January 2017 dataset is about 25 gigabytes of data.

The first step was to build an ingest pipeline for the data. The details of this pipeline are beyond the scope of this project, but are available on my GitHub with details about scraping the Coast Guard site and loading the data into a local PostGres data.

The large size of the data complicated most analytic approaches and required some novel approaches to parse. First, using PostGres’ GIS extension PostGIS, I reduced each individual ship’s track into one line segment to represent the trip. In the ‘ship\_trips\_analysis’ notebook located on my GitHub here, I analyzed the ships’ trips and additional metadata such as length of trip, number of days detected in the data, and number of position reports to select a smaller but representative sample from the data. This also helped to eliminate ships that never moved much, or seemed to have erroneous position reports from incredibly long distances.

This smaller sample size includes 200 individual ships and just over 2 million unique position reports. To help identify activity near known ports, I downloaded and ingested the World Port Index’s list of ports worldwide. Once this data was in the PostGres database, I used PostGIS’s ST\_Distance function to find the nearest port to each of the ship positions, and labeled each point with that port’s name if it was within 2km of that position. This data will be used later to validate the unsupervised clustering.

To actually cluster the positions in an attempt to find ports, I used Density-based spatial clustering of applications with noise, or DBSCAN. Originally I hoped to use Scikit-Learn’s implementation, but unfortunately because of the size of the data, DBSCAN clustering of the 2 million positions consistently crashes my Python kernel in Jupyter Notebook and Spyder. Instead, I used PostGres’ native DBSCAN algorithm which does not need to hold all of the data in memory. I used both my own local PostGres instance for validation and a PostGres cluster stood up in Amazon Web Services (AWS).