Logistic Regression - Vowpal Wabbit

By: Oseas Ayerdi and Wendy De La Rosa

Introduction

A collection of 1,240 tweets from Trump's Twitter account were formatted and splitted into both a training- and test-datasets; the former of these two was used to train a logistic regression model via Vowpal Wabbit, `vw`, that categorizes future tweets as either written by Trump, or by his staff, and achieved an AUC value of 0.9260647 on the test dataset. How the original dataset was edited—and the logistic regression model calibrated in `vw`—is presented below.

Method

Benchmark

The original dataset was formatted to run through `vw` by restructuring it in the required format, consisting of the label content, namespaces, and features. We started with a simple model consisting of two *namespaces*, mainly: `tweet_content` and `time_info`. `Tweet_content` had as a separate feature with a default value of 1 for every unique string of characters in the tweet separated by a space, after removing any ':' in order to avoid errors with `vw`. The second namespace, `time_info` contains the hour of the day the tweet was published in the form of a string, each instance with a default value of 1.

The formatted dataset was shuffled and splitted into a training and test set, via the following commands:

```
cat trump_data.tsv |  shuff > permuted_trump_data.tsv
cat permuted_trump_data.tsv | python3 vw_format_simple.py >
vw_data_simple.txt
split -l 310 vw_data_simple.txt
```

(this line spits out four different files: xaa, xab, xac, and xad)

```
cat xaa, xab, xac > training_data_simple.txt
cat xad > test_data_simple.txt
```

We chose to shuffle the original data in order to make the training dataset more heterogeneous in content, and in turn, the resulting model more generalizable. If we had split it chronologically, the training dataset could miss important behavior for specific situations. For example, say the original data was split chronologically, with all tweets happening before Jan. 27th, 2017 belonging to the training dataset. In this case, there would be small-to-no training content related to the administration's use of Twitter to the backlash caused by Executive Order 13769 (signed on Jan. 27th, 2017), or the first travel ban. By extension, the model would be less prepared to anticipate their respective behaviors on Twitter in similar situations in the future.

The resulting training data (containing 80% of the original dataset's content) from the methodology above was used to construct the `predictor_simple.vw` model, via the following line:

```
vw -d training_data_simple.txt -f predictor_simple.vw --loss function
logistic
```

The predicted values for both the training and test sets were analyzed in R, and their respective AUC values were 0.994507 and 0.8900334. We used these figures as a way to determine if additional features were to be part of our final model or not.

Final Model

Our final formatted dataset consists of 3 namespaces: `tweet_content`, `time_info`, and `ratio`, in addition to the features `length`, `exclamation_count`, and `boolean`. The first namespace,