Identifying Unknown Unknowns in the Open World: Representations and Policies for Guided Exploration

Himabindu Lakkaraju,* **Ece Kamar,+** **Rich Caruana,+** **Eric Horvitz+**
*Stanford University, +Microsoft Research
*himalv@cs.stanford.edu, + {eckamar, rcaruana, horvitz}@microsoft.com

Abstract

Predictive models deployed in the real world may assign incorrect labels to instances with high confidence. Such errors or *unknown unknowns* are rooted in model incompleteness, and typically arise because of the mismatch between training data and the cases encountered at test time. As the models are blind to such errors, input from an oracle is needed to identify these failures. In this paper, we formulate and address 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. We propose a model-agnostic methodology which uses feedback from an oracle to both identify unknown unknowns and to intelligently guide the discovery. We employ a two-phase approach which first organizes the data into multiple partitions based on the feature similarity of instances and the confidence scores assigned by the predictive model, and then utilizes an explore-exploit strategy for discovering unknown unknowns across these partitions. We demonstrate the efficacy of our framework by varying the underlying causes of unknown unknowns across various applications. To the best of our knowledge, this paper presents the first algorithmic approach to the problem of discovering unknown unknowns of predictive models.

Introduction

Predictive models are widely employed in a variety of domains ranging from judiciary and health care to autonomous driving. As we increasingly rely on these models for high-stakes decisions, identifying and characterizing their unexpected failures in the open world is critical. We categorize errors of a predictive model as: *known unknowns* and *unknown unknowns* (Attenberg, Ipeirotis, and Provost 2015). Known unknowns are those data points for which the model makes low confidence predictions and errs. On the other hand, unknown unknowns correspond to those points for which the model is highly confident about its predictions but is actually wrong. Since the model lacks awareness of its unknown unknowns, approaches developed for addressing known unknowns (e.g., active learning (Settles 2009)) cannot be used for discovering unknown unknowns.

Unknown unknowns can arise when data that is used for training a predictive model is not representative of the samples encountered at test time when the model is deployed.

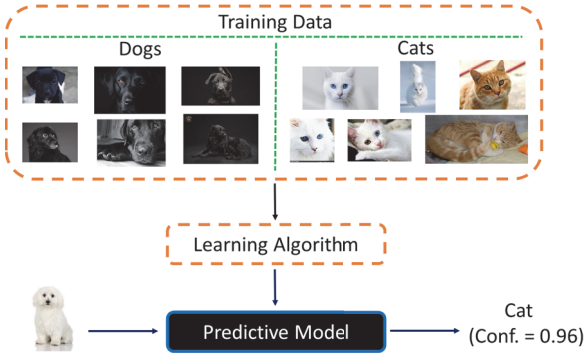


Figure 1: Unknown unknowns in an image classification task. Training data comprised only of images of black dogs and of white and brown cats. A predictive model trained on this data incorrectly labels a *white dog* (test image) as a *cat* with high confidence.

This mismatch could be a result of unmodeled 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. To illustrate, consider an image classification task where the goal is to predict if a given image corresponds to a cat or a dog (Figure 1). Let us assume that the training data is comprised of images of black dogs, and white and brown cats, and the feature set includes details such as nose shape, presence or absence of whiskers, color, and shape of the eyes. A predictive model trained on such data might learn to make predictions solely based on color despite the presence of other discriminative features because color can perfectly separate the two classes in the training data. However, during test time, such a model would classify an image of a white dog as a cat with high confidence. The images of white dogs are, therefore, unknown unknowns with regard to such a predictive model.

We formulate and address the problem of informed discovery of unknown unknowns of any given predictive model when deployed *in the wild*. More specifically, we seek to identify unknown unknowns which occur as a result of systematic biases in the training data. We formulate this as an optimization problem where unknown unknowns are discovered by querying an oracle for true labels of selected

instances under a fixed budget which limits the number of queries to the oracle. 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. These choices are motivated by real-world scenarios in domains such as healthcare and judiciary, where predictive models are being deployed in settings where end users have no access to either the model details or the associated training data (e.g., COMPAS risk assessment tool for sentencing (Brennan, Dieterich, and Ehret 2009)). Identifying the blind spots of predictive models in such high-stakes settings is critical as undetected unknown unknowns can be catastrophic. In criminal justice, biases and blindspots can lead to the inappropriate sentencing or incarceration of people charged with crimes or unintentional racial biases (Crawford 2016). To the best of our knowledge, this is the first work providing an algorithmic approach to addressing this problem.

Developing an algorithmic solution for the discovery of unknown unknowns introduces a number of challenges: 1) Since unknown unknowns can occur in any portion of the feature space, how do we develop strategies which can effectively and efficiently search the space? 2) As confidence scores associated with model predictions are typically not informative for identifying unknown unknowns, how can we make use of the feedback from an oracle to guide the discovery of unknown unknowns? 3) How can we effectively manage the trade-off between searching in neighborhoods where we previously found unknown unknowns and examining unexplored regions of the search space?

To address the problem at hand, we propose a two-step approach which first partitions the test data such that instances with similar feature values and confidence scores assigned by the predictive model are grouped together, and then employs an explore-exploit strategy for discovering unknown unknowns across these partitions based on the feedback from an oracle. The first step, which we refer to as *Descriptive Space Partitioning* (DSP), is guided by an objective function which encourages partitioning of the search space such that instances within each partition are maximally similar in terms of their feature values and confidence scores. DSP also provides interpretable explanations of the generated partitions by associating a comprehensible and compact description with each partition. As we later demonstrate in our experimental results, these interpretable explanations are very useful in understanding the properties of unknown unknowns discovered by our framework. We show that our objective is NP-hard and outline a greedy solution which is a $ln\ N$ approximation, where N is the number of data points in the search space. The second step of our methodology facilitates an effective *exploration* of the partitions generated by DSP while *exploiting* the feedback from an oracle. We propose a multi-armed bandit algorithm, *Bandit for Unknown Unknowns* (UUB), which exploits problem-specific characteristics to efficiently discover unknown unknowns.

The proposed methodology builds on the intuition that unknown unknowns occurring due to systematic biases are often concentrated in certain specific portions of the feature space and do not occur randomly (Attenberg, Ipeirotis, and Provost 2015). For instance, the example in Figure 1 illustrates a scenario where systematic biases in the training data caused the predictive model to wrongly infer color as the distinguishing feature. Consequently, images following a specific pattern (i.e., all of the images of white dogs) turn out to be unknown unknowns for the predictive model. Another key assumption that is crucial to the design of effective algorithmic solutions for the discovery of unknown unknowns is that available evidential features are informative enough to characterize different subsets of unknown unknowns. If such features were not available in the data, it would not be possible to leverage the properties of previously discovered unknown unknowns to find new ones. Consequently, learning algorithms designed to discover unknown unknowns would not be able to perform any better than blind search (no free lunch theorem (Wolpert and Macready 1997)).

We empirically evaluate the proposed framework on the task of discovering unknown unknowns occurring due to a variety of factors such as biased training data and domain adaptation across various diverse tasks, such as sentiment classification, subjectivity detection, and image classification. We experiment with a variety of base predictive models, ranging from decision trees to neural networks. The results demonstrate the effectiveness of the framework and its constituent components for the discovery of unknown unknowns across different experimental conditions, providing evidence that the method can be readily applied to discover unknown unknowns in different real-world settings.

Problem Formulation

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 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). Recall that, in this work, we focus on identifying unknown unknowns arising due to systematic biases in the training data. It is important to note that our formulation not only treats the predictive model as a black-box but also assumes no knowledge about the data used to train the predictive model.

Although our methodology is generic enough to find unknown unknowns associated with all the classes in the data, we formulate the problem for a particular class c, a *critical class*, where false positives are costly and need to be discovered (Elkan 2001). 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:

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

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.

Methodology

In this section, we present our two-step framework designed to address the problem of informed discovery of unknown unknowns which occur due to systematic biases in the training data. We begin this section by highlighting the assumptions required for our algorithmic solution to be effective:

1. Unknown unknowns arising due to biases in training data typically occur in certain specific portions of the feature space and not at random. For instance, in our image classification example, the systematic bias of not including white dog images in the training data resulted in a specific category of unknown unknowns which were all clumped together in the feature space and followed a specific pattern. Attenberg et. al. (Attenberg, Ipeirotis, and Provost 2015) observed this assumption to hold in practice and leveraged human intuition to find systematic patterns of unknown unknowns.
2. We also assume that the features available in the data can effectively characterize different kinds of unknown unknowns, but the biases in the training data prevented the predictive model from leveraging these discriminating features for prediction. If such features were not available in the data, it would not be possible to utilize the characteristics of previously discovered unknown unknowns to find new ones. Consequently, no learning algorithm would perform better than blind search if this assumption did not hold (no free lunch theorem (Wolpert and Macready 1997)).

Below we discuss our methodology in detail. 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 across these partitions while leveraging feedback from an oracle.

Descriptive Space Partitioning

Our approach exploits the aforementioned intuition that blind spots arising due to systematic biases in the data do not occur at random, but are instead concentrated in specific portions of the feature space. The first step of our approach, DSP, partitions the instances in $\mathcal{X}$ such that instances which are grouped together are similar to each other w.r.t the feature space $\mathcal{F}$ and were assigned similar confidence scores by the model $\mathcal{M}$. Partitioning $\mathcal{X}$ enables our bandit algorithm, UUB, to discover regions with high concentrations of unknown unknowns.

Algorithm 1 Greedy Algorithm for Partitioning

1: **Input:** Set of instances $\mathcal{X}$, Confidence scores $\mathcal{S}$, Patterns $\mathcal{Q}$, Metric functions $\{g_1 \cdots g_5\}$, Weights λ
2: **Procedure:**
3: $\mathcal{P} = \emptyset, \mathcal{E} = \mathcal{X}$
4: **while** $\mathcal{E} \neq \emptyset$ **do**:
5:
$$p = \arg\max_{q \in \mathcal{Q}} \frac{|\mathcal{E} \cap covered\_by(q)|}{g(q)}$$
where
$$g(q) = \lambda_1 g_1(q) - \lambda_2 g_2(q) + \lambda_3 g_3(q) - \lambda_4 g_4(q) + \lambda_5 g_5(q)$$
6: $\mathcal{P} = \mathcal{P} \cup p\,,\ \mathcal{Q} = \mathcal{Q} \setminus p\,,\ \mathcal{E} = \mathcal{E} \setminus covered\_by(p)$
7: **end while**
8: **return** $\mathcal{P}$

The intuition behind our partitioning approach is that two instances a and $a' \in \mathcal{X}$ are likely to be judged using a similar logic by model $\mathcal{M}$ if they share similar feature values and are assigned to the same class c with comparable confidence scores by $\mathcal{M}$. In such cases, if a is identified as an unknown unknown, a' is likely to be an unknown unknown as well[1]. Based on this intuition, we propose an objective function which encourages grouping of instances in $\mathcal{X}$ that are *similar* w.r.t the criteria outlined above, and facilitates separation of *dissimilar* instances. The proposed objective also associates a concise, comprehensible description with each partition, which is useful for understanding the exploration behavior of our framework and the kinds of unknown unknowns of $\mathcal{M}$ (details in the Experimental Evaluation Section).

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, operator, value) tuples where operator $\in \{=, \neq, \leq, <, \geq, >\}$. Such patterns can be obtained by running an off-the-shelf frequent pattern mining algorithm such as Apriori (Agrawal, Srikant, and others 1994) on $\mathcal{X}$. Each pattern *covers* a set of one or more instances in $\mathcal{S}$. For each pattern q, the set of instances that satisfy q is represented by $covered\_by(q)$, the centroid of such instances is denoted by $\bar{x}_q$, and their mean confidence score is $\bar{s}_q$.

The partitioning objective minimizes dissimilarities of instances within each partition and maximizes them across

[1]Note that this is not always the case, as we will see in the next section.

partitions. In particular, we define *goodness* of each pattern q in $\mathcal{Q}$ as the combination of following metrics, where d and d' are standard distance measures defined over feature vectors of instances and their confidence scores respectively:

Intra-partition feature distance:

$$g_1(q) = \sum_{\{x \in \mathcal{X}:\, x \in covered\_by(q)\}} d(x, \bar{x}_q)$$

Inter-partition feature distance:

$$g_2(q) = \sum_{\{x \in \mathcal{X}:\, x \in covered\_by(q)\}} \sum_{\{q' \in \mathcal{Q}:\, q' \neq q\}} d(x, \bar{x}_{q'})$$

Intra-partition confidence score distance:

$$g_3(q) = \sum_{\{s_i:\, x_i \in \mathcal{X} \wedge x_i \in covered\_by(q)\}} d'(s_i, \bar{s}_q)$$

Inter-partition confidence score distance:

$$g_4(q) = \sum_{\substack{\{s_i:\, x_i \in \mathcal{X} \wedge \\ x_i \in covered\_by(q)\}}} \sum_{\{q' \in \mathcal{Q}:\, q' \neq q\}} d'(s_i, \bar{s}_{q'})$$

Pattern Length: $g_5(q) = size(q)$, the number of (feature, operator, value) tuples in pattern q, included to favor concise descriptions.

Given the sets of instances $\mathcal{X}$, corresponding confidence scores $\mathcal{S}$, 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:

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

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

where f_q corresponds to an indicator variable associated with pattern q which determines if the pattern q has been added to the solution set ($f_q = 1$) or not ($f_q = 0$).

The aforementioned formulation is identical to that of a weighted set cover problem which is NP-hard (Johnson 1974). It has been shown that a greedy solution provides a $ln\ N$ approximation to the weighted set cover problem (Johnson 1974; Feige 1998) where N is the size of search space. Algorithm 1 applies a similar strategy which greedily selects patterns with maximum coverage-to-weight ratio at each step, thus resulting in a $ln\ N$ approximation guarantee. This process is repeated until no instance in $\mathcal{X}$ is left uncovered. If an instance in $\mathcal{X}$ is covered by multiple partitions, ties are broken by assigning it to a partition with the closest centroid.

Our partitioning approach is inspired by a class of clustering techniques commonly referred to as conceptual clustering (Michalski and Stepp 1983; Fisher 1987) or descriptive clustering (Weiss 2006; Li, Peng, and Wu 2008; Kim, Rudin, and Shah 2014; Lakkaraju and Leskovec 2016).

Algorithm 2 Explore-Exploit Algorithm for Unknown Unknowns

Input:
Set of partitions (arms) $\{1, 2 \cdots K\}$, Oracle o, Budget B
Procedure:
for t from 1 to B **do**:
  if $t \leq K$ **then**:
    Choose arm $A_t = t$
  else
    Choose arm $A_t = \arg\max_{1 \leq i \leq K} \bar{u}_t(i) + b_t(i)$
  end if
  Sample an instance $x(t)$ from partition p_{A_t} and query the oracle for its true label
  Observe true label of $x(t)$ and the cost of querying the oracle and compute $u(x(t))$ using Equation (1).
end for
return $\sum_{t=1}^{B} u(x(t))$

We make the following contributions to this line of research: We propose a novel objective function, whose components have not been jointly considered before. In contrast to previous solutions which employ post processing techniques or use Bayesian frameworks, we propose a simple, yet elegant solution which offers theoretical guarantees.

Multi-armed Bandit for Unknown Unknowns

The output of the first step of our approach, DSP, is a set of K partitions $\mathcal{P} = \{p_1, p_2 \cdots p_K\}$ such that each p_j corresponds to a set of data points which are similar w.r.t the feature space $\mathcal{F}$ and have been assigned similar confidence scores by the model $\mathcal{M}$. The partitioning scheme, however, does not guarantee that all data points in a partition share the same characteristic of being unknown unknown (or not being unknown unknown). It is important to note that sharing similar feature values and confidence scores does not ensure that the data points in a partition are indistinguishable as far as the model logic is concerned. This is due to the fact that the model $\mathcal{M}$ is a black-box and we do not actually observe the underlying functional forms and/or feature importance weights being used by $\mathcal{M}$. Consequently, each partition has an unobservable concentration of unknown unknown instances. The goal of the second step of our approach is to compute an exploration policy over the partitions generated by DSP such that it maximizes the cumulative utility of the discovery of unknown unknowns (as defined in the Problem Formulation section).

We formalize this problem as a multi-armed bandit problem and propose an algorithm for deciding which partition to query at each step (See Algorithm 2). In this formalization, each partition p_j corresponds to an arm j of the bandit. At each step, the algorithm chooses a partition and then randomly samples a data point from that partition 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 partition. Then, at each step t, the exploration decisions

are guided by a combination of $\bar{u}_t(i)$, the empirical mean utility (reward) of the partition i at time t, and $b_t(i)$, which 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 partition changes the concentration of unknown unknowns in the partition and consequently changes the expected utility of that partition in future steps. Therefore, stationary MAB algorithms such as UCB (Auer, Cesa-Bianchi, and Fischer 2002) 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 $b_t(i)$ (Garivier and Moulines 2008).

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

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

The main idea of discounted UCB is to weight 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 $\gamma \in (0, 1)$. Garivier et. al. established a lower bound on the regret in the presence of abrupt changes in the reward distributions of the arms and also showed that discounted UCB matches this lower bound upto a logarithmic factor (Garivier and Moulines 2008).

The discounting factor of discounted UCB is designed to handle arbitrary changes in the utility distribution, whereas the way the utility of a partition changes in our setting has a certain structure: The utility estimate of arm i only changes by a bounded quantity when the arm is queried. Using this observation, we can customize the calculation of $\vartheta_{j,t}^i$ for our setting and eliminate the need to set up the value of γ, which affects the quality of decisions made by discounted UCB. We compute $\vartheta_{j,t}^i$ as the ratio of the number of data points in the partition i at time j to the number of data points in the partition i at time t:

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

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 i is not pulled during this interval, indicating that the expected utility of i remained unchanged. We refer to the version of Algorithm 2 that uses the discounting factor specific to our setting (Eqn. 3) as Bandit for Unknown Unknowns (UUB).

Experimental Evaluation

We now present details of the experimental evaluation of constituent components of our framework as well as the entire pipeline.

Datasets and Nature of Biases: We evaluate the performance of our methodology across four different data sets in which the underlying cause of unknown unknowns vary from biases in training data to domain adaptation:
(1) *Sentiment Snippets:* A collection of 10K sentiment snippets/sentences expressing opinions on various movies (Pang and Lee 2005). Each snippet (sentence) corresponds to a data point and is labeled as positive or negative. We split the data equally into train and test sets. We then bias the training data by randomly removing *sub-groups* of negative snippets from it. We consider *positive sentiment* as the critical class for this data.
(2) *Subjectivity:* A set of 10K subjective and objective snippets extracted from Rotten Tomatoes webpages (Pang and Lee 2004). We consider the *objective* class in this dataset as the critical class, split the data equally into train and test sets, and introduce bias in the same way as described above.
(3) *Amazon Reviews:* A random sample of 50K reviews of books and electronics collected from Amazon (McAuley, Pandey, and Leskovec 2015). We use this data set to study unknown unknowns introduced by domain adaptation; we train the predictive models on the electronics reviews and then *test* them on the book reviews. Similar to the *sentiment snippets* data set, the *positive sentiment* is the critical class.
(4) *Image Data:* A set of 25K cat and dog images (Kaggle 2013). 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.
Experimental Setting: We use bag of words features to train the predictive models for all of our textual data sets. As the features for the images, we use super-pixels obtained using the standard algorithms (Ribeiro, Singh, and Guestrin 2016). Images are represented with a feature vector comprising of 1's and 0's indicating the presence or absence of the corresponding super pixels. We experiment with multiple predictive models: decision trees, SVMs, logistic regression, random forests and neural network. Due to space constraints, this section presents results for decision trees as model $\mathcal{M}$ but detailed results for all the other models are included in an online Appendix (Lakkaraju et al. 2016). We set the threshold for confidence scores τ to 0.65 to construct our search space $\mathcal{X}$ for each data set. We consider two settings for the cost function (refer Eqn. 1): The cost is set to 1 for all instances (uniform cost) in the image dataset and it is set to $[(\text{length}(x) - \text{minlength})/(\text{maxlength} - \text{minlength})]$ (variable cost) for all textual data. length(x) denotes the number of words in a snippet (or review) x; minlength and maxlength denote the minimum and maximum number of words in any given snippet (or review). Note that these cost functions are only available to the oracle. The tradeoff parameter γ is set to 0.2. The parameters of DSP $\{\lambda_1, \cdots \lambda_5\}$ are estimated by setting aside as a validation set 5% of the test instances assigned to the critical class by the predictive models. We search the parameter space using coordinate de-

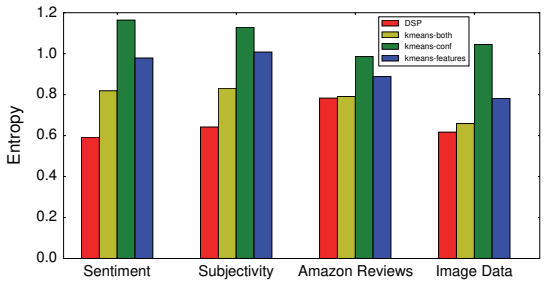


Figure 2: Evaluating partitioning strategies using entropy (smaller values are better).

scent to find parameters which result in the minimum value of the objective function defined in Eqn. 2. We set the budget B to 20% of all the instances in the set $\mathcal{X}$ through out our experiments. Further, the results presented for UUB are all averaged across 100 runs.

Evaluating the Partitioning Scheme

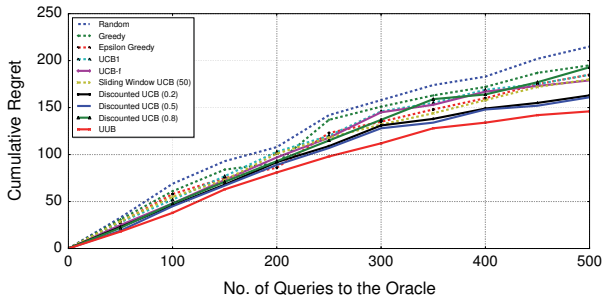
The effectiveness of our framework relies on the notion that our partitioning scheme, DSP, creates partitions such that unknown unknowns are concentrated in a specific subset of partitions as opposed to being evenly spread out across them. If unknown unknowns are distributed evenly across all the partitions, our bandit algorithm cannot perform better than a strategy which randomly chooses a partition at each step of the exploration process. We, therefore, measure the quality of partitions created by DSP by measuring the entropy of the distribution of unknown unknowns across the partitions in $\mathcal{P}$. For each partition $p \in \mathcal{P}$, we count the number of unknown unknowns, U_p based on the true labels which are only known to the oracle. We then compute entropy of $\mathcal{P}$ as follows:

$$\text{Entropy}(\mathcal{P}) = -\sum_{p \in \mathcal{P}} \frac{U_p}{\sum_{p' \in \mathcal{P}} U_{p'}} log_2(\frac{U_p}{\sum_{p' \in \mathcal{P}} U_{p'}})$$

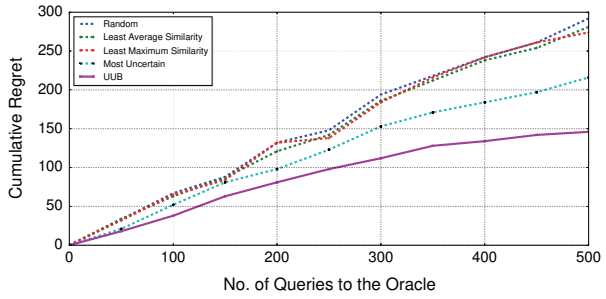
Smaller entropy values are desired as they indicate higher concentrations of unknown unknowns in fewer partitions.

Figure 2 compares the entropy of the partitions generated by DSP with clusters generated by k-means algorithms using only features in $\mathcal{F}$ (kmeans-features), only confidence scores in $\mathcal{S}$ (kmeans-conf) and both (kmeans-both) by first clustering using confidence scores and then using features. The entropy values for DSP are consistently smaller compared to alternative approaches using kmeans across all the datasets. This can be explained by the fact that DSP jointly optimizes inter and intra-partition distances over both features and confidence scores. As shown in Figure 2, the entropy values are much higher when k-means considers only features or only confidence scores indicating the importance of jointly reasoning about them.

We also compare the entropy values obtained for DSP as well as other k-means based approaches to an upper bound computed with random partitioning. For each of the algorithms (DSP and other k-means based approaches), we de-



(a)



(b)

Figure 3: (a) Evaluating the bandit framework on image data, (b) Evaluating the complete pipeline on image data (decision trees as predictive model).

signed a corresponding random partitioning scheme which randomly re-assigns all the data points in the set $\mathcal{X}$ to partitions while keeping the number of partitions and the number of data points within each partition same as that of the corresponding algorithm. We observe that the entropy values obtained for DSP and all the other baselines are consistently smaller than those of the corresponding random partitioning schemes. Also, the entropy values for DSP are about 32-37% lower compared to its random counterpart across all of the datasets.

Evaluating the Bandit Algorithm

We measure the performance of our multi-armed bandit algorithm UUB in terms of a standard evaluation metric in the MAB literature called *cumulative regret*. Cumulative regret of a policy π is computed as the difference between the total reward collected by an optimal policy π^*, which at each step plays the arm with the highest expected utility (or reward) and the total reward collected by the policy π. Small values of cumulative regret indicate better policies. The utility function defined in Eqn. 1 determines the reward associated with each instance.

We compare the performance of our algorithm, UUB, with that of several baseline algorithms such as random, greedy, ϵ-greedy strategies (Chapelle and Li 2011), UCB, UCB$_f$ (Slivkins and Upfal 2008), sliding window and discounted UCB (Garivier and Moulines 2008) for various values of the discounting factor $\gamma = \{0.2, 0.5, 0.8\}$. All algorithms take as input the partitions created by DSP. Figure

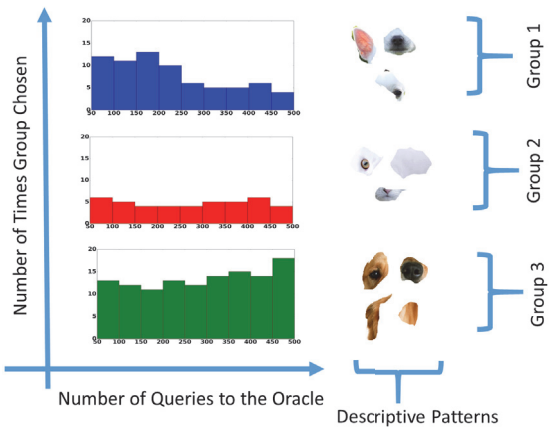


Figure 4: Illustration of the methodology on image data.

3(a) shows the cumulative regret of each of these algorithms on the image data set. Results for the other data sets can be seen in the Appendix (Lakkaraju et al. 2016). The figure shows that UUB achieves the smallest cumulative regret compared to other baselines on the image data set. Similarly, UUB is the best performing algorithm on the sentiment snippets and subjectivity snippets data sets, whereas discounted UCB ($\gamma = 0.5$) achieves slightly smaller regret than UUB on the Amazon reviews data set. The experiment also highlights a disadvantage of the discounted UCB algorithm as its performance is sensitive to the choice of the discounting factor γ, where as UUB is parameter free. Further, both UCB and its variant UCB$_f$ which are designed for stationary and slowly changing reward distributions respectively have higher cumulative regret than UUB and discounted UCB indicating that they are not as effective in our setting.

Evaluating the Overall Methodology

In the previous section, we compared the performance of UUB to other bandit methods when they are given the same data partitions to explore. In this section, we evaluate the performance of our complete pipeline (DSP + UUB). Due to the lack of existing baselines which address the problem at hand, we compare the performance of our framework to other end-to-end heuristic methods we devised as baselines. Due to space constraints, we present results only for the image dataset. Results for other data sets can be seen in the Appendix (Lakkaraju et al. 2016).

We compare the cumulative regret of our framework to that of a variety of 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. 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. The least average similarity and least maximum similarity baselines are related to research on outlier detection (Chandola, Banerjee, and Kumar 2007). Furthermore, the baseline titled *most uncertain* is similar to the uncertainty sampling query strategy used in active learning literature. Note that the least average similarity and the least maximum similarity baselines assume access to the data used to train the predictive model unlike our framework which makes no such assumptions. Figure 3(b) shows the cumulative regret of our framework and the baselines for the image data. It can be seen that UUB achieves the least cumulative regret of all the strategies across all data sets. It is interesting to note that the least average similarity and the least maximum similarity approaches perform worse than UUB in spite of having access to additional information in the form of training data.

Qualitative Analysis Figure 4 presents an illustrative example of how our methodology explores three of the partitions generated for the image data set. Our partitioning framework associated the super pixels shown in the Figure with each partition. Examining the super pixels reveals that partitions 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 the arms corresponding to these partitions have been played by our bandit algorithm. The figure shows that partition 2 is chosen fewer times compared to partitions 1 and 3 — because white cat images are part of the training data used by the predictive models and there are not many unknown unknowns in this partition. On the other hand, white and brown dogs are not part of the training data and our bandit algorithm explores these partitions often. Figure 4 also indicates that partition 1 was explored often during the initial plays but not later on. This is because there were fewer data points in that partition and the algorithm had exhausted all of them after a certain number of plays.

Related Work

In this section, we review prior research relevant to the discovery of unknown unknowns.

Unknown Unknowns The problem of model incompleteness and the challenge of grappling with unknown unknowns in the real world has been coming to the fore as a critical topic in discussions about the utility of AI technologies (Horvitz 2008). Attenberg et. al. introduced the idea of harnessing human input to identify unknown unknowns but their studies left the task of exploration and discovery completely to humans without any assistance (Attenberg, Ipeirotis, and Provost 2015). In contrast, we propose an algorithmic framework in which the role of the oracle is simpler and more realistic: The oracle is only queried for labels of selected instances chosen by our algorithmic framework.

Dataset Shift A common cause of unknown unknowns is *dataset shift*, which represents the mismatch between train-

ing and test distributions (Quionero-Candela et al. 2009; Jiang and Zhai 2007). Multiple approaches have been proposed to address dataset shift, including importance weighting of training instances based on similarity to test set (Shimodaira 2000), online learning of prediction models (Cesa-Bianchi and Lugosi 2006), and learning models robust to adversarial actions (Teo et al. 2007; Graepel and Herbrich 2004; Decoste and Schölkopf 2002). These approaches cannot be applied to our setting as they make one or more of the following assumptions which limit their applicability to real-world settings: 1) the model is not a black box 2) the data used to train the predictive model is accessible 3) the model can be adaptively retrained. Further, the goal of this work is different as we study the problem of discovering unknown unknowns of models which are already deployed.

Active Learning Active learning techniques aim to build highly accurate predictive models while requiring fewer labeled instances. These approaches typically involve querying an oracle for labels of certain selected instances and utilizing the obtained labels to adaptively retrain the predictive models (Settles 2009). Various query strategies have been proposed to choose the instances to be labeled (e.g., uncertainty sampling (Lewis and Gale 1994; Settles 2009), query by committee (Seung, Opper, and Sompolinsky 1992), expected model change (Settles, Craven, and Ray 2008), expected error reduction (Zhu, Lafferty, and Ghahramani 2003), expected variance reduction (Zhang and Oles 2000)). Active learning frameworks were designed to be employed during the learning phase of a predictive model and are therefore not readily applicable to our setting where the goal is to find blind spots of a black box model which has already been deployed. Furthermore, query strategies employed in active learning are guided towards the discovery of *known unknowns*, utilizing information from the predictive model to determine which instances should be labeled by the oracle. These approaches are not suitable for the discovery of unknown unknowns as the model is not aware of unknown unknowns and it lacks meaningful information towards their discovery.

Outlier Detection Outlier detection involves identifying individual data points (global outliers) or groups of data points (collective outliers) which either do not conform to a target distribution or are dissimilar compared to majority of the instances in the data (Han, Pei, and Kamber 2011; Chandola, Banerjee, and Kumar 2007). Several parametric approaches (Agarwal 2007; Abraham and Box 1979; Eskin 2000) were proposed to address the problem of outlier detection. These methods made assumptions about the underlying data distribution, and characterized those points with a smaller likelihood of being generated from the assumed distribution, as outliers. Non-parametric approaches (Eskin 2000; Eskin et al. 2002; Fawcett and Provost 1997) which made fewer assumptions about the distribution of the data such as histogram based methods, distance and density based methods were also proposed to address this problem. Though unknown unknowns of any given predictive model can be regarded as collective outliers w.r.t the data used to train that model, the aforementioned approaches are not applicable to our setting as we assume no access to the training data.

Discussion & Conclusions

We presented an algorithmic approach to discovering unknown unknowns of predictive models. The approach assumes no knowledge of the functional form or the associated training data of the predictive models, thus, allowing the method to be used to build insights about the behavior of deployed predictive models. In order to guide the discovery of unknown unknowns, we partition the search space and then use bandit algorithms to identify partitions with larger concentrations of unknown unknowns. To this end, we propose novel algorithms both for partitioning the search space as well as sifting through the generated partitions to discover unknown unknowns.

We see several research directions ahead, including opportunities to employ alernate objective functions. For instance, the budget $\mathcal{B}$ could be defined in terms of the total cost of querying the oracle instead of the number of queries to the oracle. Our method can also be extended to more sophisticated settings where the utility of some types of unknown unknowns decreases with time as sufficient examples of the type are discovered (e.g., after informing the engineering team about the discovered problem). In many settings, the oracle can be approximated via the acquisition of labels from crowdworkers, and the labeling noise of the crowd might be addressed by incorporating repeated labeling into our framework.

The discovery of unknown unknowns can help system designers when deploying predictive models in numerous ways. The partitioning scheme that we have explored provides interpretable descriptions of each of the generated partitions. These descriptions could help a system designer to readily understand the characteristics of the discovered unknown unknowns and devise strategies to prevent errors or recover from them (e.g., silencing the model when a query falls into a particular partition where unknown unknowns were discovered previously). Discovered unknown unknowns can further be used to retrain the predictive model which in turn can recognize its mistakes and even correct them.

Formal machinery that can shine light on limitations of our models and systems will be critical in moving AI solutions into the open world–especially for high-stakes, safety critical applications. We hope that this work on an algorithmic approach to identifying unknown unknowns in predictive models will stimulate additional research on incompleteness in our models and systems.

Acknowledgments

Himabindu Lakkaraju carried out this research during an internship at Microsoft Research. The authors would like to thank Lihong Li, Janardhan Kulkarni, and the anonymous reviewers for their insightful comments and feedback.

References

Abraham, B., and Box, G. E. 1979. Bayesian analysis of some outlier problems in time series. *Biometrika* 66(2):229–236.

Agarwal, D. 2007. Detecting anomalies in cross-classified streams: a bayesian approach. *Knowledge and information systems* 11(1):29–44.

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

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

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

Brennan, T.; Dieterich, W.; and Ehret, B. 2009. Evaluating the predictive validity of the compas risk and needs assessment system. *Criminal Justice and Behavior* 36(1):21–40.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Chandola, V.; Banerjee, A.; and Kumar, V. 2007. Outlier detection: A survey.

Chapelle, O., and Li, L. 2011. An empirical evaluation of thompson sampling. In *NIPS*, 2249–2257.

Crawford, K. 2016. Artificial intelligence's white guy problem. New York Times. http://www.nytimes.com/2016/06/26/opinion/sunday/artificial-intelligences-white-guy-problem.html.

Decoste, D., and Schölkopf, B. 2002. Training invariant support vector machines. *Machine learning* 46(1-3):161–190.

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

Eskin, E.; Arnold, A.; Prerau, M.; Portnoy, L.; and Stolfo, S. 2002. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*. Springer. 77–101.

Eskin, E. 2000. Anomaly detection over noisy data using learned probability distributions. In *ICML*, 255–262.

Fawcett, T., and Provost, F. 1997. Adaptive fraud detection. *Data mining and knowledge discovery* 1(3):291–316.

Feige, U. 1998. A threshold of ln n for approximating set cover. *Journal of the ACM* 45(4):634–652.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine learning* 2(2):139–172.

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

Graepel, T., and Herbrich, R. 2004. Invariant pattern recognition by semidefinite programming machines. In *NIPS*, 33.

Han, J.; Pei, J.; and Kamber, M. 2011. *Data mining: concepts and techniques*. Elsevier.

Horvitz, E. 2008. Artificial intelligence in the open world. Presidential Address, AAAI. http://bit.ly/2gCN7t9.

Jiang, J., and Zhai, C. 2007. A two-stage approach to domain adaptation for statistical classifiers. In *CIKM*, 401–410.

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

Kaggle. 2013. Dogs vs cats dataset. https://www.kaggle.com/c/dogs-vs-cats/data.

Kim, B.; Rudin, C.; and Shah, J. A. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 1952–1960.

Lakkaraju, H., and Leskovec, J. 2016. Confusions over time: An interpretable bayesian model to characterize trends in decision making. In *NIPS*, 3261–3269.

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

Lewis, D. D., and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *SIGIR*, 3–12.

Li, Z.; Peng, H.; and Wu, X. 2008. A new descriptive clustering algorithm based on nonnegative matrix factorization. In *IEEE International Conference on Granular Computing*, 407–412.

McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *KDD*, 785–794.

Michalski, R. S., and Stepp, R. E. 1983. Learning from observation: Conceptual clustering. In *Machine learning: An artificial intelligence approach*. Springer. 331–363.

Pang, B., and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, 271.

Pang, B., and Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, 115–124.

Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2009. *Dataset Shift in Machine Learning*. The MIT Press.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " why should i trust you?": Explaining the predictions of any classifier. In *KDD*.

Settles, B.; Craven, M.; and Ray, S. 2008. Multiple-instance active learning. In *NIPS*, 1289–1296.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *COLT*, 287–294.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90(2):227–244.

Slivkins, A., and Upfal, E. 2008. Adapting to a changing environment: the brownian restless bandits. In *COLT*, 343–354.

Teo, C. H.; Globerson, A.; Roweis, S. T.; and Smola, A. J. 2007. Convex learning with invariances. In *NIPS*, 1489–1496.

Weiss, D. 2006. *Descriptive clustering as a method for exploring text collections*. Ph.D. Dissertation.

Wolpert, D. H., and Macready, W. G. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1(1):67–82.

Zhang, T., and Oles, F. 2000. The value of unlabeled data for classification problems. In *ICML*, 1191–1198.

Zhu, X.; Lafferty, J.; and Ghahramani, Z. 2003. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*.