Chapter 1

Typhoon Prediction: A Supervised Machine Learning Approach

The enormous devastation caused by super typhoon Haiyan (2013) motivates us to consider whether data from past typhoons hold a key to predicting the grade (“category”) of an emerging tropical storm from early best track data (lat, lon, windspeed and pressure in the first 18 hours). In particular, we describe a machine learning (ML) approach to classification of typhoon grade as well as ML regression models of maximum wind speed. We train and validate our model and make ’hindcasts’ using Japan Meteorological Agency (JMA) best-track data.

1.1. Introduction

In November 2013, Typhoon Haiyan struck the Philippines with sustained wind speeds of nearly 200 miles per hour, claiming over 6,000 lives, displacing 4 million people, damaging over a million houses, and resulting in a request for over 750 million dollars in humanitarian aid. Devastation caused by “super typhoons" like Haiyan makes the ongoing development of early warning systems an absolute necessity to provide vulnerable communities sufficient preparation time to mitigate damage to both life and property. This motivates us to apply Machine Learning algorithms to predict the typhoon grade (category model), and maximum wind speed (regression model).

1.2. Overview of Supervised Machine Learning

Supervised Machine learning (ML) refers to a set of algorithms which utilize a training data set to develop a model which takes one or more input variables called *predictors* and outputs the value of a variable called a *response*. We will use Japan Meteorological Agency (JMA) data for 2009-2014 as training data:

- Predictor Variables: Lat, Lon, WS and Pressure at times $t = 0, 6, 12, 18$ hours.
- Response Variables: Typhoon grade (3,4, or 5) and maximum sustained wind speed.

In the next section we describe a number of different ML algorithms such as decision tree, k-nearest neighbor, ensemble. In this chapter, we do not consider *unsupervised* ML algorithms where the response variables are not known in advance, and hence not specified in the training data set.

When the response is a discrete set, such as is the case for typhoon grade, we use a *categorical* ML algorithm. When the response is continuous, such as maximum wind speed, we use a *regression* algorithm. In both cases, the accuracy of the prediction is our primary performance measure.

In the case of categorical models, one way to report accuracy is via a *confusion matrix*. Figure 1.1 gives an example of the latter, for a model which on the training data set accurately classifies 13 grade, 13 grade 4, and 47 grade 5 typhoons. In addition, specific information about the classification errors is given. For example, 8 typhoons predicted as grade 5 were actually grade 4, and 9 typhoons predicted as grade 4 were actually grade 5.

For a regression model, a scatter plot which compares the actual to the predicted values, with the difference between them called residuals is a standard visual representation of the error. The root mean squared error (RMSE) is calculated as the square root of the mean squared residuals. Figure 1.2 shows such a scatterplot, with in particular shows fairly good wind speed prediction of Grade 5 violent typhoon (maximum wind speeds of at least 105 knots/hr) as well as grade 3 tropical storms, but has greater difficulty in achieving accurate wind speed prediction for the typhoons in the mid-range (category 4).

Once a model has been developed using the training data set, it should be validated using a *validation* data set. In our case, we use JMA data for 2015 as our validation set. If the model does not perform well on the validation data, we must revise our model,

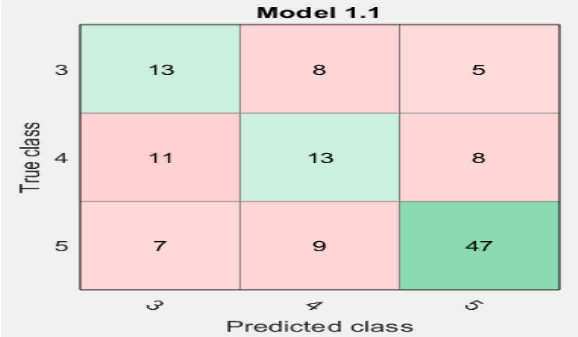


Figure 1.1. Confusion matrix for a typhoon grade Fine Tree categorical ML model.

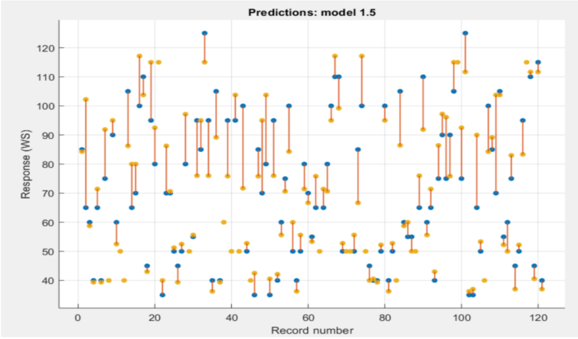


Figure 1.2. Fine Tree ML regression model RMSE: 11.0

returning to the training stage. If the model does perform well on the validation set, we are ready to test it on new data. For a quasi-operational (“hind-casting") model, our "new" data is JMA data for 2016.

1.3. Algorithms

Predicting the JMA grade of an emerging tropical storm based on its initial 18 hour best track data is an example of a standard supervised learning problem ([2]). A training data set has the form $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$ where

- $\mathbf{x}_*$=(YEAR,MONTH,DAY,LAT1,LON1, PRES1,WS1,LAT2,LON2,PRES2,WS2, LAT3,LON3,PRES3,WS3,LAT4,LON4,PRES4,WS4) specifies the date and first four best track points of an emerging tropical storm (recorded windspeed of 35 knots); and

- $y_* \in \{3,4,5\}$ is the highest typhoon grade achieved by the storm.

A learning algorithm outputs a function $f(\mathbf{x})$ called a classifier, which given values for the predictor variable $\mathbf{x}$, outputs a value for the response variable $y = f(\mathbf{x})$ (typhoon grade). An ensemble method combines responses of two or more classifiers to improve the prediction accuracy of individual classifiers.

There are a number of basic supervised ML algorithms [1] such as

- Decision Trees and Random Forests)

- Support Vector Machines (SVM)

- K Nearest Neighbors (KNN)

- Ensemble

1.3.1. Decision Trees and Random Forests

One type of ML classification algorithm is based on a decision tree such as shown in Figure 1.3

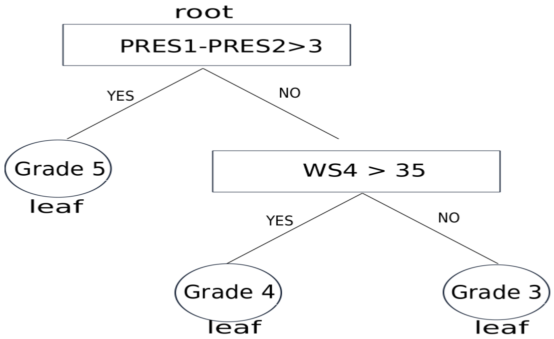


Figure 1.3. Simple example of a decision tree used to classify typhoon grade.

A Random forest algorithm uses N decision trees and selects the majority category. By increasing the number of leaves in the Decisions trees, a classification algorithm becomes a type of regression model.

1.3.2. Support Vector Machines (SVM)

The idea of support vector machines (SVM), a type of ML algorithm mainly applied to classification problems, is to use hyperplanes to separate predictors into categories as shown in Figure 1.5. An optimal hyperplane maximizes the distance between the separated categories. In some cases, a transformation of the data is needed before SVM can be applied.

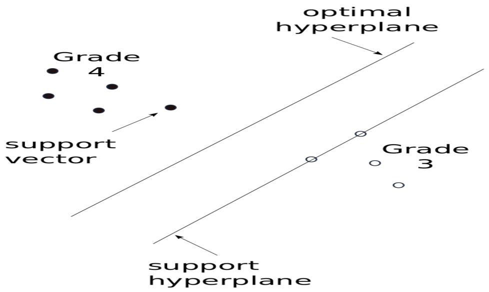


Figure 1.4. Simple illustration of SVM classification.

1.3.3. K Nearest Neighbors (KNN)

The K Nearest Neighbor algorithm (KNN) makes a prediction based on the k closest predictors. In the case of classification, the category with the highest frequency among the K neighbors is selected. In the case of regression, an average of the k closest predictors may be used. Different choices of K may be investigated to give the best separation between categories.

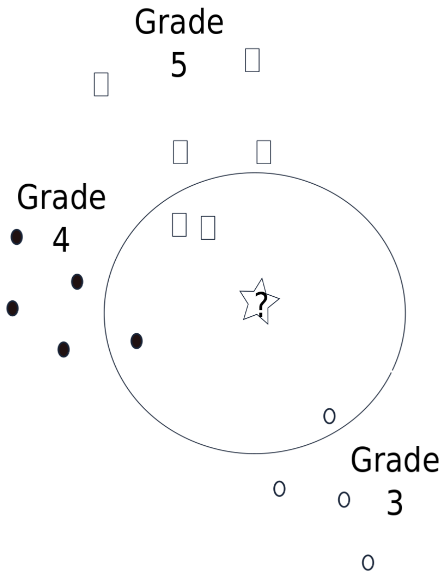


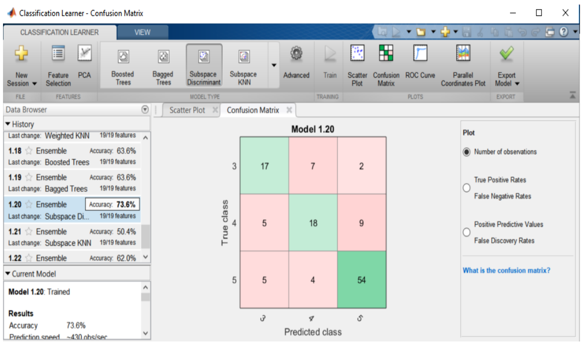
Figure 1.5. Simple illustration of KNN classification. In this case K=4.

1.3.4. Ensemble

The ensemble method [2] takes a set of individual algorithms and combines them in some way, for example by a majority classification or a weighted average of regression values. An important research area in supervised ML is development of methods to construct good ensembles.

1.4. Exercises

1. What is the overall percentage of correct typhoon grade prediction based on the information in an ensemble model ML confusion matrix shown in Figure 1.6? How well does the model predict grade 5 typhoons (give percentages of correct, false positives and false negatives).)



Last change: Weighted KNN 19/19 features

1.18 ☆ Ensemble Accuracy: 63.6%
Last change: Boosted Trees 19/19 features

1.19 ☆ Ensemble Accuracy: 63.6%
Last change: Bagged Trees 19/19 features

1.20 ☆ Ensemble Accuracy: 73.6%
Last change: Subspace Di... 19/19 features

1.21 ☆ Ensemble Accuracy: 50.4%
Last change: Subspace KNN 19/19 features

1.22 ☆ Ensemble Accuracy: 62.0%

Model 1.20: Trained

Results
Accuracy 73.6%
Prediction speed ~430 obs/sec

Model 1.20

True class \ Predicted class	3	4	5
3	17	7	2
4	5	18	9
5	5	4	54

Plot
● Number of observations
○ True Positive Rates / False Negative Rates
○ Positive Predictive Values / False Discovery Rates

Figure 1.6. Confusion matrix for a MATLAB subspace discriminant ensemble classification algorithm.

2. An *accurate* classifier is one that has an error rate of better than random guessing on new **x** values. Two classifiers are *diverse* if they make different errors on new data points. Prove that a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse. (See [3].

1.5. Computer Project

Project website: https://www.overleaf.com/project/5c057323a87843644f4893a6

MATLAB has a Statistics and Machine Learning Toolbox which has apps to build both classification and regression models. The main steps in creating these ML models are:

- Train MATLAB's suite of ML models using a specified dataset.
- Choose one of the models with high accuracy, and export a MATLAB function based on your selected trained model.
- Apply the exported model to a validation dataset. If satisfied, continue to the next step. If not, repeat the previous steps.
- Test the model on a new data set and record the error.

1. Using CategoryTrainingData.xlsx, CategoryData15.xlsx, and CategoryData16.xlsx for your training, validation, and "new" data sets respectively, use MATLAB to develop a ensemble classification ML model for typhoon grade. Include the confusion matrices in your report.
2. Using the RegressionTrainingData.xlsx, RegressionData15.xlsx, and RegressionData16.xlsx for your training, validation and "new" data sets respectively, use MATLAB to develop a KNN regression model to predict the time until the maximum wind speed is first attained on the best track. Include a residual error analysis in your report.

i Select the Classification Learner App, New Session, From File

ii. Open CategoryTraingData.xlsx. After a few seconds you should see the uploaded file in an IMPORT tab. Click on Import Selection.

iii. Choose "CATEGORY" for response and then click on "Start Session".

iv In the CLASSIFICATION LEARNER tab, click on All, and then Train.

v. In the History window, you should see a number of ML algorithms: 1.1 Tree (Fine Tree), 1.2 Tree (Medium Tree) etc.). Select Ensemble, then Bagged Trees.

vi. Click on Confusion Matrix to get information on the trained model's accuracy.

vii. Click on Export Model, and select Generate Code. A function trainClassifier(trainingData) should appear in the MATLAB editor. Change the function name and data file to TyphoonGradeClassifier.

viii. Run the file MakeTyphoonGradeClassifierModel.m

Project Team: Michaela Flitsch, Mark Nussbaum, Zach Oslund, Matthew Rueger.

EXAMPLE SCRIPT MakeTyphoonGradeCalssifierModel.m

```
clear all; close all;
%% Read in Traing Data
X=xlsread('CategoryTrainingData.xlsx','Sheet1','A2:T122');
YEAR=X(:,1);MONTH=X(:,2);DAY=X(:,3);
LAT1=X(:,4);LON1=X(:,5);PRES1=X(:,6),WS1=X(:,7);
LAT2=X(:,8);LON2=X(:,9);PRES2=X(:,10);WS2=X(:,11);
LAT3=X(:,12);LON3=X(:,13);PRES3=X(:,14);WS3=X(:,15);
LAT4=X(:,16);LON4=X(:,17);PRES4=X(:,18);WS4=X(:,19);
CATEGORY=X(:,20);
T=table(YEAR,MONTH,DAY,LAT1,LON1,PRES1,WS1,LAT2,LON2,PRES2,
   WS2,LAT3,LON3,PRES3,WS3,LAT4,LON4,PRES4,WS4,CATEGORY)

%% Run Exported Function to Create Trained Model
[trainedClassifier, validationAccuracy]=
   TyphoonGradeClassifier(T);
```

References

[1] Bonnardot, G., 8 Machine Learning Algorithms Explained in Human Language. Available at https://www.datakeen.co/en/8-machine-learning-algorithms-explained-in-human-language/

[2] Dietterich, T. Ensemble Methods in Machine Learning. Available at http://web.engr.oregonstate.edu/∼tgd/publications/mcs-ensembles.pdf

[3] Hansen, L. and Salamon, P. Neural network ensembles. 1990. IEEE Trans. *Pattern Analysis and Machine Intell*, 12, 993-1001.

[4] Japan Meteorological Agency, Best track archives. Available at http://www.jma.go.jp/ jma/jma-eng/ jma-center/ rsmc-hp-pub-eg/trackarchives.html.