The Search for Twitter Bots

Amir Yazdani[1]

Abstract— **Finding spam users and bots in Twitter is an important issue for the company. As a course project for machine learning, students are doing a competition to use machine learning techniques to find those users with highest accuracy. In this report, seven algorithms are developed and implemented on training dataset and results of accuracy on test dataset is provided.**

I. INTRODUCTION

Recent developments of technologies specially in computer science has vastly effected human life. One of the areas that has been effected more is media, specifically social media. In comparison to 20 years ago that most of people used TV and newspapers to reach news, people use their smartphones and tablets and even their smartwatches to connect to the world. Nowadays, social media are playing very important role in taking control and guiding people's life, comparing to public medias. Among all social media, Twitter has is popular since it provides important information in very simple and short way. If you have very short time to check the headlines, you are tend to use Twitter and these people are the target market of Twitter. Regarding that target population, keeping information clean in Twitter is very important. People will not tolerate if in their 2 minutes of checking twitter, they see a lot of ads and bots. Fake news are also another concerning issue in twitter and a robot or a person can start a fake news and due to likes and retweets, it can goes around and spread so fast.

As it can be learned from above, removing spams, bots and ads from twitter and clean the information is very demanding task for companies governing social media. The good news is those recent technologies in computer science such as Machine Learning, Deep Learning, etc can help to do the task. There are several learning algorithms that can be useful detecting bad information and bad users in social media. They normally use a large training dataset which is provided by users and and train algorithms.

Solving this problem is very interesting specially under a competition. It is a very good setup to actually use what students have learned through the semester in a real problem solving case that is a current research in academics society and high-tech companies.

A. *Problem statement*

In Machine Learning course project, same problem is provided to solve using different algorithm covered in learned in class. Dataset is provided in Kaggle website[1] as well as a leader-board page. It includes different files for training, evaluation and test. Training dataset includes 29049 training examples with 16 features. Features are continues real numbers with different mean and standard deviations. Label is either 1 (spam) or 0 (not spam). evaluation and test dataset include 6226 and 6224 examples, respectively.

II. METHODOLOGY

To solve the problem presented in section I, different machine learning algorithms(totally 7 algorithms) have been used.

A. *ID3 Decision Tree*

The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy $H(S)$ (or information gain $IG(S)$) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split by the selected attribute (e.g. age is less than 50, age is between 50 and 100, age is greater than 100) to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before[2].

First, 5-fold cross-validation is used to find best maximum depth and best depth is used to train the algorithm over training dataset. To have ability to split the dataset when tree is growing, feature values are normalized to have zero mean with unit standard deviation. Values of each feature is divided into four sections, based on standard deviation(less than -std, -std to zero, zero to 1std, greater than 1std).

B. *CART: Classification And Regression Trees*

Classification and Regression Trees (CART) [3] is a term introduced to refer to Decision Tree algorithms that can be used for classification or regression predictive modeling problems. The representation for the CART model is a binary tree.Splitting the dataset is based on two costs:

- *Regression*: The cost function that is minimized to choose split points is the sum squared error across all training samples that fall within the rectangle.
- *Classification*: The Gini cost function is used which provides an indication of how pure the nodes are, where node purity refers to how mixed the training data assigned to each node is.

[1] Amir Yazdani is with Department of Mechanical Engineering, University of Utah, Salt Lake City, UT `mojtaba.yazdani at utah.edu`

[1] https://www.kaggle.com/c/uofu-ml-fall-2017

[2] https://en.wikipedia.org/wiki/ID3-algorithm

[3] https://en.wikipedia.org/wiki/Decision-tree-learning

C. Gaussian Naive Bayes

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let μ_k be the mean of the values in x associated with class Ck, and let σ_k^2 be the variance of the values in x associated with class Ck. Suppose we have collected some observation value v. Then, the probability distribution of v given a class C_k, $p(x = v \mid C_k)$ can be computed by plugging v into the equation for a Normal distribution[4]:

$$p(x|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu)^2}{2\sigma_k^2}} \tag{1}$$

Normalization of feature values is important in this method of learning. normalized data has mean of zero with unit standard deviation.

D. SVM

A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier[5].

E. Logistic Regression

Unlike linear regression, the prediction for the output is transformed using a non-linear function called the logistic function (sigmoid). Logistic regression measures the relationship between the categorical dependent variable (target class: spam/not spam) and one or more independent variables by estimating probabilities.

F. SVM on CART

During implementation of normal CART it was found that it takes a lot of time since it needs to generate k number of trees in each iteration of k. This makes it hard to use. The idea that came up is to boost the CART using a SVM. Using SVM over CART makes benefit of low depth trees developed with low amount of data in CART and drives a stronger prediction.

G. Bagged Forest CART

Same as previous one, we use Bagging instead of SVM which returns the most-voted one.

III. Implementation

All of above algorithms implemented in Python and some Python packages such as Numpy, statistics, etc has been used to make calculations faster and easier. During the coding, It found that python is very sensitive to parameter types such as float, int, list, ndarray, etc. To make sure that Numpy calculation does not change the type to ndarray, in each main function, there is a checkpoint to check if the required data type is provided.

In implementation of algorithms with boosting, due to the time consumption, hyper-parameters were chosen manually based on the trend observed in results. Doing cross-validation made them very hard to debug if any small error happened to be there. Those trends are discussed in the following section.

In trainings and cross-validation, for 10 epochs data is shuffled and used for learning.

To make the result ready for submission to Kaggle, user IDs corresponding to datasets were read from files and CSV package in python is used to put them together with predicted labels in a CSV file.

At start, algorithm without boosting were used such as ID3, SVM and Logistic Regression. Based on results, it was concluded that to get higher accuracies, boosting is required. So, decided to use bagging and SVM on trees developed by CART result revealed the effect of boosting on increasing the accuracy of prediction.

IV. Results and discussions

As indicated in the project description, trained algorithms were implemented on both evaluation and test dataset. results for the evaluation dateset are submitted to Kaggle for competition and results for test dataset are provided in figure 1 and table 1.

A. Gaussian Naive Bayes

Gaussian Naive Bayes did the fastest performance among all algorithms. It did not have any hyper parameters and based on Gaussian likelihood of training dataset predicted value for the labels of evaluation and test dataset. Resulted accuracy is around 0.67 and it is the lowest accuracy among algorithms. Normalization of feature values improved the accuracy a lot.

B. ID3 Decision Tree

5-fold cross-validation of training resulted that best depth is 9 and using this depth, accuracy is 0.715 on test dataset. It did better that Gaussian Naive Bayes. It seems that the way we split the data should effect the accuracy a lot. Since we assume that we do not have any prior knowledge about each feature, so we have normalized all and it may cause low level of learning a feature.

[4] https://en.wikipedia.org/wiki/Naive-Bayes-classifier

[5] http://scikit-learn.org/stable/modules/svm.html

Algorithm	Accuracy	Hyper-parameter
Gaussian Naive Bayes	0.667	
ID3 Decision Tree	0.715	Depth=9
Logistic Regression	0.821	learning ratio=0.0001, $\sigma^2 = 10$
SVM	0.823	Learning ratio=0.001, Trade-off=10
CART	0.914	Depth=5
Bagged Forest CART	0.936	Depth=10, number of trees=5000, number of samples=500

TABLE I

ACCURACY AND HYPER-PARAMETER OF DIFFERENT ALGORITHMS

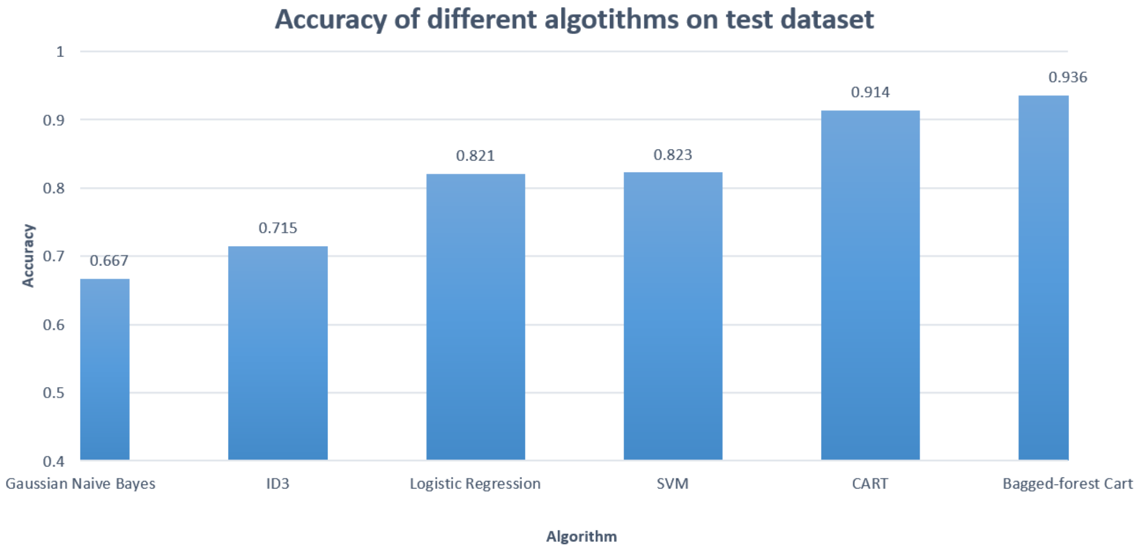

Fig. 1. Accuracy of algorithms

C. *Logistic Regression*

In logistic Regression, 5-fold cross-validation with 10 epoch of training resulted that best learning ratio is 0.0001 and best σ^2 is 10. It results with accuracy of 0.821 on test dataset. It has a step up toward accuracy in comparison with Gaussian Naive Bayes and ID3 what had accuracy around 0.7

D. *SVM*

SVM behaves very similar to logistic Regression. Here 5-fold cross-validation with 10 epoch of training resulted that best learning ratio is 0.001 and best σ^2 is 10. It results with accuracy of 0.823 on test dataset.

E. *CART*

CART gives very good accuracy of 0.914 with max depth of 5. Due to very long run-time and debugging requirement, cross-validation is not used here to find best hyper-parameter. Instead, they are changed manually. There is a trend that. First, a small portion of the training dataset used for training and it gave pretty good result. Then, by increasing it gradually to the whole training dataset, improvement in accuracy observed. We also changed the max depth to 9 and result was almost the same. They key feature concluded to be the cause of higher accuracy than normal ID3 is the way CART uses regression when it wants to split the dataset to build trees.

F. *Bagged-forest CART*

As predicted and based on what learned in class, add boosting and ensemble on top of normal algorithms has improved the accuracy. normal CART gave maximum of 0.914 accuracy but adding bagging, gives accuracy of 0.936. It is found out that increasing number of trees has the most effect on improving accuracy. It is done until the point that it stops improving the accuracy (number of trees =5000). The other parameter that had effect on accuracy is max depth of trees and max-depth=10 gives us best result. In all cases, number of samples is 500.

G. *SVM on CART*

The better-expected algorithm used for the project is SVM on CART. Unfortunately, against what we expected, the accuracy is less than 0.45 and it is believed that somewhere in the code there is problem. It is still under debugging and result will be posted as soon as we get the final results. the code for this algorithm is also attached in the zip folder.

V. Lesson Learned

This project was a great opportunity to actually implement what we have learn in class and homeworks on a big dataset.

Competition-type of project helped students to try hard to improve their algorithms and codes. It showed that only writing a code based on what we learned is not enough to reach very good and reliable accuracy that can be used in real applications. It also showed the author(as a mechanical engineer)that in field of machine learning and computer science, creativity and developing new ideas can strongly improve algorithms. IT requires a design procedure and knowledge about each algorithm requirements and in what situation which one gives better output.

VI. FUTURE WORK

To peruse the project of fining spams in Twitter, new algorithms should be used. First of all, it is learned that before selecting algorithm to implement, it is required to do a search through different algorithms and find the one that are suitable for the provided dataset. It also required to design combination of algorithm together using boosting and ensembles.

What is learned in this project and codes that are developed can be used for future project in research such. In particular, they can be used to predict collision between human and robot when there are working side by side or in swarm robots.