Discovering Unknown Unknowns of Predictive Models

Himabindu Lakkaraju
Stanford University
himalv@cs.stanford.edu

Ece Kamar
Microsoft Research
eckamar@microsoft.com

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

Eric Horvitz
Microsoft Research
horvitz@microsoft.com

1 Introduction

Predictive models are widely used in domains ranging from judiciary and healthcare to autonomous driving. As we increasingly rely on these models for high-stakes decisions, identifying and characterizing their unexpected failures in the real world is critical. We categorize errors of a predictive model as: *known unknowns* and *unknown unknowns* [3]. Known unknowns are those data points for which the model makes low confidence predictions and errs, whereas unknown unknowns correspond to those points where the model is highly confident about its predictions, but is actually wrong. Since the model lacks awareness of such unknown unknowns, approaches developed for addressing known unknowns (e.g., active learning) cannot be used for discovering unknown unknowns. Unknown unknowns primarily occur when the data used for training a predictive model is not representative of the samples encountered during test time, i.e., when the model is deployed *in the wild*. This mismatch could be a result of biases in the collection of training data or differences between the train and test distributions due to temporal, spatial or other factors such as a subtle shift in task definition.

Here, we study the problem of informed discovery of unknown unknowns of any given predictive model where unknown unknowns occur due to systematic biases in the training data. Consequently, the resulting unknown unknowns are likely to be concentrated in certain portions of the feature space (and not randomly). We also assume that the features which are informative enough to distinguish between different kinds of unknown unknowns are present in the data, but the biases in the training data prevented the predictive model from learning that these features are indeed discriminative. We propose a methodology that guides the discovery of such unknown unknowns by querying true labels of selected instances from an oracle under a fixed budget that limits the number of queries. The formulation assumes no knowledge of the functional form or the associated training data of the predictive model and treats it as a black box which outputs a label and a confidence score (or a proxy) for a given data point. To the best of our knowledge, this is the first work providing an algorithmic solution to address this problem.

Our methodology follows a two-step approach which first partitions the search space such that instances which share similar feature values and have been assigned similar confidence scores by the black box model are grouped together. This step also associates an interpretable description with each such group. The second step then employs an explore-exploit strategy for navigating through these groups systematically based on the feedback from an oracle. We also present qualitative and quantitative evaluations of our framework on an image classification task.

2 Our Framework

Given a black-box predictive model $\mathcal{M}$ which takes as input a data point x with features $\mathcal{F} = \{f_1, f_2, \cdots f_L\}$, and returns a class label $c' \in C$ and a confidence score $s \in [0, 1]$, our goal is to find the unknown unknowns of $\mathcal{M}$ w.r.t a given test set $\mathcal{D}$ using a limited number of queries, B, to the oracle, and, more broadly, to maximize the utility associated with the discovered unknown

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

unknowns. The discovery process is guided by a utility function, which not only incentivizes the discovery of unknowns unknowns, but also accounts for the costs associated with querying the oracle (e.g., monetary and time costs of labeling in crowdsourcing).

Although our methodology is generic enough to find unknown unknowns associated with all the classes in the test set, we formulate the problem for a particular class c, a *critical class*, where false positives are costly and need to be discovered [5]. Based on the decisions of the system designer regarding critical class c and confidence threshold τ, our search space for unknown unknown discovery constitutes all of those data points in $\mathcal{D}$ which are assigned the critical class c by model $\mathcal{M}$ with confidence higher than τ .

Our approach takes the following inputs: 1) A set of N instances, $\mathcal{X} = \{x_1, x_2 \cdots x_N\} \subseteq \mathcal{D}$, which were *confidently* assigned to the critical class c by the model $\mathcal{M}$, and the corresponding confidence scores, $\mathcal{S} = \{s_1, s_2 \cdots s_N\}$, assigned to these points by $\mathcal{M}$, 2) An oracle o which takes as input a data point x and returns its true label $o(x)$ as well as the cost incurred to determine the true label of x, $cost(x)$ 3) A budget B on the number of times the oracle can be queried.

Our utility function, $u(x(t))$, for querying the label of data point $x(t)$ at the t^{th} step of exploration is defined as:

$$u(x(t)) = \mathbb{1}_{\{o(x_t) \neq c\}} - \gamma \times cost(x(t)) \tag{1}$$

where $\mathbb{1}_{\{o(x_t) \neq c\}}$ is an indicator function which returns 1 if $x(t)$ is identified as an unknown unknown, and a 0 otherwise. $cost(x(t)) \in [0, 1]$ is the cost incurred by the oracle to determine the label of $x(t)$. Both the indicator and the cost functions in Equation 1 are initially unknown and observed based on oracle's feedback on $x(t)$. $\gamma \in [0, 1]$ is a tradeoff parameter which can be provided by the end user.

Problem Statement: Find a sequence of B instances $\{x(1), x(2) \cdots x(B)\} \subseteq \mathcal{X}$ for which the cumulative utility $\sum_{t=1}^{B} u(x(t))$ is maximum.

Below, we present the two-phase approach that we propose to address this problem. First we present *Descriptive Space Partitioning* (DSP), which induces a similarity preserving partition on the set $\mathcal{X}$. Then, we present a novel multi-armed bandit algorithm, which we refer to as *Bandit for Unknown Unknowns* (UUB), for systematically *searching* for unknown unknowns within the resulting groups while leveraging feedback from an oracle.

Descriptive Space Partitioning The goal of the DSP is to partition the instances in $\mathcal{X}$ such that instances grouped together are likely to be *indistinguishable* w.r.t the model $\mathcal{M}$ and the feture set $\mathcal{F}$. DSP creates groups that can be explored by our bandit algorithm, UUB, to discover regions with high concentrations of unknown unknowns. In addition, each group is associated with a descriptive pattern (details below) which can help us understand what kinds of instances are part of that group.

DSP takes as input a set of candidate patterns $\mathcal{Q} = \{q_1, q_2, \cdots\}$ where each q_i is a conjunction of (feature, value) pairs. Such patterns can be obtained by running an off-the-shelf frequent pattern mining algorithm such as Apriori [2] on $\mathcal{X}$. Each pattern characterizes or *covers* a group of one or more instances in $\mathcal{X}$. For each pattern q, the set of instances that satisfy q is denoted by $covered\_by(q)$, the centroid of such instances is $\bar{x}_q$, and their mean confidence score is $\bar{s}_q$.

Our partitioning objective minimizes dissimilarities of instances within each group, maximizes them across groups, and favors concise descriptions for each group. In particular, we define *goodness* of each pattern q in $\mathcal{Q}$ using the following metrics, where d and d' are standard distance measures defined over feature vectors of instances and their confidence scores respectively:

$$\textbf{Intra-partition feature distance: } g_1(q) = \sum_{\{x \in \mathcal{X} \colon x \in covered\_by(q)\}} d(x, \bar{x}_q)$$

$$\textbf{Inter-partition feature distance: } g_2(q) = \sum_{\{x \in \mathcal{X} \colon x \in covered\_by(q)\}} \sum_{\{q' \in \mathcal{Q} \colon q' \neq q\}} d(x, \bar{x}_{q'})$$

$$\textbf{Intra-partition confidence score distance: } g_3(q) = \sum_{\{s_i \colon x_i \in \mathcal{X} \wedge x_i \in covered\_by(q)\}} d'(s_i, \bar{s}_q)$$

$$\textbf{Inter-partition confidence score distance: } g_4(q) = \sum_{\{s_i \colon x_i \in \mathcal{X} \wedge x_i \in covered\_by(q)\}} \sum_{\{q' \in \mathcal{Q} \colon q' \neq q\}} d'(s_i, \bar{s}_{q'})$$

Pattern Length: $g_5(q) = size(q)$, the number of (feature, value) pairs in pattern q

Given the sets $\mathcal{X}$, $\mathcal{S}$, a collection of patterns $\mathcal{Q}$, and weight vector λ used to combine g_1 through g_5, our goal is to find a set of patterns $\mathcal{P} \subseteq \mathcal{Q}$ such that it covers all the points in $\mathcal{X}$ and minimizes the following objective:

$$min \sum_{q \in Q} f_q \left(\lambda_1 g_1(q) - \lambda_2 g_2(q) + \lambda_3 g_3(q) - \lambda_4 g_4(q) + \lambda_5 g_5(q)\right) \tag{2}$$

$$\text{subject to} \sum_{q:\, x \in covered\_by(q)} f_q \geq 1 \; \forall x \in \mathcal{X}, \text{ where } f_q \in \{0,1\} \; \forall q \in \mathcal{Q}$$

This formulation is identical to that of a weighted set cover problem which is NP-hard [7]. We employ a strategy which greedily selects patterns with maximum coverage-to-weight ratio at each step (Refer Appendix [8] for a detailed pseudocode), thus resulting in a $ln\ N$ approximation guarantee [7]. This process is repeated until no instance in $\mathcal{X}$ is left uncovered. In case of instances in $\mathcal{X}$ which are covered by multiple patterns, ties are broken by assigning it to the group with the closest centroid.

Multi-armed Bandit for Unknown Unknowns The output of the first step of our approach, DSP, is a set of K groups $\mathcal{P} = \{p_1, p_2 \cdots p_K\}$ such that each group p_j clusters together data points that are *indistinguishable* w.r.t the model $\mathcal{M}$ and feature space $\mathcal{F}$. The partitioning, however, does not guarantee that data points in a group are identical and each group has an unobservable concentration of unknown unknown instances. The goal of the second step of our approach is to compute an exploration policy over these groups such that it maximizes the overall utility of the discovery of unknown unknowns.

We formalize this problem as a multi-armed bandit problem and propose an algorithm which decides what group should be queried at each step (Refer Appendix [8] for detailed pseudocode). In this formalization, each group p_j corresponds to an arm j of the bandit. At each step, the algorithm strategically picks a group (details below) and then randomly samples a data point from that group without replacement and queries its true label from the oracle. Since querying the data point reveals whether it is an unknown unknown, the point is excluded from future steps. In the first K steps, the algorithm samples a point from each group. Then, at each step t, the algorithm chooses a group which maximizes $\bar{u}_t(i) + b_t(i)$ and then samples a point randomly from it. $\bar{u}_t(i)$ denotes the empirical mean utility (reward) of the group i at time t, and $b_t(i)$ represents the uncertainty over the estimate of $\bar{u}_t(i)$.

Our problem setting has the characteristic that the expected utility of each arm is non-stationary; querying a data point from a group changes the concentration of unknown unknowns in the group and consequently changes the expected utility of that group in future steps. Therefore, stationary MAB algorithms such as UCB [4] are not suitable. A variant of UCB, *discounted UCB*, addresses the non-stationary settings and can be used as follows to compute $\bar{u}_t(i)$ and $b_t(i)$ [6].

$$\bar{u}_t(i) = \frac{1}{N_t(\vartheta_t^i, i)} \sum_{j=1}^{t} \vartheta_{j,t}^i \; u(x(j)) \; \mathbb{1}_{A_j = i}, \; b_t(i) = \sqrt{\frac{2 \; log \sum_{i=1}^{K} N_t(\vartheta_t^i, i)}{N_t(\vartheta_t^i, i)}}, N_t(\vartheta_t^i, i) = \sum_{j=1}^{t} \vartheta_{j,t}^i \; \mathbb{1}_{A_j = i}$$

The main idea of Discounted UCB is weighing recent observations more to account for the non-stationary nature of the utility function. If $\vartheta_{j,t}^i$ denotes the discounting factor applied at time t to the reward obtained from arm i at time $j < t$, $\vartheta_{j,t}^i = \gamma^{t-j}$ in the case of discounted UCB, where $\gamma \in (0, 1)$.

The discounting factor of Discounting UCB is designed to handle arbitrary changes in the utility distribution, whereas the way the utility of a group changes in our setting has a certain structure: The utility estimate of arm i only changes when the arm is queried and the magnitude of the change corresponds to the influence of a single data point in the size of the group, $\mathcal{N}_i$. Using this observation, we can customize the calculation of $\vartheta_{j,t}^i$:

$$\vartheta_{j,t}^i = (\mathcal{N}_i - \sum_{l=1}^{t} \mathbb{1}_{A_l = i}) \; / \; (\mathcal{N}_i - \sum_{l=1}^{j} \mathbb{1}_{A_l = i}) \tag{3}$$

The value of $\vartheta_{j,t}^i$ is inversely proportional to the number of pulls of arm i during the interval (j, t). $\vartheta_{j,t}^i$ is 1, if the arm is not pulled during this interval, indicating that the expected utility of i remained unchanged.

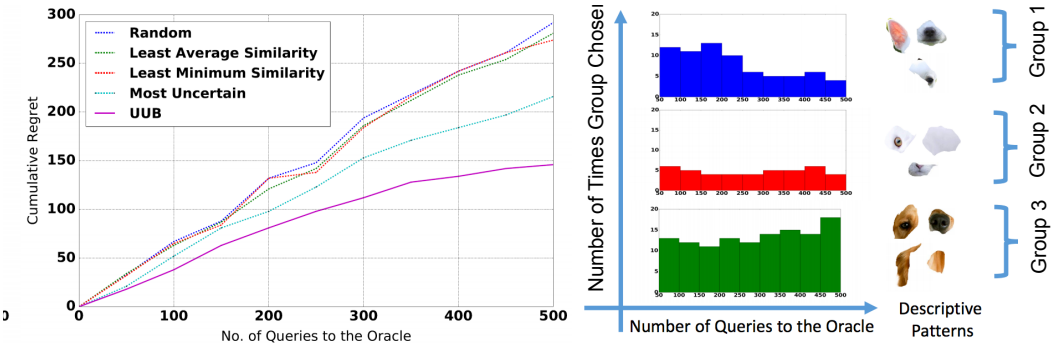

Figure 1: (a) Quantitative evaluation of our framework on image data (b) Illustration of our methodology on image data.

3 Experimental Evaluation

We evaluate our framework[1] on a set of 25K cat and dog images [1]. We use this data set to assess whether our framework can recognize unknown unknowns that occur when semantically meaningful sub-groups are missing from the training data. To this end, we split the data equally into train and test and bias the training data such that it comprises only of images of dogs which are black, and cats which are not black. We set the class label *cat* to be the critical class in our experiments. We use super-pixels obtained from the first max-pooling layer of Google's pre-trained Inception neural network [10, 9] as the feature set. Each image is represented with a feature vector comprising of 1's and 0's indicating the presence or absence of the corresponding super pixel. We experiment with multiple predictive models: decision trees, SVMs, logistic regression, random forests and neural network. Due to space constraints, we present results for decision trees as model $\mathcal{M}$ but detailed results for all the other models are included in the Appendix [8]. We set confidence threshold τ to 0.65 to construct our search space $\mathcal{X}$. The cost is 1 for all instances (uniform cost) and the budget B is set to 20% of all the instances of the set $\mathcal{X}$ through out our experiments. We search the parameter space using coordinate descent to find parameters which result in the minimum value of the objective function defined in Eqn. 2. Further, the results presented for UUB are all averaged across 100 runs.

Quantitative Analysis We compare the performance of our complete pipeline (DSP + UUB) is to other end-to-end heuristic methods we devised as baselines (Refer Appendix [8] for a comprehensive evaluation of each phase of the pipeline). We compare the cumulative regret of our framework to several baselines: 1) Random sampling: Randomly select B instances from set $\mathcal{X}$ for querying the oracle. 2) Least average similarity: For each instance in $\mathcal{X}$, compute the average Euclidean distance w.r.t all the data points in the training set and choose B instances with the largest distance. The idea is that instances dissimilar to those encountered during the training of the model could potentially be unknown unknowns. 3) Least maximum similarity: Compute minimum Euclidean distance of each instance in $\mathcal{X}$ from the training set and choose B instances with the highest distances. 4) Most uncertain: Rank the instances in $\mathcal{X}$ in increasing order of the confidence scores assigned by the model $\mathcal{M}$ and pick the top B instances. Figure 1(a) shows the cumulative regret of our framework and the baselines for the image data. Our framework achieves the least cumulative regret of all the strategies.

Qualitative Analysis Figure 1(b) presents an illustrative example of how our methodology explores three of the groups generated for the image data set. Our partitioning framework associated the super pixels shown in the Figure with each group. Examining the super pixels reveals that groups 1, 2 and 3 correspond to the images of white chihuahuas (dog), white cats, and brown dogs respectively. The plot shows the number of times these groups have been *played* by our bandit algorithm. The figure shows that group 2 is selected very few times compared to groups 1 and 3 — because white cat images are part of the training data for our predictive models and there are not many unknown unknowns in this group. On the other hand, white and brown dogs are not part of the training data and the algorithm explores these groups often. Figure 1(c) also indicates that group 1 was explored often during the initial steps but not later on. This is because there were fewer unknown unknowns in that group and the algorithm had exhausted all of them after a certain number of plays.

[1] Detailed experiments on other tasks such as sentiment detection, subjectivity analysis, domain adaption are in Appendix [8]

References

[1] Dogs vs cats dataset. https://www.kaggle.com/c/dogs-vs-cats/data, 2013.

[2] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *VLDB*, 1994.

[3] J. Attenberg, P. Ipeirotis, and F. Provost. Beat the machine: Challenging humans to find a predictive model's “unknown unknowns”. *J. Data and Information Quality*, 6(1):1:1–1:17, Mar. 2015.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[5] C. Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

[6] A. Garivier and E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

[7] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of computer and system sciences*, 9(3):256–278, 1974.

[8] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz. Discovering blind spots of predictive models: Representations and policies for guided exploration. https://arxiv.org/abs/1610.09064, 2016.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?": Explaining the predictions of any classifier. *arXiv preprint arXiv:1602.04938*, 2016.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.