Detecting Social Bots on Twitter: An Anytime Active Learning Approach [*]

Maria E. Ramirez-Loaiza
mramire8@hawk.iit.edu
Illinois Institute of Technology
Chicago, IL 60616

Abstract

Abstract here.

1 Introduction

Outline: Active learning

Outline: Use in social networks

Outline: tweeter classficiation task

Outline: challenge

Outline: proposed method

Outline: challenges

Outline: experiments

Outline: research questions

2 Methodology

In this section, we first review standard active learning and then formulate our proposed method.

2.1 Problem Formulation

Let $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be a labeled dataset where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector that represents a Twitter user timeline and $y_i \in \{y^0 = \text{human}, y^1 = \text{bot}\}$ is its class label. Let $\mathcal{U} = \{\mathbf{x}_i\}_{i=l+1}^{m}$ be a set of unlabeled examples. Let $P_{\mathcal{L}}(y|\mathbf{x})$ be the conditional probability of y given $\mathbf{x}$ according to a classifier trained on $\mathcal{L}$.

Typical pool-based active learning selects instances $\mathcal{U}^* \subseteq \mathcal{U}$ to be labeled by a human annotator (*oracle*) and appended to $\mathcal{L}$. Assuming a prespecified annotation budget B and an annotation cost function $C(\mathbf{x})$, the goal of the active learning algorithm (*student*) is to select $\mathcal{U}^*$ to minimize the classifier's generalization error subject to the budget constraints:

$$\mathcal{U}^* \leftarrow \underset{\mathcal{U}_i \subseteq \mathcal{U}}{\operatorname{argmin}} Err(P_{\mathcal{L} \,\cup\, \mathcal{U}_i}(y|\mathbf{x})) \quad \text{s.t.} \sum_{\mathbf{x}_j \in \mathcal{U}_i} C(\mathbf{x}_j) \leq B \qquad (1)$$

Equation 1 is typically optimized by greedy algorithms, selecting one or more examples at a time according to some heuristic criterion that estimates the utility of each labeled example. A common approach is to request a label for the unlabeled instance that maximizes benefit-cost ratio: $\mathbf{x}_i^* \leftarrow \arg\max_{\mathbf{x}_i \in \mathcal{U}} \frac{U(\mathbf{x}_i)}{C(\mathbf{x}_i)}$.

Various definitions of utility $U(\cdot)$ are used in the literature, such as expected error reduction Roy and McCallum (2001) and classifier uncertainty Lewis and Gale (1994). In this paper, we use uncertainty sampling for our formulation. More formally, uncertainty sampling queries the instances whose predicted posterior probability is the least confident, redefining Equation 1:

$$\mathbf{x}^* \leftarrow \underset{\mathbf{x}_i \in \mathcal{U}}{\arg\max} \left(1 - \max_{y \in Y} P_{\mathcal{L}}(\hat{y}|\mathbf{x}_i)\right) \qquad (2)$$

Equation 2 uses conditional error as a measure of confidence.

We propose the use of an anytime active learning technique to save annotation cost by controlling

[*] This report was developed as part of a course project for CS579-F14 and is part of the AAL research project at the Machine Learning Lab at IIT.

what the oracle sees Ramirez-Loaiza et al. (2014). The idea is that the student selects the most representative tweet of a user timeline and queries the oracle for the label. We define an alternative formulation of the active learning problem in which the student has the added capability of presenting one tweet the human oracle to request a label.

Let s_i^k be the k-th tweet in user $\mathbf{x}_i$ timeline. Each tweet is scored by a function $TS(\cdot)$. We build upon Equation 2 and incorporate the tweet selection into the student's objective:

$$\underset{s_i^k \in \mathbf{x}_i \in \mathcal{U}}{\arg\max} \left(1 - \max_{y \in Y} P_{\mathcal{L}}(\hat{y}|\mathbf{x}_i) \times TS(s_i^k)\right) \quad (3)$$

Intuitively, we want to select a tweet that allows the oracle to determine if the author of the tweet is human or not. Also, we want that tweet to be likely to be labeled since we optimize the budget simultaneously. A simple way to define $TS(\cdot)$ that selects the best tweet for annotation, is to select the tweet that a tweet classifier is most confident about. Thus we define $TweetScore(\cdot)$ as follows:

$$\underset{s_i^k \in \mathbf{x}_i \in \mathcal{U}}{\arg\max} \left(1 - \max_{y \in Y} P_{\mathcal{L}}(\hat{y}|\mathbf{x}_i) \times P_T(y_{max}|s_i^k)\right) \quad (4)$$

where P_T is a probabilistic classifier that determines how likely is an individual tweet to be bot or human generated. In practice, we use tweets $s_i^k \in \mathbf{x}_i \in \mathcal{L}$ to train the tweet classifier.

3 Experimental Evaluation

3.1 Data Collection

Our experiments uses data collected from Twitter based on the **Social Honeypots** dataset Lee et al. (2011). We use a subset of a collection of known legitimate and bot account user names, to build our own dataset. We collected the most recent timeline of each user up to 200 tweets. We collected 883 legitimate user and 898 bot accounts.

The data was further filtered by removing accounts whose last activity was older than 2014, eliminating 45 and 173 accounts from legitimate and bots respectively. For our active learning experiments, we report average of five trials on a train-test split.

3.2 Data Preprocessing

Each tweet was tokenized by words, ignoring punctuation, collapsing URLs and mentions. Every token was converted to lower case and stemmed using a Porter stemmer. The resulting tokens were used to form a TF-IDF feature vector using unigrams. To reduce the size of the dictionary terms that appear less than five times are removed. The final dictionary size is of 13399 features. Table 1 characteristics of the representation.

Table 1: Collected data from Twitter per type of user

Class	N. Users	Avg. Tweets
Human	838	199
Bots	725	196

3.3 Simulations

oracle

student

4 Results and Discussion

4.1 Classification Methods

4.2 Data Representation

4.3 Classification of Individual Tweets

4.4 Bootstrap Effect

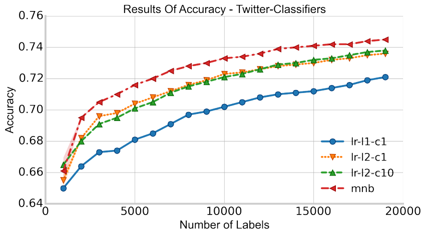
4.5 calibration effect

5 Related Work

The current work on social bot detection typically uses network analysis techniques to construct complex features and induce a probabilistic classifier. proposed a classifier trained on [[list the features]]

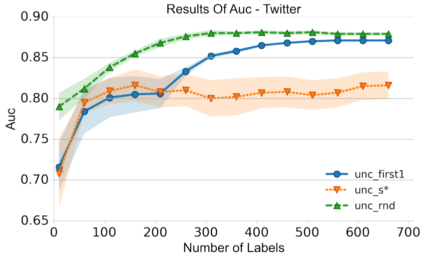
ME: add more related work

The described methods on previous studies rely on available datasets, which are expensive to build. To the best of our knowledge, our work is the first attempt to use active learning to build a social bot classifier. Furthermore, it is the first attempt to use an anytime active learning approach to address the task.



(a) IMDB

Figure 1: Classification models induced on tweets.



(a) IMDB

Figure 2: Active Learning.

6 Conclusions

We proposed an active learning method to detect bot operated twitter accounts using a tweet from the users timeline. Identifying the automated nature of an account is a hard problem. We discuss possible alternative to address the tweet selection method.

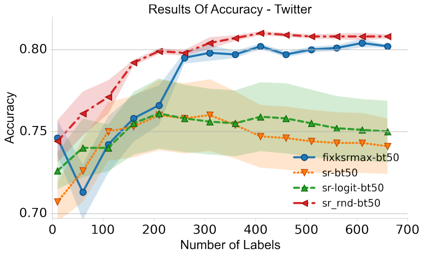
References

Kyumin Lee, BD Eoff, and James Caverlee. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

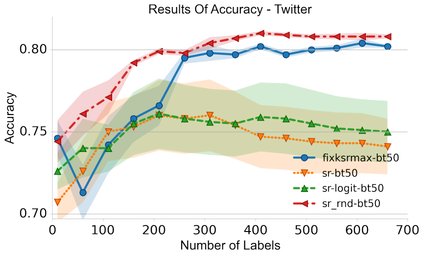
Maria E. Ramirez-Loaiza, Aron Culotta, and Mustafa Bilgic. Anytime Active Learning. In *AAAI Conference on Artificial Intelligence*, 2014.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, pages 441–448, 2001.



(a) IMDB

Figure 3: Active Learning.



(a) IMDB

Figure 4: [[Active Learning.]]