Case Study 2: Vessel Prediction from Automatic Identification System (AIS) Data

CSDS 340: Machine Learning for Big Data (Kevin S. Xu)
Case Western Reserve University

Assigned: Wednesday, November 23, 2022

Pre-submission deadline: Thursday, December 1, 2022, at 10:00am

Due: Thursday, December 8, 2022, at 10:00am

Objective

This case study considers the problem of tracking multiple moving targets over time. This problem is present in many application settings, including object tracking from video streams or radar data. In practice, the multiple target tracking problem requires first identifying objects from "raw" data such as pixel intensities and computing features from such data.

To simplify the problem to fit into a machine learning context, this case study considers automatic identification system (AIS)[1] data generated by maritime vessels. Each vessel has a maritime mobile service identity (MMSI) number that uniquely identifies each vessel. A vessel's AIS unit periodically sends a report on its position. The report contains the vessel's MMSI so that the vessel can be tracked over time.

In this case study, your job is to track the movements of the different vessels given position reports ***without the MMSIs***. In other words, *you must predict which vessel each report came from* given multiple reports from each vessel over time.

Data Description

Each row of the data corresponds to an observation of a single maritime vessel at a single point in time. A snippet of the complete (labeled) data is shown in Table 1.

Table 1. A short snippet of the complete (labeled) data.

Object ID	Vessel ID	Timestamp	Latitude	Longitude	Speed Over Ground	Course Over Ground
1	100008	14:00:00	36.90685	-76.089	1	1641
2	100015	14:00:00	36.95	-76.0268	11	2815
3	100016	14:00:00	36.90678	-76.0891	0	2632
4	100019	14:00:00	37.003	-76.2832	148	2460
5	100016	14:00:01	36.90678	-76.0891	0	2632
6	100005	14:00:01	36.90682	-76.0888	1	1740
7	100006	14:00:01	36.90689	-76.0893	0	1440

The column descriptions are as follows:

- Object ID (OBJECT_ID): A unique ID assigned to each observation (position report).

[1] https://www.navcen.uscg.gov/automatic-identification-system-overview

- Vessel ID (VID): Anonymized version of the vessel's maritime mobile service identity (MMSI) number, which uniquely identifies each vessel. The VID *uniquely identifies a vessel within a single data set* but ***are not consistent across data sets***! For example, the VID of 100001 in two different data sets, which represent two different time periods, may not refer to the same vessel.
- Timestamp (SEQUENCE_DTTM): Time of reporting in the form of hh:mm:ss in Coordinated Universal Time (UTC), where hh is hours, mm is minutes, and ss is seconds. The date information has been removed so that only time information is available.
- Latitude (LAT): Latitude of vessel position in decimal degrees.
- Longitude (LON): Longitude of vessel position in decimal degrees.
- Speed over ground (SPEED_OVER_GROUND): Vessel speed in tenths of a knot (nautical mile per hour), up to a saturation limit of 1022, which indicates that the speed is 102.2 knots or higher.
- Course over ground (COURSE_OVER_GROUND): Angle of vessel movement[2] in tenths of a degree, with a range between 0 and 3599 (359.9 degrees). The course over ground is still reported even when the speed over ground is 0 because vessels always have slight movements due to currents and other factors.

This case study will involve 3 data sets, each consisting of AIS data collected from the same area at the same time of day, but over 3 different days. Initially, you are provided with data set 1 with VIDs and data set 2 without VIDs. At the pre-submission deadline, you will have the opportunity to submit your algorithm for evaluation on data set 2. The VIDs for data set 2 will be released at this time along with data set 3 without VIDs. ***The final evaluation will take place on data set 3.*** See the Pre-submission (optional) section for additional details on this pre-submission.

Table 2. Timeline of data set releases.

Date	Data Set 1	Data Set 2	Data Set 3
Assigned date (11/23)	Released with VIDs	Released without VIDs	
After pre-submission (12/1)		Released with VIDs	Released without VIDs
After final submission (12/8)			Released with VIDs

Grading: 100 points total

Students should form groups of 2 students, who should both participate in the algorithm development and report writing.

Prediction accuracy: 50 points

You must submit two algorithms: one that predicts the VIDs given the number of vessels K and one that predicts the VIDs without being given K. Since you are provided with the evaluation data set without VIDs, you can do model selection either manually, e.g. by hard-coding the desired value of K, or algorithmically.

[2] The course over ground is similar to but not the same as the *heading*, which denotes the angle at which the vessel is pointing. The heading is not necessarily the angle at which the vessel is moving due to external factors such as wind.

Your algorithm will be evaluated for accuracy using the *adjusted Rand index* on the evaluation data set with VIDs. Since the adjusted Rand index is permutation-invariant, ***the actual values you predict for the VIDs are not important as long as each predicted VID denotes a single predicted vessel***.

The VIDs for the evaluation data set are not available to you, so this case study will require you to use your judgment to select a model that provides a good balance of accuracy and model complexity (to avoid overfitting). Accuracy on each metric is worth 25 points.

We will only run your algorithm once, so if your algorithm has any random component, e.g. for random initialization of centroids in K-means clustering, you may want to *set the seed* for the random number generator in your classifier to ensure that you are submitting your best algorithm. Within `scikit-learn`, you can do this by setting the `random_state` parameter to a fixed integer seed value.

Your grade for accuracy on each metric will be determined by the accuracy of your algorithm *compared to the rest of the class*, i.e. the most accurate algorithm receives 25 points, the next most accurate algorithm receives 24 points, etc. If two algorithms are extremely close in accuracy, they may be assigned the same grade. Also, if there is a large gap in accuracy between two groups' algorithms, there may be also be a gap in the grades (e.g. no group assigned a grade of 23 points because there is a large gap between the model that received 24 points and the model that received 22 points).

Your algorithm will also be compared to a simple baseline that is not expected to be a competitive solution. In this case study, the baseline will be K-means clustering fit to the data after standardizing the features to have zero mean and unit variance. For the case of unknown K, the baseline arbitrarily chooses $K = 20$, the number of vessels in data set 1. The accuracy of the baseline model will be considered as 0 points, so the accuracy of your algorithm ***must exceed that of the baseline*** in order to receive a grade for accuracy! See the `predictVessel.py` file posted on Canvas with the assignment for code implementing the baseline.

Note: If your code generates an error and does not run correctly, then your grade for accuracy will be a 0. This may occur due to your code not following the given specification (see Source code specifications section) or due to faulty assumptions that result in a non-functional algorithm or output that does not meet the specifications. ***We will not make any attempts to modify your code to correct it!*** You will have the opportunity to pre-submit your code for evaluation about one week prior to the deadline to try to catch such errors (see Pre-submission (optional) section).

Report: 50 points

Submit a report of not more than 5 pages that includes descriptions of

- Your final choice of algorithm, including the values of important parameters required by the algorithm. ***Explain whether your algorithm is supervised or unsupervised and how it uses training data*** (if at all). Communicate your approach in enough detail for someone else to be able to implement and deploy your vessel tracking system. (5 points)
- Any pre-processing you performed and your model selection criteria, e.g. how you choose the number of vessels for the unknown K setting. Describe how these choices affected your results. (10 points)
- How you selected and tested your algorithm and what other algorithms you compared against. *Explain why you chose an algorithm and justify your decision!* It is certainly OK

to blindly test many algorithms, but you will likely find it a better use of your time to be selective based on the specifics of this data set and application. (10 points)

- Recommendations on how to evaluate the effectiveness of your algorithm if it were to be deployed as a multiple target tracking system. What might be a good choice of metric, and what are the implications on your algorithm? How might you deal with practical issues such as gaps in data, e.g. no data available from all vessels for 10 minutes? (10 points)

Your report will be evaluated on both technical aspects (35 points, distributed as shown above) and presentation quality (10 points). Think of your submitted model as a representation of your "final answer" for a problem, while your report contains your "rough work." If you tried several approaches that failed before arriving at a model that you are satisfied with, describe the failed approaches in your report (including results) and what you learned from the failed approaches to guide you to your final model.

You may include additional descriptions, figures, tables, etc. in an appendix beyond the 5-page limit, but content beyond page 5 may not necessarily be considered when grading the report.

Submission details

Although you are expected to work in groups of 2 students, each student is required to submit their case study individually on Canvas. The contents of the submissions for all group members should be the same.

For this case study, submit the following to Canvas:

- A file named `predictVessel.py` containing your source code.
- A Word document or PDF file containing your report.

Do not place contents into a ZIP file or other type of archive!

Source code specifications

Your source code must contain the following function with *no modifications to the function header*:

```
def predictWithK(testFeatures, numVessels, trainFeatures=None,
                 trainLabels=None):
```

This function predicts the VIDs for the vessels given the test data features (all columns from Table 1 except for object and vessel ID) and the number of vessels. The function should output an integer array of predicted VIDs with the ith entry corresponding to the ith row of `testFeatures`. Since the evaluation metric, the adjusted Rand index, is permutation-invariant, you may represent the vessels using any unique set of integers, e.g. `1` to `numVessels`.

Note that your predictor must output at most `numVessels` unique VIDs in order to meet the specification![3] ***If your array of predicted VIDs contains more than*** `numVessels` ***unique VIDs, then your adjusted Rand index will be set to*** $-\infty$***, and you will receive a 0 accuracy score for this portion!***

[3] The reason that we specify at most rather than exactly `numVessels` VIDs is because many clustering algorithms may return less than K clusters even when the value of K is specified. For example, in K-means clustering, sometimes a centroid may not end up being the closest to any example after an iteration, which usually results in that centroid being dropped so that the result has $K - 1$ clusters.

For the final evaluation, `testFeatures` will correspond to features from data set 3, while `trainFeatures` and `trainLabels` will correspond to features and labels, respectively, from data sets 1 and 2 concatenated. Since the same VIDs in data sets 1 and 2 may correspond to different vessels, we will add 200,000 to the VIDs in data set 2 to avoid any overlap in VIDs. It is not required to use training data (hence why they default to `None`), but we have included arguments for them in the function header if you do choose to use them in your algorithm.

```
def predictWithoutK(testFeatures, trainFeatures=None,
                    trainLabels=None):
```

This function predicts the VIDs for the vessels given only the test data features ***without*** the number of vessels. You may use the same algorithm as in `predictWithK()` or a different algorithm altogether. There are no constraints here on the number of unique VIDs for this function.

If you have any other code in your `predictVessel.py` file, please place it under the following `if` block:

```
if __name__ == "__main__":
```

This prevents the code from running when we import the `predictWithK()` and `predictWithoutK()` functions from your `predictVessel.py` file. For an example, see the structure in the `predictVessel.py` file on Canvas with the assignment. We will use the script `evaluatePredictor.py` (also posted on Canvas) to evaluate the accuracy of your algorithm.

I have also provided a third file `utils.py` that contains functions for loading data and plotting vessel tracks.

Pre-submission (optional)

Groups may optionally participate in a pre-submission that mimics the actual grading process for classification accuracy. *This is purely for informational purposes and will not count towards your final grade!* If your group elects to participate, you should submit their code *on Canvas the same way you would submit your actual submission*, being sure to ***follow the source code specifications***!

We will run your algorithm on data set 2 with VIDs and report the accuracy metrics for each group on a leaderboard visible to all groups. Groups may use this leaderboard to gauge their standing within the class. *The leaderboard will be anonymous* (no group or student names), but each group who submitted their code will be informed of the accuracy of their submission. The VIDs used for evaluating this pre-submission will be made available to all groups following the evaluation, regardless of whether they participated in the pre-submission.

Note: the data used for evaluating the pre-submission (data set 2) is ***different*** from the data used for the final accuracy evaluation (data set 3). However, the final accuracy evaluation will be conducted using the same process as for the pre-submission, so groups may elect to use this as a “test run.” In particular, it allows groups to test if their code has any issues that prevent it from working properly prior to the final evaluation.