Tracking Hurricanes with Gradient Boosted Regressors And Multi-Output Regressors

Grant Hunter
CIS377 - Final Project
grant.hunter@students.fhu.edu

Abstract

This paper explores the benefits and challenges of using simplistic models for predicting hurricane trajectories. To accomplish this, I used two models: one that predicted the direction of the hurricane and one that predicted the distance to be traveled. Putting both models together allowed me to compute a trajectory path to reasonably low error.

1. Introduction/Background/Motivation

1.1. Hurricane Tracking

Hurricanes are becoming increasingly catastrophic due to global climate changes. Tracking these disastrous hurricanes becomes dire. Doing so in a computationally time-effective matter is equally necessary since people must act fast to prepare for the worst.

People cannot always rely on a centralized agency for weather forecasts, so it is valuable to have open-source models that people with basic setups can utilize.

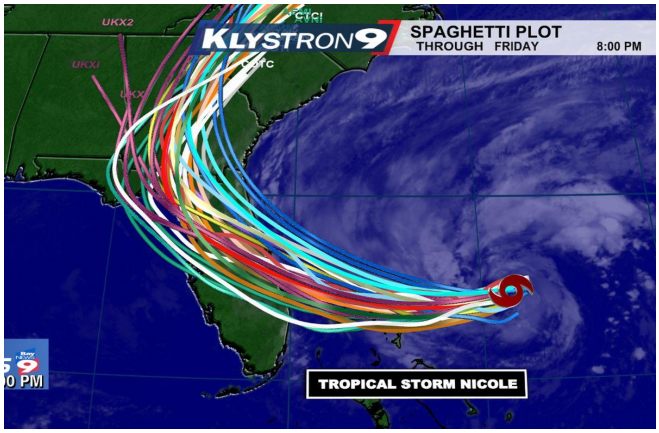
Another important characteristic to focus on is that since hurricanes are quite large (~300 miles wide), so minor imprecisions can be acceptable.

1.2. Commonly Used Approaches

The most prominent models for predicting hurricanes involve a few kinds of structures: GANs, LSTMs, and Ensembles. These can be relatively computationally expensive, so I sought to achieve the best error possible with more simple time-effective regressors.

1.3. Possible Applications

I believe that my model could be used as an additional model to be averaged into ensemble methods computed by the National Hurricane Center (NHC), or at least alongside the many other models of a spaghetti plot. An integral part of hurricane tracking is the existence of many unique models that can interact together to give us a greater sense of possibilities.



While predicting hurricane trajectories has become manageable, they are still chaotic systems with enough randomness to demand adjusting for uncertainty.

1.4. Data

The dataset used for this project was HURDAT2 retrieved from Kaggle, which NOAA shared. The data contains 1814 distinct hurricanes from the years 1851-2015.

The HURDAT2 dataset contains a total of 22 features on 49,105 instances. I only utilized 7 of those original features within my models. I also computed features discussed in section 2 of the paper.

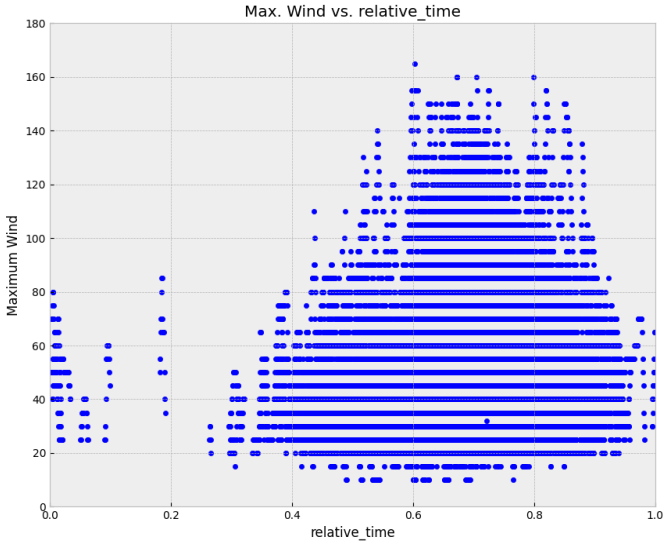
2. Approach

2.1. Determining Relevant Features

An important first step in regression problems is determining the important features for informing desired output. Obvious candidates were the *Latitude* and *Longitude* features, which were used in both the independent variable(s) and dependent variable(s).

To simplify the regression task, *Latitude* and *Longitude* values were converted to an XY-coordinate

plane using a geographic transformation. The location of the storm on the globe can contain information about typical behaviors of storms at a given coordinate. To best estimate the trajectory of storms, I used the vectors between points as a calculated field, which led to building two models: predicting the direction and length of the vector.



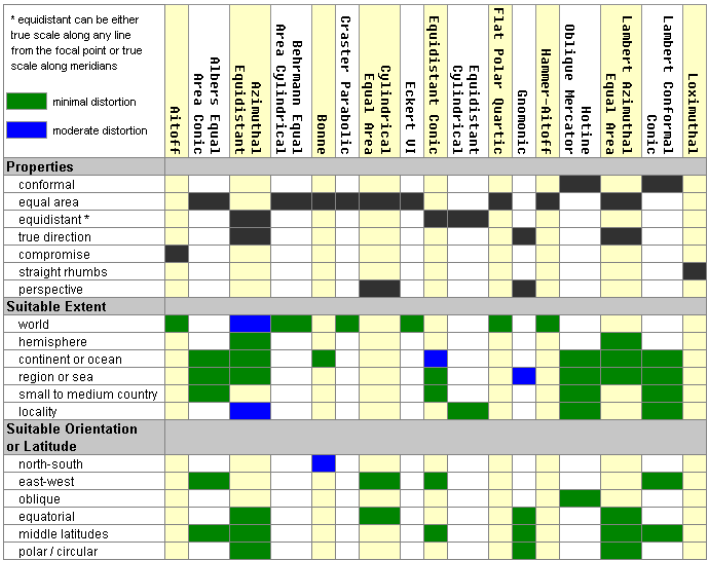
In examining the *Maximum Wind* feature against *Relative Time* (decimal value in the range (0,1) representing completion of the year), it became apparent that there was indeed a “hurricane season” as is colloquially accepted. I decided to include discretized DateTime features in the regression models.

Maximum Wind was another feature that was used in the regression models, as I believe it contains information on the intensity of the storm, which could inform how far the storm may travel.

The final features used in the regression models were lagged copies of the *Direction* and *Lengths* of the vectors. This will be further discussed in section 2.3.

2.2. Geographic Features and Map Transformations

Meteorologists take geographic transformations of coordinates as an important metric to have into consideration. Every transformation from a spherical globe onto an XY-coordinate plane will have distortions. The goal in choosing a transformation is to pick what distortions you want to allow.



Some are better at preserving direction, and some are good at preserving shape. It is not possible to be ideal at both. However, there do exist transformations that are good enough over a specific-sized region. Since the dataset contained points from nearly the entire Northern Atlantic, I decided to go with the Equidistant Sphere transformation, which was good up to hemisphere-sized regions and preserved both direction and shape sufficiently well.

```
Name: World Equidistant Cylindrical (Sphere)
Axis Info [cartesian]:
- E[east]: Easting (metre)
- N[north]: Northing (metre)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Coordinate Operation:
- name: World Equidistant Cylindrical (Sphere)
- method: Equidistant Cylindrical (Spherical)
Datum: Not specified (based on GRS 1980 Authalic Sphere)
- Ellipsoid: GRS 1980 Authalic Sphere
- Prime Meridian: Greenwich
```

It takes the shortest distance between two points on the plane to be equivalent to the shortest distance on a sphere: a great circle. This important characteristic would help inform a more realistic path for hurricanes, as they do not take what a Mercator map projection would show to be the shortest distance.

2.3. Converting from Time-Series Data to Supervised

An important step in taking time-series data and preparing it for regression was the addition of lagged vectors and multi-output predictions. Lagged vectors helped to communicate to the model some level of change over time in the vectors, which would help better classify what was to come.

Multi-output predictions are highly valuable in hurricane prediction because having an online model that can only predict one step (6 hours in this case) would be unhelpful for those who shelter or evacuate. I experimented with a 4x4 grid of input and output sizes for my models (up to 24-hour prediction window).

2.3.1. Chained Regressors

Most regression models do not natively adapt to predicting multiple outputs (except Decision Trees). It became necessary to use *RegressorChain* from *sklearn*'s *multioutput* sub-module to wrap around the Gradient Boosted Regressors. The wrapper tells the models to predict one output at a time, much like an iterator.

2.4. Error Function

I used the MSE error function for the predicted direction and length of a given vector versus the ground truth best track vectors provided in the dataset. MSE provides a good error function because it punishes large deviations from the truth.

I considered using the MAPE error function for a quicker interpretation of error as a percentage difference from the ground truth. I saw both used in different code examples for both geographic data problems and time-series problems.

2.5. Anticipated Problems

One of my greatest problems was figuring out how effectively transform the data. I wanted to experiment with the benefits of using a transformation other than the simple Mercator projection, which was used Kaggle notebook I adapted. Many geographic transformations exist, but they often do not meet the needs of accurately preserving both shape and direction over large areas. I decided to compromise on the transformation that I believed would benefit me most, as discussed in section 2.2.

Another challenge I anticipated was interpreting errors. The error values returned by my models were in terms of radians and distance under a transformed coordinate system. Converting those to miles was something I found to be rather challenging.

2.6. Encountered Problems

I discovered that the *RegressorChain* seemingly would not work with a shuffled dataset. I therefore had to abandon *sklearn*'s *train_test_split* function and opt for a less preferable split of training and test sets. I split the sets into the standard 80-20 ratio, but from the beginning. This may have introduced biases in my models and may have also led to poor generalization.

I also found trouble in accurately demonstrating the models with an output greater than one graphically. I thought this would be neat to see since it would likely present something akin to the spaghetti model shown in Figure 1.

3. Experiments and Results

3.1. Grid of Input and Output Sizes

Iterating over the 4x4 grid of input and output sizes helped me better understand how more input information did not necessarily mean better output errors. I was impressed to find how close in scale the errors were even over varying input and output windows.

Top 10 Direction Models

	i	j	model	error
16	2	1	dt_direction_seq	7.990577e-08
32	3	1	dt_direction_seq	1.318795e-06
48	4	1	dt_direction_seq	3.068838e-06
49	4	1	chained_direction_gbr	1.892545e-04
53	4	2	chained_direction_gbr	1.932355e-04
33	3	1	chained_direction_gbr	2.114391e-04
17	2	1	chained_direction_gbr	4.270107e-04
52	4	2	dt_direction_seq	5.977801e-04
37	3	2	chained_direction_gbr	5.392164e-02
56	4	3	dt_direction_seq	9.621630e-02

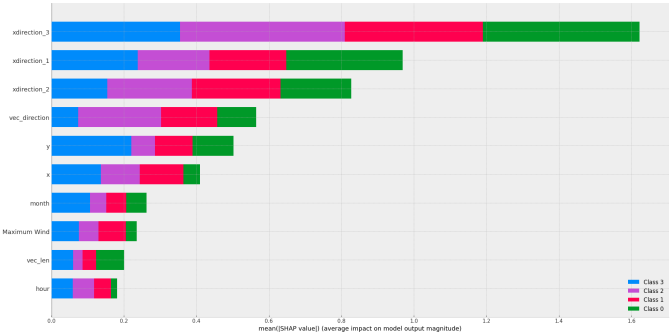
There are clear trends in the errors. The models that predicted a single output performed significantly better than those that predicted more. Also, the Decision Trees

outperformed the Gradient Boosted Regressors almost always. This is likely an overfitting issue.

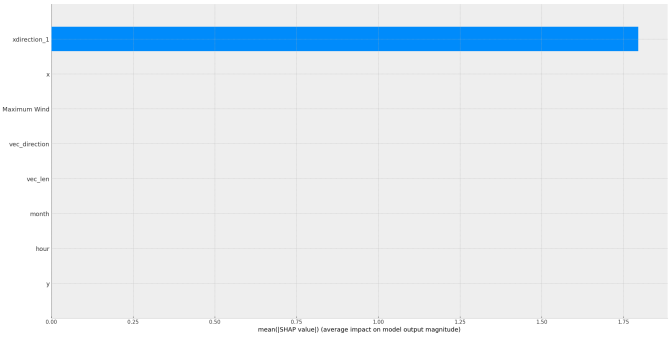
Top 10 Length Models

	i	j	model	error
18	2	1	dt_len_seq	1.549201e-13
50	4	1	dt_len_seq	6.876965e-13
34	3	1	dt_len_seq	3.279842e-12
19	2	1	chained_len_gbr	2.135574e-08
35	3	1	chained_len_gbr	3.310921e-08
51	4	1	chained_len_gbr	3.979585e-08
55	4	2	chained_len_gbr	4.019998e-08
54	4	2	dt_len_seq	1.273266e-07
39	3	2	chained_len_gbr	1.746868e-05
3	1	1	chained_len_gbr	1.898897e-05

After evaluating the models, I used the *shap* module to display the average feature importance for both a model of size input=4, output=4 and a model of input=2, output=1. This revealed that the lagged vectors were highly impactful on the models.



The input=4, output=4 model showcased that the most recent vectors provided the greatest impact, in order. The input=2, output=1 model, which had the best error out of all the models tested, communicated that the most recent vector was by far the most valuable feature in the regression, with all other features very far behind.



3.2. Decision Tree Regressors vs. Gradient Boosted Regressors

I implemented Decision Trees primarily to help directly compare my errors to the Kaggle notebook linked in my sources. I used Gradient Boosted Regressors to provide a more balanced and optimized regressor that was less prone to overfitting. While the Decision Trees performed better, the Gradient Boosted Regressors provided more trustworthy outputs.

4. Future Work for Improvement

I think a valuable edit to make would be computing distances so that the errors are more easily interpretable. My current errors show some overfitting, but not in a way that it is obvious what changes need to be made to the models since the outputs seem to have much more variation than the project on the map.

I also believe that it would be valuable to interpolate between data points to finetune my model down to smaller chunks of time, as is common with most SoTA models. Taking smaller chunks of time would possibly allow the time and location features to play a bigger role. I also think that interpolation would give more data which could help improve the models' ability to predict larger windows.

5. Sources

http://www.physics.drexel.edu/~goldberg/projections/goldberg_gott.pdf

https://learningweather.psu.edu/node/59

http://www.geo.hunter.cuny.edu/~jochen/gtech201/lectures/lec6concepts/map%20coordinate%20systems/how%20to%20choose%20a%20projection.htm

https://faculty.cs.niu.edu/~dakoop/cs680-2020sp/assignment2.html

https://michaelminn.net/tutorials/r-hurricanes/index.html

https://learningweather.psu.edu/node/60

https://moez-62905.medium.com/multiple-time-series-forecasting-in-python-fab0a8a1afdb

https://towardsdatascience.com/how-to-predict-a-time-series-part-1-6d7eb182b540

https://towardsdatascience.com/the-complete-guide-to-time-series-forecasting-using-sklearn-pandas-and-numpy-7694c90e45c1

https://towardsdatascience.com/simple-guide-on-using-supervised-learning-model-to-forecast-for-time-series-data-a570720add84

https://unidata.github.io/python-gallery/examples/HurricaneTracker.html

https://github.com/cqwangding/OMuLeT

https://developers.arcgis.com/python/samples/part3-analyze-hurricane-tracks/

https://developers.arcgis.com/python/samples/forecasting-pm2.5-using-big-data-analysis/

https://developers.arcgis.com/python/samples/creating-hurricane-tracks-using-geoanalytics/

https://web.uwm.edu/hurricane-models/models/verif.html

https://s3.us-east-1.amazonaws.com/climate-change-ai/papers/icml2021/38/paper.pdf

https://www.osti.gov/servlets/purl/1491971

https://arxiv.org/pdf/1802.02548.pdf