Fake Follower Detection

Adam David, Houston Koester

Introduction

In an effort to increase their perceived popularity, Twitter users can purchase services that use large quantities of fake bot accounts to increase their follower count or amplify their tweets. Given that such bot accounts have, in recent years, been used by malicious parties to disrupt or modify Twitter discussions on a large scale, it is beneficial to have the capability to distinguish between legitimate Twitter users and the aforementioned bot accounts in such a way that one could analyze the followers of a particular user and estimate how many of them are bots. As such, in this project, we focus on developing a model that distinguishes legitimate users from bot accounts.

Problem Description

Given a Twitter user and the current state of their profile, we seek to determine if they are a bot or a legitimate user, with the goal of identifying the proportion of bots present in a user's list of followers. With regards to the given project titles, this is Project 3.2: Fake Followers Detection. We emphasize that we seek to analyze the user based off of the *current* state of their profile. Despite the fact that information about the past state of the account may be useful for determining if they are a legitimate user or not, this type of information is not practical in use, as it requires accounts to have existed for some period of time, and requires data collection over some period of time, neither of which are necessarily available to analyze -- If an account has only existed for a week and a model requires months of data, the model can not determine whether or not the account is a bot until such data is collected. By that time, the account could have caused damage. Examples of this kind of data include the number of follows and followers an account has over time.

Model description

To select a model, we first selected a set of features to use, which, for each user ID included number of followers, number of following, number of tweets, name length, the average number of mentions in the user’s tweets, the percentage of a user’s tweets that contain links, and the percentage of a user’s tweets that are similar to another of their tweets. The latter was computed for each user by converting each of the user’s tweets into a word count vector with a set of stop words removed, each element representing the number of times a particular word appeared in the tweet. Two tweets were considered to be similar if the cosine between their word count vectors was greater than 0.9. This was chosen over a direct approach of comparing the text of two tweets, as two tweets that differ in for example only one word or perhaps a URL should not be considered different.

Data was randomly split into training and test splits, with 80% of the data being for training and testing and the remaining 20% being used for test. Using the training set, a selection of models were trained across a variety of parameters, with 5-fold cross validation used within the training

set to identify the best of the parameters. After tuning all parameters, the best of the trained models, which ended up being a random forest approach, was selected based on performance in the test set. A table of tested models, as well as their best found parameters and the associated estimate of their out of sample misclassification rate is provided below.

Model	Parameters	Misclassification
Random Forest	min_samples_split = 5 n_estimators = 500	6.06%
Gradient Boosting	learning_rate = 0.2 n_estimators = 500	6.89%
k-Nearest Neighbors	n_neighbors = 20	8.33%
Regularized Logistic Regression	L2 norm penalty with coefficient 10	9.11%

As can be seen from the table, the random forest model performed the best.

Dataset used

The dataset used for this model was the caverlee-2011 dataset accessible from (Bot Repository (iu.edu)) This dataset consists of 22,223 Twitter bots that engage in mass following, link spamming, or spam promotion, as well as 19,276 legitimate users. For each entry, the dataset contains the account's user ID, creation date, number of followers and following at the time of data collection, the time of data collection, the number of tweets the user has posted, the length of the user's profile description, and a list of tweets posted by the user. To reduce computation time, we randomly selected 1000 bot users and 1000 legitimate users from this dataset and filtered out entries with malformed or incomplete data, resulting in a data set consisting of 982 bot users and 999 legitimate users. Three additional features, namely, the average number of mentions in a user's tweets, the percentage of a user's tweets that are similar to another tweet that they've posted, and the percentage of a user's tweets that contain a link were computed and added to the dataset, as described in the model description. The label associated with each entry is a binary indicator "Bot", with a value of 0 if the user is legitimate, and 1 if the user is a bot. This randomly sampled dataset, as well as the extra features computed, are included in the file completedData.xlsx.

Results

As can be seen in the model description, a random forest model with 500 trees, splitting nodes only if at least 2 samples are in the resulting split. Testing this model on the test set yields an out of sample misclassification rate of 6.00% and an AUC of 0.940, slightly better than the gradient boosting model's 6.55% misclassification rate and AUC of 0.935. Perhaps

unsurprisingly, given that these are slightly more sophisticated methods, both the random forest and gradient boosting performed significantly better than k-nearest neighbors and regularized logistic regression, having misclassification rates of 7.80% and 8.82% and AUC values of 0.92 and 0.91, respectively. We produce below the out of sample confusion matrices of each of these models below.

	Random Forest		**Gradient Boosting**		**k-Nearest Neighbors**		**Logistic Regression**	
Label	**User**	**Bot**	**User**	**Bot**	**User**	**Bot**	**User**	**Bot**
User	176	11	176	11	177	10	169	18
Bot	13	197	15	195	21	189	17	193

From this, we can see that all models besides k-nearest neighbors, misclassifications are fairly evenly distributed across false positives and false negatives, while k-nearest neighbors seems to produce false negatives more often -- i.e., it more frequently reports bots as legitimate users than it does legitimate users as bots.

Discussion of results

While a misclassification rate of 6% is not perfect, it is still significant enough to be a useful result. If given a Twitter user with a suspiciously high number of followers, one could extract features from their followers and run one of the models discussed on each of them with an expectation that 94% of the predictions are correct. Taking action solely based off of the result of the algorithm -- i.e., banning users that are predicted to be bots may be ill-advised given that a legitimate user may be banned as a result. Results from the algorithm, however, could be used in an automated system to flag users for manual inspection. In some cases though, this isn't possible. Consider the possibility that Twitter wants to execute a mass ban of illegitimate users. Given the rate and scale at which bot accounts can be made, it may be unreasonable to expect any sort of manual inspection to occur, and, as such, basing decisions off of a model like those discussed in this report may be necessary.

There is a possibility that the use of more data in training may have improved the performance of these models, but a lack of computational power (specifically in computing the percentage of similar tweets a user has posted), restricted us from testing this. However, it is important to note that the reported misclassification rates are limited to the given dataset. While a relatively low out of sample misclassification rate indicates that these models were somewhat effective in generalizing properties of the dataset, it makes no guarantee that this generalization extends to other data. This is illustrated in the reported variable importance of our random forest model, in which the features, ordered by Gini importance, are the number of accounts the user is following, the number of followers the user has, the percentage of a user's tweets that contain links, the percentage of a user's tweets that are similar to another of their tweets, and finally the

average number of “@” mentions present in the user’s tweets. The first of these, namely, the number of accounts the user is following, was reported to be significantly more important than the rest. This makes some sense -- the average number of accounts a bot in this dataset is following is 8476, as opposed to 723 for legitimate users. An entity that provides fake follower services could thus defeat this classifier by limiting the number of users that a particular one of their bot accounts follows, and just make more bots to achieve the same follower numbers. In summary, further work can be done, and further work is needed to provide a more applicable solution for fake follower and bot detection.