Biased vs. Random Sampling for Abusive Language Detection

Dante Razo
Indiana University, Bloomington, IN
Department of Computational Linguistics
drazo@indiana.edu

Abstract

I can't say for certain whether boosting improves testing accuracy and/or makes data more explicit or implicit. It does, however, drastically reduce the size of the Kaggle dataset to anywhere between 10-20% of the given sample size. For the most representative models, you want as much data as you can get.

1 Introduction

My research question this semester was a convoluted one. Originally, I asked "why would we want to use biased datasets when boosted random sampling nets better results?" I was referring to test accuracy and F1 scores specifically.

I rewrote it to be "Why does boosted random sampling give better results than biased?" To answer this, I set up an experiment that resamples the massive Kaggle dataset with either boosted or random sampling. The plan was to compare results and determine whether the data can be made more explicit or implicit depending on the sampling used.

2 Sampling Experiment

For this experiment I used two kinds of sampling on the Kaggle dataset: boosted sampling and random sampling. The *target* column contains a **float** value that measures how toxic a message is. According to the documentation, messages with *target* ≥ 0.5 are considered abusive. I left this as an easy-to-change parameter called *kaggle_threshold* so that I can test different values in the future.

2.1 Data

The Kaggle dataset contains 1,804,874 points of 45-dimensional data.

2.1.1 Data Preprocessing

I removed the ID column and all categorical data columns from the dataset, leaving me with 7 columns of numerical data. Dimension reduction made working with the data *much* faster on my laptop.

2.1.2 Boosting

The goal was to create a list of keywords for a topic, and filter comments based on whether that have that word or not. I didn't implement this yet though it should be easy to do so given enough time.

Instead, I boosted on the lexicon of abusive words featured in the NAACL-2018 paper "Inducing a Lexicon of Abusive Words – A Feature-Based Approach" by Michael Wiegand, Josef Ruppenhofer, Anna Schmidt and Clayton Greenberg. This left me with a considerably smaller dataset than what I put in — with a sample size of 10000, the boosted set shrunk to 13% of that.

2.2 What Went Wrong

The most egregious mistake was filtering on whether comments contained hate speech vs. filtering on a topic, as described in the previous section. I misunderstood the goal.

Also, I trained my original Kaggle SVM on the wrong dataset, so I wasn't able to build upon my code. I had to rewrite the data import from scratch and unfortunately was not able to get it to work.

3 Results

The following results are taken from the current experiment and past experiments on the Kaggle dataset and related datasets. Values that have yet to be computed are marked as "**TBD**".

Using a linear kernel **SVM**, **word** analyzer, and **ngram range** of [1,10]:

Sampling	Test Accuracy
Boosted	**TBD**
Random	**TBD**

Compare these to old Kaggle SVM (linear **kernel**, **ngram range**=[1,10]) results (trained on the wrong dataset but trained nonetheless):

analyzer	Accuracy
Word	0.784221427226511
Char	**TBD**

Finally, some results from the other datasets described in Wiegand's paper with different models (RF being Random Forest, and RF+GS being Random Forest with GridSearch):

Dataset	SVM	RF	RF+GS
Founta	0.9394993045897079	0.9047983310152990	0.9037552155771905
Kumar	0.6826262626262626	0.675959595959596	0.6953535353535354
Razavi	**TBD**	**TBD**	**TBD**
Warner	**TBD**	**TBD**	**TBD**

4 Conclusion