Adaptive Model For Location Detection

By: Pranav Chopde, Jordan Uzan Supervisor: Omer Tzidki

Project Number:**18-1-1-1583**

Project carried out at Tel Aviv University

Iby and Aladar Fleischman Faculty of Engineering Tel Aviv University

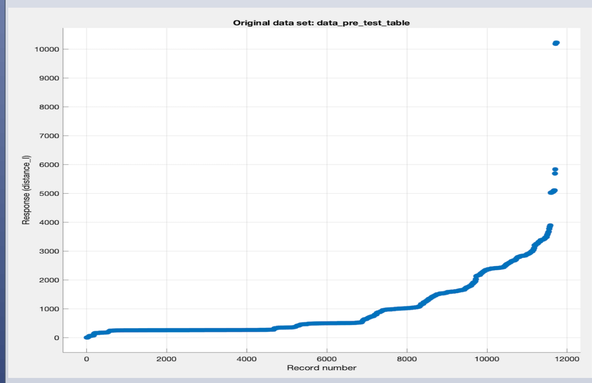
הפקולטה להנדסה ע"ש איבי ואלדר פליישמן אוניברסיטת תל-אביב

INTRODUCTION

The Project comprises an analysis of a problem, Which involves algorithms of improving the solution of the location detection with Machine Learning using MATLAB. It improves previous techniques, which detects the distance between the transmitter and the receiver by measuring Inputs and outputs phase over all different frequencies. It includes having datasets of the communication system, building an algorithm that estimates the parameters of the system. after that, we will develop an improved algorithm, using the same system, in order to find the system parameters. We will compare between those algorithms and make sure that indeed it is an improved algorithm. We will simulate the system and analyze the output properties of the system. We will make this process in different frequencies and have conclusions about the estimation.

DATASET AND FEATURE TABLE

- Our data set involves 70k samples of distances between 2 receiving and 2 transmitting antennas, which is been processed from Receiver.
- We have split the dataset into two sets, 10k dataset for training and 60k dataset for testing, we split the data set accordingly.
- The Dataset is a table in MATLAB made by raw data, originally containing the features and the accurate distance between the receiving and the transmitting antenna which is processed in MATLAB and in the Signal Processing Toolbox
- The Features Table consists of 11 features and 1 Response variable in the Features Table, features are selected from a MATLAB code written in the old model, Signal Processing Toolbox and the raw data



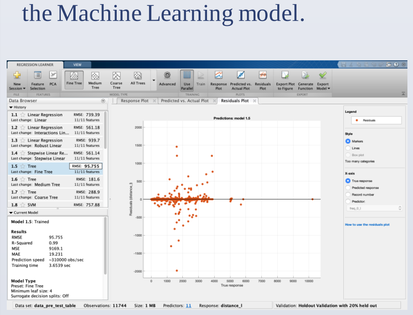
Distance vs Samples

METHODS

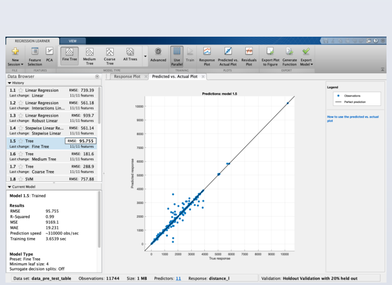
- We split our data into 10k samples of training set and 60k samples for testing the model.
- For the Model, We used 80% training and 20% of Validation, We used 20% Hold-on Validation.
- We found out that training the model with Fine Tree yields estimation of distance with RMSE 95.55 cm.
- Introduced in Regression Tree , variance reduction is often employed in cases where the target variable is continuous (regression tree), meaning that use of many other metrics would first require discretization before being applied. The variance reduction of a node N is defined as the total reduction of the variance of the target variable x due to the split at this node:

$$I_V(N) = \frac{1}{|S|^2}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}(x_i - x_j)^2 - \left(\frac{1}{|S_t|^2}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}(x_i - x_j)^2 + \frac{1}{|S_f|^2}\sum_{i\in S_f}\sum_{j\in S_f}\frac{1}{2}(x_i - x_j)^2\right)$$

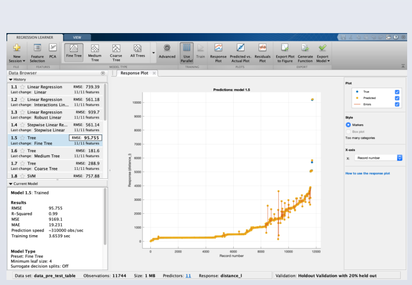
- Where S, S_t, and S_f are the set of presplit sample indices, set of sample indices for which the split test is true, and set of sample indices for which the split test is false, respectively. Each of the above summands are indeed variance estimates, though, written in a form without directly referring to the mean.
- We used Feature selection using neighborhood component analysis for regression, the feature weights diagonal adaptation of neighborhood component analysis (NCA) with regularization which indeed improves the Machine Learning model.



Residue Plot



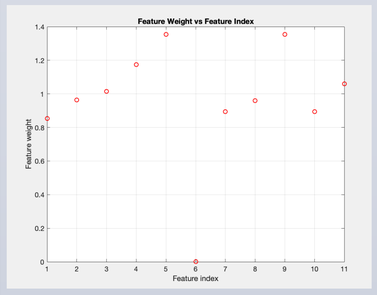
Actual vs Predicted Distance



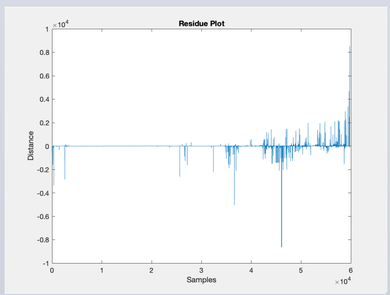
Response Plot

RESULTS

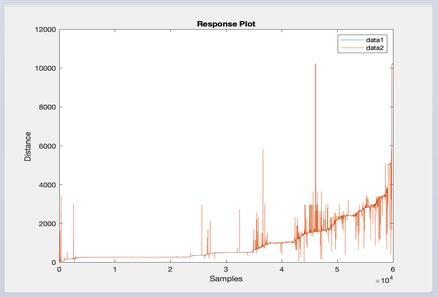
- The accuracy of the Validation data for the 20% Hold-out Validation is 99.4% with the estimation of distance with RMSE of 95.55 cm.
- The accuracy of the Testing data was 98.03% with the estimation of distance with RMSE of 111.94 cm.
- We used Fine Tree with neighborhood component analysis to increase the RMSE of Validation data from 48.51 cm RMSE to 95.55 cm RMSE.
- With Fine Tree and neighborhood component analysis to increase the accuracy of the model from 80.11% with estimation of distance with RMSE of 671.11 cm to accuracy of 98.03% estimation of distance with RMSE of 111.94 cm.



Feature Weight vs Feature Index



Residue Plot



Response Plot

PROCEDURE

- Collect the Raw Data from the Receiver, signal processing tool box and the text files
- Process the Raw Data in MATLAB into a Table
- Building a Machine Learning Algorithm using Regression Learner App in MATLAB
- Validation and Testing of the Machine Learning Model
- Improving the Model by Feature Selection using Neighborhood component analysis
- Validation and Testing of the New Machine Learning Model
- Transforming the MATLAB Model into Python code.
- Making an App that takes the features and predicts the Distance according to the Machine Learning Model
- Deployment of the Model in Real life

CONCLUSION AND FUTURE AREA OF INVESTIGATION

- Well-tuned regression algorithm preform excellent in predicting the distance between the transmitting and receiving antenna
- Choosing the dataset for training to increase the validation error in order to decrease the testing error was done quite successfully
- This project helped us to explore the opportunity to learn Machine learning in MATLAB. The project helped us to improve our skills in MATLAB environment, working with new directories.
- The Future Area of Investigation of the new model will be to improve the algorithm by hierarchal neural network
- Given More time we would like to compare the results provided by the Machine learning algorithm vs the results provided by the hierarchal neural network vs the MATLAB code that compute the predicted distance without the help of Machine Learning

REFERENCES

- https://www.mathworks.com
- https://www.mathworks.com/help/stats/regression-learner-app.html
- https://www.mathworks.com/help/stats/regression-tree-ensembles.html
- https://towardsdatascience.com/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054