Machine Learning Approaches to Predict Forest Fires Level

Sicheng Fu

1 Introduction

Forest fire is a kind of natural disaster that cause terrible damage to environment, ecosystem and economy. On average, there are more than 200,000 times forest fires around the world every year and the burned forest area accounts for more than 1 percent of the world forest total area. Despite many countries have invested a lot of money to prevent forest fires, some environmental factor still can not be well controlled. So fast prediction is a key factor in forest fire fighting.

Therefore, this project aims to predict the forest fire levels based on some machine learning algorithms. The data we processed in this project is from UCI machine learning repository forest fire data[1], which is a real-world forest fire data collected from the Montesinho natural park in northeast of Portugal. Several experiments were conducted by considering different machine learning approaches (i.e. several linear regression methods, neural network and SVM), and give analysis for each methods. The expected results of the model can provide a relatively accurate prediction of forest fire levels.

The report is organized as follows. In section 2, we describe the structure of forest fire data. In section 3, we states the three classification algorithms being used and results. In section 4, assessment and conclusions are drawn.

2 Dataset Description

The original forest fire dataset is from UCI machine learning repository[1], which collected in Portugal Montesinho natural park. The data using two sources with a total of 517 entries, one was collected from park inspectors, another was collected by meteorological station[2].

2.1 Features

This dataset has total 12 features from 4 different aspects.

(1). Spatial location: $[X, Y]$

(2). Time information: $[month, day]$

(3). FWI components: $[FFMC, DMC, DC, ISI]$
(4). Weather condition: $[temp, RH, wind, rain]$
The map of Montesinho park has been divided into a 9X9 grid, the values of X axis and Y axis ranges from 1 to 9 represent the grid location when the fire burns. Month (January to December) and day of the week (Monday to Sunday) selected as temporal variables, season is an important factor to affect forest fire[2].
Following four features $[FFMC, DMC, DC, ISI]$ are Forest Fire Weather Index (FWI) components that account for the effects of fuel moisture and weather conditions on fire behavior[3].
Weather condition also included in this dataset, temperature (℃), relative humidity (%), wind speed (km/h) and rain (mm/m^2). These features are basic weather variables related to forest fires.

2.2 Label

The burned area (h) will be the output label, which represent the area size of the fire burned. Zero value means the burned area is smaller than $100m^2$. According to wiki[4], related department usually classify 4 levels for forest fire damage: normal (burned area $<$1 hectare), high (1$<$burned area$<$100), very high (100$<$burned area$<$1000) and extreme (burned area$>$1000 hectare). The distribution of forest fire data in each fire danger level is shown below:

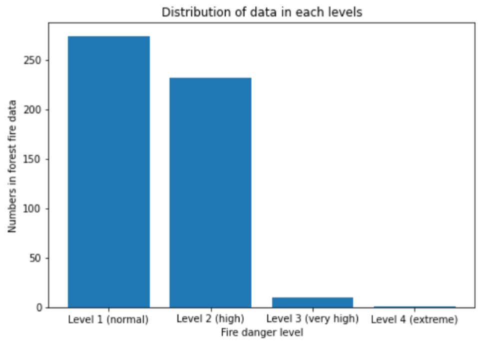

Figure 1: Distribution of forest fire data

It presents the majority of forest fire area is in a small size. Consider this limitation, we just manually assign binary labels for fire burned area, if burned area is smaller than 1 hectare as normal fire danger level $[y_i = -1]$ (243 samples), otherwise as high fire danger level $[y_i = 1]$ (274 samples).

3 Algorithms and results

Before conduct experiment, data processing is required. The nominal variables (month and day) need transform to numeric variables first. Using each corresponding numbers to represent month and day.
Since the unit for each features is different, normalize data is necessary. In this project, we apply zero-mean normalization method with zero mean and one standard deviation.

70% of the data is splitted into the training set and the rest 30% of the data is splitted into the testing set. A k-fold cross-validation is applied to the best model by splitting the training set into new training groups and validation groups. $Accuracy = 1 - \frac{number\_of\_errors}{total\_size\_of\_vectors}$ is used to evaluate the model.

3.1 Linear Regression

In this experiment, we use four kinds of linear regression methods: Least square method, Ridge regression, LASSO regression and Trucated SVD, to predict the fire danger level based on the input features. 5-fold cross validation is applied for each methods to prevent overfitting and get average accuracy.
Least square method: $argmin_w = ||Xw - y||_2^2, w = (X^T X)^{-1} X^T y$
Ridge regression: $argmin_w = ||Xw - y||_2^2 + \lambda||w||_2^2, w = (X^T X + \lambda I)^{-1} X^T y$
LASSO regression: $argmin_w = ||Xw - y||_2^2 + \lambda||w||_1$
Truncated SVD: $argmin_w = V\Sigma_r^{-1} U^T y$
(1) Experiment
Randomly split the training set into 5 subsets, complete the 5 times cross validation, and use the best prediction accuracy rate to find the optimal weight and compute the average accuracy. Optimal weight is used to get the test accuracy.
The λ for ridge and LASSO regression may range from 10^{-10} to 10, randomly produce 30 points spaced logarithmically and repeat this range of λ values to obtain a set of solutions w. Use the solution that has best accuracy to find the best λ for each validation. A function that implement iterative soft thresholding via proximal gradient descent is been used to solve the LASSO problem. And this function use a hot procedure to find solution with different λ.
For Truncated SVD, use the pseudo-inverse to estimate w for each choice of the rank-approximation r. Rank approximation r ranges from 1 to 12. Find the optimal w and corresponding r that minimize the prediction error to compute the testing accuracy.
(2) Prediction results

Linear regression results				
Model/Accuracy	Least square	Ridge regression	LASSO regression	Truncated SVD
Training Accuracy(%)	51.64	58.22	55.78	51.64
Testing Accuracy(%)	46.45	53.55	47.10	47.74

For all linear regression models, they obviously did not present a good result. The fire burned area may not have a simple linear relation with FWI components and weather condition. Among all these regression methods, ridge regression gives a relatively better result, about 54% testing accuracy. The other three regression methods only have 47% testing accuracy. To eliminate this drawback, nonlinear functions like NN and SVM should be considered.

3.2 Neural Network

For neural network, we apply multilayer perceptron, which is a classical artificial neural network (ANN),to train our model. The network we design involves four layers, one in-

put layer with input dimensions equals to number of features, two hidden layers with 100 neurons, one output layer with one unit. Each hidden layers utilize ReLU as activation function, and output layer utilize tanh as activation function. The structure of network is in figure 2.
(1) Experiment

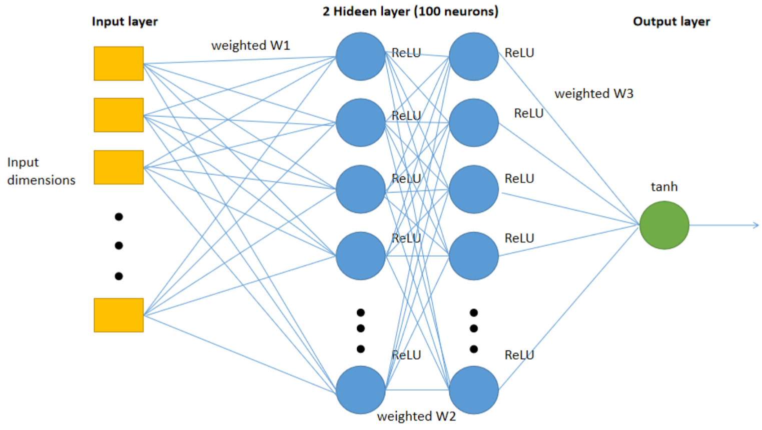

Figure 2: Structure of MLP

The model is built using the keras library[5]. The packages of keras.model.Sequential, keras.utils.np_utils and keras.layers.core have been used. To avoid overfitting problem, we can add a dropout layer in the hidden layer, randomly remove 20% of data to avoid neurons assign too much weights in one feature.
To compile the model, we choose binary cross entropy as loss function and Adam (which is different to classical stochastic gradient descent,unilike the GSD maintains a learning rate for all weight updates without change, Adam calculates an exponential moving average of the gradient[6]) as optimizer, and evaluate each epoch by accuracy. 20% of data assign as validation data, 60 times training epochs and 128 batch size is setting.
(2) Prediction Results

Neural Network results	
Model/Accuracy	Mulilayer perceptron
Training Accuracy(%)	61.89
Testing Accuracy(%)	59.97

Neural Network present best accuracy among all approaches, it has nearly 60% prediction accuracy, since NN can better fit nonlinear data. $tanh$ activation function and $Adam$ optimizer works better in this problem condition.

3.3 Support Vector Machine

Support Vector Machine (SVM) is a perfect algorithm for binary classification problems. Since the input data might be non-linearly separable. To design a better SVM model, the performance of the models using linear or non-linear kernel functions will be compared. The kernels functions are:
Linear kernel : $x^T y$

polynomial functions : $(x^T y + 1)^p$
radial basis function : $exp(-\frac{||x-y||_2^2}{2\sigma})$
(1) Experiment
The model is built using the scikit-learn library[7]. The function sklearn.svm.LinearSVC is used for linear support vector classification, the function sklearn.svm.SVC(kernel='rbf') with RBF kernel and the function sklearn.svm.SVC(kernel='poly') with polynomial kernel are used for kernel based SVM. 5-fold cross validation is applied for each kernel method to prevent overfitting and get average training accuracy.
(2) Prediction Results

SVM results			
Model/Accuracy	SVM(RBF)	SVM(polynomial)	SVM(Linear)
Training Accuracy(%)	54.97	56.45	50.81
Testing Accuracy(%)	54.13	57.61	47.74

For three SVM models, polynomial kernel gets a relatively better result, in contrast linear kernel did not perform well.

4 Assessment and conclusion

4.1 Assessment

For four linear regression models, ridge regression and LASSO regression has a λ penalty term, helps balance the bias and variance. With appropriate λ, they can gives better results compare with lease square. But many real-world data not strictly follow a linear relation, non-linear method usually has better performance.
Neural network has strong nonlinear fitting ability and can map complex nonlinear relationships. Consider the forest fire data is irregular and complex, it is expected that NN works better. However, the exact optimal value for batch size and epoch in network is hard to determine, thus multiple times testing can make more sense. SVM can map the low-dimensional space into high-dimensional space, because fire data is non-linear separable, RBF and polynomial present higher accuracy than linear kernel.

4.2 Conclusion

Overall, the highest prediction accuracy is about 60% among all approaches, which seems not work well with this forest fire data. But in reality, various reasons affect the fire burn area size, especially the human factors, but this feature did not present in data. As we can see from this dataset, two samples with similar feature values may have totally different classification. This phenomenon will greatly influence the prediction accuracy. Therefore, forest fire prediction problem is a challenge task, involves more factors (e.g. types of vegetation, human activity frequency) to consider. The paper just give roughly 70% prediction accuracy for forest fire[2].
For future works, more samples from different danger levels and more features related with forest fire need to consider. To keep the rigorous, the data of weather condition and FWI components from longer time period in one region should be collected. So more levels can be classified to improve the prediction accuracy.

5 Github Link

https://github.com/SichengIce/CS532_project

6 References

[1] UCI Machine Learning Repository Forest Fire Data Set:
http://archive.ics.uci.edu/ml/datasets/Forest+Fires
[2] P. Cortez and A. Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, December, Guimaraes, Portugal, pp. 512-523, 2007. APPIA, ISBN-13 978-989-95618-0-9.
[3] Canadian Forest Fire Weather Index (FWI) System:
https://cwfis.cfs.nrcan.gc.ca/background/summary/fwi
[4] Wildfire: https://en.wikipedia.org/wiki/Wildfire
[5] Keras library: https://keras.io/api/models/sequential/
[6] Introducion to Adam Optimizer:
https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/: :text=Adam
[7] scikit-learn library. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html