Ahmed Baraka, Nathan Lindsay, Kevin Trejo, Michael Smith

Project Stage 4 Report

Introduction:

We will explore the use of three regression algorithms to estimate the encountered wind speed of a drone based off of the onboard IMU data. This report will discuss the different aspects of the project regarding motivation, problem definition, code implementation and the results we have gathered.

Motivation:

The motivation of this project follows Ahmed Baraka's research thesis on wind estimation using unmanned aerial vehicles (UAV), also known as drones. As part of the research project, our first step was to replicate and implement the wind estimation technique found in "A K Nearest Neighborhood-Based Wind Estimation for Rotary-Wing VTOL UAVs" by Liyang Wang and Guarav Mishrav. This paper employs K-Nearest Neighbors (KNN) regression to estimate the wind speed given UAV Inertial Measurement Unit (IMU) sensor data. One of the main concerns for vertical take-off and landing vehicles (VTOL) is wind estimation. Typically sensors are very inaccurate, especially at low speeds, and are affected severely by the rotor down-wash effect. This lack of accuracy can dramatically impact the performance of the UAV, leading to unpredictable flight behaviors and instability. In addition, the utilization of add-on sensors increases the weight of the UAV and decreases flight time. The proposed wind estimation technique eliminates the need for an additional sensor and provides the necessary information to the UAV for stable flight in windy environments.

Problem Definition:

The problem we sought to study is the effectiveness of the K-Nearest Neighbors (KNN) regression algorithm at predicting the encountered wind speed of a drone using only onboard IMU data. To further justify the use of the KNN regression algorithm in this application we compare its results to other regression algorithms that are suitable for nonlinear datasets. We chose to compare our KNN results with a Decision Tree Regression algorithm and a Support Vector Machine Regression algorithm.

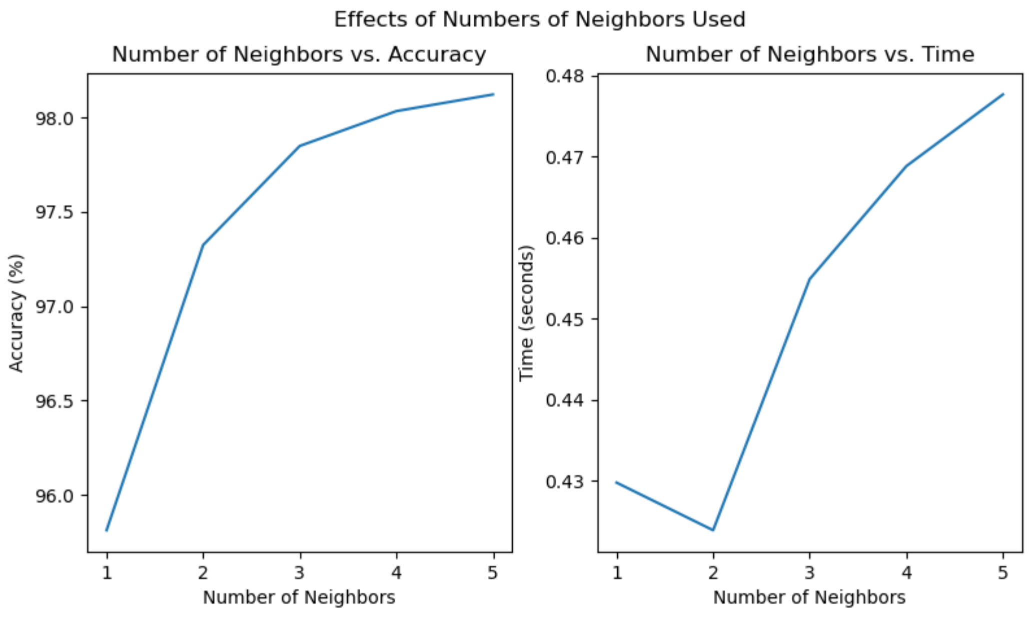
Solution:

Generating Datasets: We used a flight simulation in Matlab to collect 30 data points per second for 30 seconds (900 data points in total). This is similar to what was done In Wang's implementation that used 19 features and built the dataset from the average and the variance of the error between the desired value and the actual value for the UAVs physical states (positional, rotational) and control inputs. We chose to not utilize the error in the position and angle, but to use the normalized position and angle values as they are to train and test out datasets.

Training and Testing: Wang's algorithm was trained using software and tested using real hardware. We chose to do both our training and testing in Python to gather preliminary results before we physically implement these algorithms. For our training and testing we used the first 6 physical states (x, y, z,roll, pitch, yaw) of the UAV as our features and the corresponding wind speed for our target variable. We ran the data we generated through our KNN algorithm where the number of agents was incrementally increased beginning at 1 and ending at 5 neighbors. The Decision Tree regression algorithm ran in a similar manner where the maximum depth was set beginning at 1 and ending at 20. For the Support Vector Machine regression algorithm, the kernel type was set to RBF.

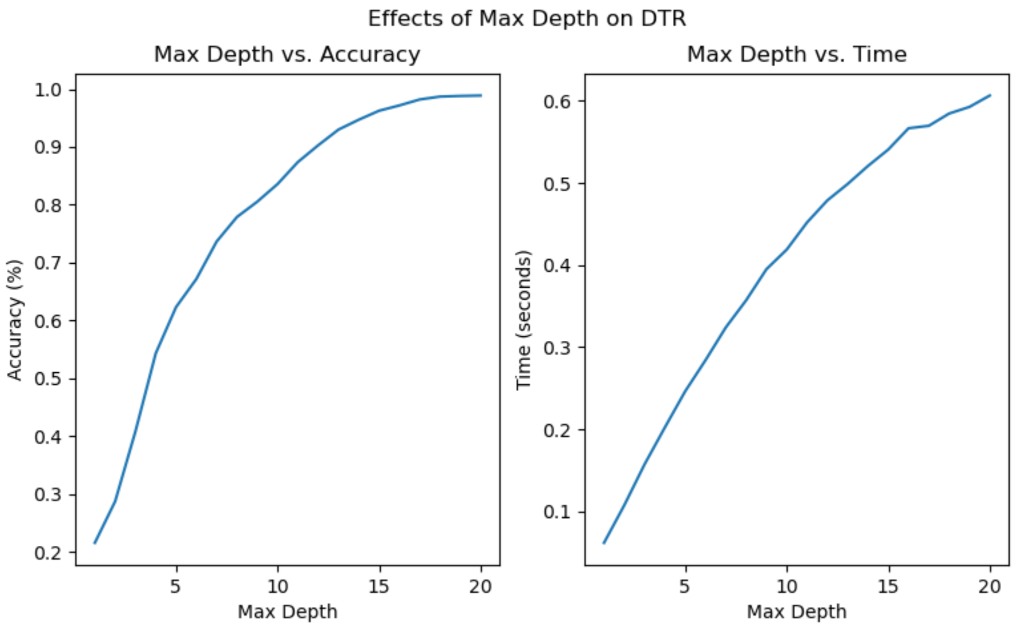
<u>Results</u>:

The below plots show how the number of neighbors affected the accuracy of the prediction and the run time of the KNN algorithm. It is seen in this plot that increasing the number of neighbors used increases the accuracy up until a certain point at which it begins to level off. We also see an increase in the time taken to train and test the algorithm, however this increase is very small showing that this algorithm is robust to the change in the number of neighbors used. It should also be noted that a fairly high accuracy of 95% is achieved while only using one neighbor which leads us to believe that the data is clustered together fairly close.

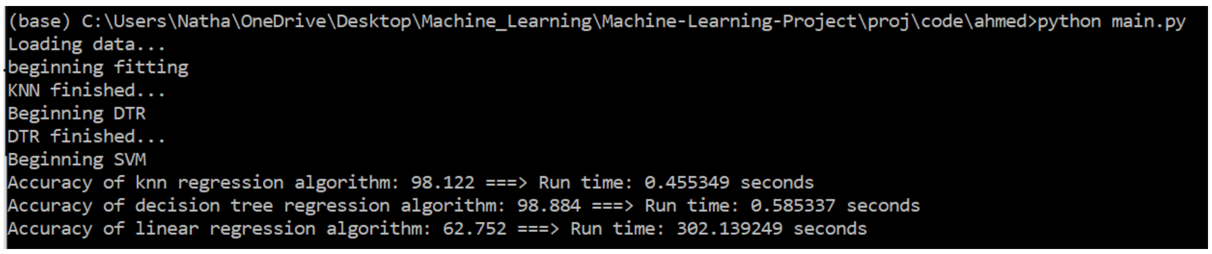


Ahmed Baraka, Nathan Lindsay, Kevin Trejo, Michael Smith

The below plots show the performance of the decision tree regression algorithm as the maximum depth of the decision tree is increased from 1 to 20. The accuracy has a similar trend to the KNN regression algorithm, where as the maximum depth of the decision tree increases, the accuracy and time taken for training and testing also increase. The decision tree regression algorithm is very sensitive to the maximum depth number and there is a larger rise in accuracy as the depth number is increased as well as a larger rise in the training and testing time than the KNN.



Below are all the accuracies and runtimes of all three algorithms. The KNN regression and decision tree regression algorithms are very close in performance with accuracies of 98.1% and 98.9% respectively. These two algorithms would be the most suitable algorithms for this application. The support vector machine regression algorithm performed the worst, taking over 300 seconds, or about 5 minutes, to train and only provided an accuracy of 62.7%, making this the least desirable algorithm to be used in this application. We found the RBF kernel type provided the best results for the support vector machine.



```
(base) C:\Users\Natha\OneDrive\Desktop\Machine_Learning\Machine-Learning-Project\proj\code\ahmed>python main.py
Loading data...
beginning fitting
KNN finished...
Beginning DTR
DTR finished...
Beginning SVM
Accuracy of knn regression algorithm: 98.122 ===> Run time: 0.455349 seconds
Accuracy of decision tree regression algorithm: 98.884 ===> Run time: 0.585337 seconds
Accuracy of linear regression algorithm: 62.752 ===> Run time: 302.139249 seconds
```

Ahmed Baraka, Nathan Lindsay, Kevin Trejo, Michael Smith

Reference

Liyang Wang, Guarav Misrav, Xiaoli Bai. (March 2019). A K Nearest Neighborhood-Based Wind Estimation for Rotary-Wing VTOL UAVs. *drones.* Retrieved from https://www.researchgate.net/publication/332175471_A_K_Nearest_Neighborhood-Based_Wind_Estimation_for_Rotary-Wing_VTOL_UAVs