Contradict the Machine: A Hybrid Approach to Identifying Unknown Unknowns

Extended Abstract

Colin Vandenhof
University of Waterloo
Waterloo, Ontario
cm5vande@uwaterloo.ca

Edith Law
University of Waterloo
Waterloo, Ontario
edith.law@uwaterloo.ca

ABSTRACT

Machine predictions that are highly confident yet incorrect, i.e. unknown unknowns, are crucial errors to identify, especially in high-stakes settings like medicine or law. We describe a hybrid approach to identifying unknown unknowns that combines the previous algorithmic and crowdsourcing strategies. Our method uses a set of decision rules to approximate how the model makes high confidence predictions. We present the rules to crowd workers, and challenge them to generate instances that contradict the rules. To select the most promising rule to next present to workers, we use a multi-armed bandit algorithm. We evaluate our method by conducting a user study on Amazon Mechanical Turk. Experimental results on three datasets indicate that our approach discovers unknown unknowns more efficiently than state-of-the-art baselines.

KEYWORDS

unknown unknowns; crowdsourcing; multi-armed bandits

ACM Reference Format:
Colin Vandenhof and Edith Law. 2019. Contradict the Machine: A Hybrid Approach to Identifying Unknown Unknowns. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Predictive models are becoming increasingly prevalent in real world applications, from facial recognition to fraud detection. With more sophisticated learning algorithms, the accuracy of these models has increased; however, it is often at the expense of transparency. An important challenge lies in characterizing the errors made by black-box models, whose inner workings are inaccessible or poorly understood.

A confidence score is typically used to convey the level of uncertainty of a model in its prediction. This signal can be misleading when instances are predicted incorrectly with high-confidence. These mistakes are called the unknown unknowns (UUs). Attenberg et al. observed that UUs are generally present as systematic errors in specific regions of the feature space [2, 3]. These "blind spots" generally occur due to discrepancies between the training and target data distributions (*dataset shift* [5]). The underlying source of this discrepancy might be, for example, uncorrected bias in the training data.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

Whatever their cause, UUs are critical errors to identify in deployed predictive models. Decision makers are likely to use the model's uncertainty as a basis for how much to trust a prediction; hence, any incorrect predictions that are made with high confidence can have serious consequences.

Two approaches currently exist for identifying UUs. The first is a crowdsourcing approach, in which candidates are proposed by workers [2, 3]. The second is an algorithmic approach, in which candidates are automatically selected from a fixed set of test instances [4, 7]. We devise a framework that combines both approaches. Experiments are performed with three datasets on Amazon Mechanical Turk, and results indicate that our hybrid approach achieves superior performance in terms of cumulative utility.

2 PROBLEM STATEMENT

For any instance $\boldsymbol{x} \in X$, the black-box model M provides a predicted label, $\hat{y} \in C$, where C is the set of classes, and a confidence score $s \in [0, 1]$. We define a UU as an instance that is predicted incorrectly, $\hat{y} \neq y$, with a confidence score above some threshold τ, $s > \tau$. For simplicity, we target UUs predicted to some critical class c, for which false positives are particularly costly and need to be identified.

We also have access to a set of (unlabelled) test instances, $X_{test} \subseteq X$. To obtain the label, the system can give a query to a worker. In particular, the system only queries those instances in $X_{cand} \subseteq X_{test}$ which are valid UU candidates:

$$X_{cand} = \{\boldsymbol{x} | \boldsymbol{x} \in X_{test}, \hat{y} = c, s > \tau\} \quad (1)$$

Given a query instance $\boldsymbol{x_i} \in X_{cand}$, we assume that the worker knows the label y_i and has three actions available:

(1) UU identification: if $y_i \neq c$, return `identify`, $(\boldsymbol{x_i}, y_i)$
(2) modification: if $y_i = c$, modify $\boldsymbol{x_i}$ to produce some new instance $\boldsymbol{x_j} \in X$ such that $y_j \neq c$, and return `modify`, $(\boldsymbol{x_j}, y_j)$
(3) rejection: if $y_i = c$, but the worker is unable to modify $\boldsymbol{x_i}$ to produce some new instance $\boldsymbol{x_j} \in X$ such that $y_j \neq c$, return `reject`, $(\boldsymbol{x_i}, y_i)$

We assign a fixed cost to each of the three actions, and give unit utility to the discovery of a UU. Hence, the utility function for a given query q_t is the following:

$$u(q_t) = disc(q_t) - cost(q_t) \quad (2)$$

Here, $disc(q_t)$ is a function that returns 1 if the query q_t resulted in a UU discovery and 0 otherwise. $cost(q_t)$ is the cost of the action performed by the worker for that query. The objective is to find the sequence of queries that maximizes the total utility, $\sum_{t=1}^{n} u(q_t)$.

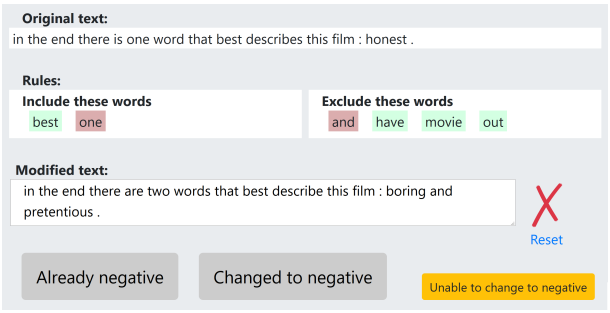

Figure 1: Crowdsourcing interface. In this case, the label has been successfully modified from positive (the critical class) to negative, but the rule is not yet satisfied.

3 METHODOLOGY

Our method proceeds in two phases. In the first phase, a set of decision rules is generated to approximate how the model makes high confidence predictions to the critical class. The second phase entails a crowdsourcing task in which workers are sequentially queried. The querying strategy is formulated as a multi-armed bandit.

3.1 Phase 1: Decision Rule Learning

The first phase aims to learn set of decision rules that distinguish instances predicted with high confidence to the critical class (class 1) from the rest (class 0). The left-hand side of the rule is a conjunction of predicates. Each predicate is of the form (attribute=value), where the attribute is an interpretable feature, and the value is 1 or 0 to indicate the presence or absence of that feature. For example, if the critical class is positive movie reviews, a rule might take the form (`best=1 AND bad=0 => 1`). To encourage interpretability, we favor short rules and set L_{max} to be the maximum number of predicates in any rule. To ensure decomposability, the rules must be non-overlapping, such that each instance is covered by at most one rule. These rules can be efficiently generated by adapting the Classification And Regression Tree (CART) algorithm [6].

3.2 Phase 2: Contradict the Machine

In the second phase, we use the decision rules to search for UUs via a crowdsourcing task that we call Contradict the Machine (CTM). For this task, the worker is given an instance from X_{cand} that is covered by a decision rule. If the label is not the critical class, it is confirmed to be a UU and the worker returns the instance (`identify` action). Otherwise, the worker is challenged to modify the instance such that its label changes, while ensuring that it is still covered by the rule (`modify` action). The result is a contradictory example — one that according to the decision rule, should be confidently predicted to the critical class, yet whose label is *not* the critical class. The worker then returns the modified instance. If the worker is somehow unable to generate such an example, a third `reject` action is available (see Figure 1).

To select which rule and instance to present to the worker, the rules are treated as arms of a multi-armed bandit. At each step, we query the rule with the highest expected utility, and present the worker with a random candidate instance that is covered by that rule. To compute expected utility, we maintain statistics on the frequency of `reject` actions, the posterior probability that an instance covered by rule R will already be a UU (i.e. `identify` action), and the posterior probability that a modified instance covered by rule R will be predicted to the critical class with high confidence (i.e. `modify` action resulting in a UU). The posteriors are represented with Beta distributions. Thompson sampling is used to trade off exploitation of the most promising rules with exploration [10].

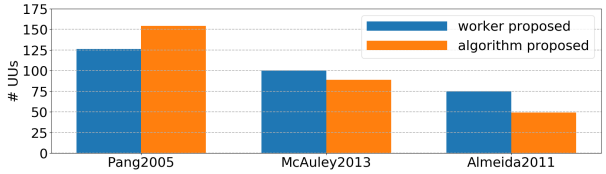

Figure 2: UUs proposed by algorithm vs. worker using CTM.

4 EXPERIMENTAL RESULTS

We performed experiments on Amazon Mechanical Turk with three datasets: Pang2005, comprised of 10k movie reviews from Rotten Tomatoes classified as negative or positive [9]; McAuley2013, containing 500k reviews from the Amazon Fine Food Store classified as low (1-2 stars) or high (4-5 stars) [8]; and Almeida2011, comprised of 5k SMS messages classified as spam or non-spam [1]. A logistic regression classifier was used as the black-box model, trained on a bag-of-words representation of the text. We induced bias in the training data to ensure that there were sufficient UUs to be discovered by (1) clustering the training data and discarding data from an arbitrary cluster and (2) class balancing.

Our method was evaluated against several baselines. Our primary comparison was with UUB, the algorithmic approach proposed by Lakkaraju et al. [7]. We also tested a variant of CTM that does not present the worker with any rule to satisfy (CTM-NoRule) and a variant that randomly selects which rule to present to workers instead of the bandit algorithm (CTM-Random).

We find that CTM outperforms UUB in terms of cumulative utility, achieving gains of 67.5% 32.1%, and 68.5% on Pang2005, McAuley2013, and Almeida2011 respectively. No large difference in utility is observed between querying strategies (CTM vs. CTM-Random). Comparison of CTM with CTM-NoRule suggests that rules vastly improve performance, with the exception of Pang2005. This finding is not surprising because the Pang2005 rule set has relatively poor precision (60.7%). Figure 2 shows the quantity of UUs discovered from the test set (i.e. algorithm proposed) and UUs generated by the workers (i.e. worker proposed). Both contributions are substantial, showing the merit of a hybrid approach.

5 CONCLUSION

Two general directions of research have emerged for identifying UUs of predictive models: crowdsourcing and algorithms. We describe a promising new framework that combines both approaches. Possible areas of future work include adapting CTM to other types of data and incorporating mechanisms to take advantage of worker expertise.

REFERENCES

[1] Tiago A. Almeida, José María Gómez Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. In *Proceedings of the 2011 ACM Symposium on Document Engineering, Mountain View, CA, USA, September 19-22, 2011*. 259–262. https://doi.org/10.1145/2034691.2034742

[2] Joshua Attenberg, Panos Ipeirotis, and Foster J. Provost. 2015. Beat the Machine: Challenging Humans to Find a Predictive Model's "Unknown Unknowns". *J. Data and Information Quality* 6, 1 (2015), 1:1–1:17. https://doi.org/10.1145/2700832

[3] Josh Attenberg, Panagiotis G. Ipeirotis, and Foster J. Provost. 2011. Beat the Machine: Challenging Workers to Find the Unknown Unknowns. In *Human Computation, Papers from the 2011 AAAI Workshop*.

[4] Gagan Bansal and Daniel S. Weld. 2018. A Coverage-Based Utility Model for Identifying Unknown Unknowns. In *Proc. of AAAI*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17110

[5] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of Representations for Domain Adaptation. In *Proc. of NIPS*. 137–144. http://papers.nips.cc/paper/2983-analysis-of-representations-for-domain-adaptation

[6] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Wadsworth.

[7] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. 2017. Identifying Unknown Unknowns in the Open World: Representations and Policies for Guided Exploration. In *Proc. of AAAI*. 2124–2132. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14434

[8] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*. 897–908. https://doi.org/10.1145/2488388.2488466

[9] Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proc. of ACL*. 115–124. http://aclweb.org/anthology/P/P05/P05-1015.pdf

[10] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.