Facility Location Utility for Uncovering Unknown Unknowns

Karsten Maurer[1] **Walter Bennette**[2]
[1]Miami University Department of Statistics - maurerkt@miamioh.edu
[2]Air Force Research Lab Information Directorate - walter.bennette.1@us.af.mil

Abstract

AAAI creates proceedings, working notes, and technical reports directly from electronic source furnished by the authors. To ensure that all papers in the publication have a uniform appearance, authors must adhere to the following instructions.

Introduction

Techniques such as active learning [Settles, 2010] and domain adaptation [Patel et al., 2014] can be used to create machine learning classifiers when large labeled datasets are not available for a specific task. For example, the black box classifiers made available through many online services (list services) require no training data and can be thought of as a kind of domain adaptation. However, with limited amounts of labeled data, users may not be able to properly evaluate a model, and are left hoping the model will be useful for their intended task. In this paper we build upon previous work to develop an interactive method to help evaluate classifiers in the absence of labeled data. Specifically, we develop an interactive method to uncover unknown unknowns [Attenberg, Ipeirotis, and Provost, 2015]: instances for which a classifier is confident in its prediction, but is wrong.

Intuitive methods can be used to evaluate the performance of a model in the absence of labeled data. For example, given a labeling budget one could sample instances following an experimental design, sample instances with the lowest classifier confidence, or sample instances identified as informative to the classifier through active learning strategies. These methods could provide a sense of a model's performance but will potentially miss high confidence mistakes, referred to as *unknown unknowns* (UUs).

Unknown unknowns can be thought of as blind spots to a classification model, and can be caused by dataset bias during training, domain shift during use, lack of model expressibility, and other causes of poor model fit. From the viewpoint of a rational actor, unknown unknowns represent costly mistakes because minimal risk mitigation strategies will have been deployed for these high confidence predictions. The discovery of unknown unknowns may allow new mitigation strategies to be formulated [Nushi et al., 2016]. Additionally, as enumerated in [Bansal, Weld, and Allen, 2018], finding unknown unknowns is valuable to understand classifier limitations and prevent attacks (stole this hard) what do you mean by attacks?

Attenberg, Ipeirotis, and Provost [2015] gamified the search for unknown unknowns and relied on human oracles to discover misclassified instances. A utility-based search algorithm for discovering unknown unknowns was then proposed by Lakkaraju et al. [2016] and expanded upon by Bansal, Weld, and Allen [2018]. In this paper, we propose a new utility function that shifts the emphasis from simply finding UUs, to finding UUs that occur more often than the confidence would suggest, thus searching for classifier *overconfidence*.

Previous Works

Lakkaraju et al. [2016] uses a utility model that simply counts the number of discovered unknown unknowns and searches using multi-armed bandits. Bansal, Weld, and Allen [2018] argued that this unit utility model motivates the discovery of very similar unknown unknowns. To fix this problem they proposed an adaptive coverage-based utility model that attempts to encourage the discovery of unknown unknowns throughout the feature space, favoring high confidence regions. They then search for unknown unknowns via a greedy algorithm to maximize utility.

The utility in Bansal, Weld, and Allen [2018] has the form:

$$U(Q) = \sum_{x \in \mathbb{X}} c_x \cdot \max_{q \in S} \{sim\,(x, q)\}$$

where $\mathbb{X} \subset \mathbb{R}^p$ is the set of available p-dimensional unlabeled test instances, $Q \subset \mathbb{X}$ is the set of instances labeled by an oracle, $S = \{x | x \in Q, y_x \neq M(x)\}$ is the set of discovered unknowns unknowns for some classifier $M(x) : \mathbb{X} \to class$, c_x is the classifier's confidence in its prediction of x, and $sim(x, q)$ is a distance based similarity metric. The instance q' that maximizes the expected utility increase is greedily selected to be labeled by the oracle

$$E\left[U_x\left(Q \cup q'\right)\right] = \hat{\phi}(x) \cdot c_x \cdot \max_{q \in S \cup q'} \{sim\,(x, q)\}$$

where $\hat{\phi}(x) = P\left(y_x \neq M(x) | Q\right)$ is a conditional probability that x is misclassified given the query set. As previously stated, this method is designed to incentivize a broader

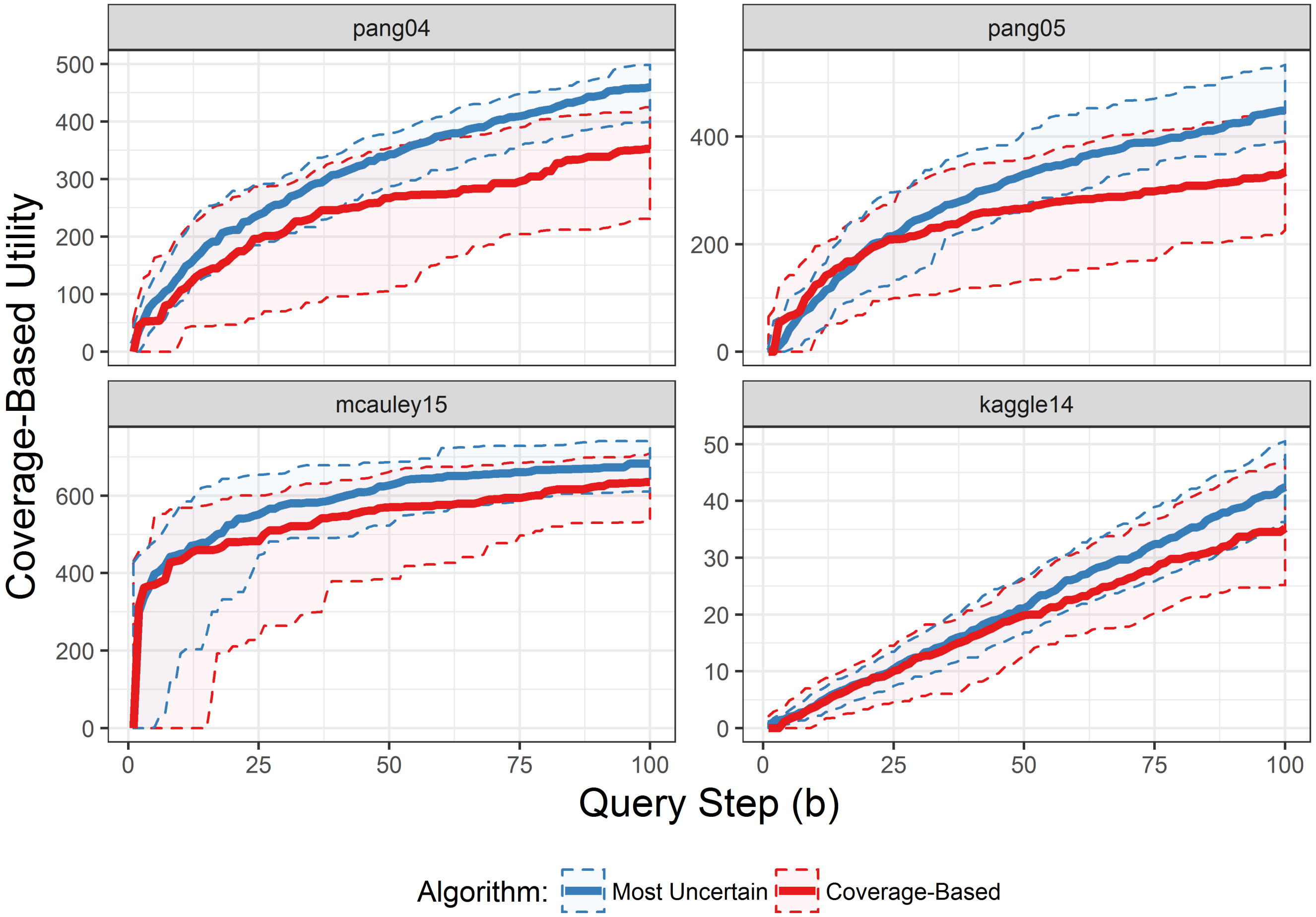

Figure 1: Empirical evidence that Coverage-Based Utility reward low confidence search.

search for unknown unknowns and gives higher utility for finding misclassifications in higher confidence regions.

Unfortunately, the method of Bansal, Weld, and Allen [2018] is consistently outperformed by sequentially querying the instances for which the classifier is most uncertain. Query results were collected under the two search algorithms across 100 random samples of size n=1000 from each set of test data, with budget of B=100 quieries, because the greedy algorithms can unfold with high variability depending on the points gathered for testing. Figure 1 displays the Monte Carlo medians and 90% confidence envelopes for the coverage-based utility for the four empirical data sets used in Bansal, Weld, and Allen [2018]. We see that the most-uncertain algorithm is typically given higher utility early in the algorithm than the results of the greedy-adaptive algorithm using the cluster-based probabilities to optimally select the next point. We believe this exposes a flaw in the coverage-based utility model. We see that the most-uncertain algorithm is typically given higher utility early in the algorithm than the results of the greedy-adaptive algorithm using the cluster-based probabilities to optimally select the next point. This reward strategy would help find high confidence unknown unknowns if instances with similar model prediction confidence scores were near each other in the feature space, but this does not seem to be the case.

The coverage-based utility model attributes high utility for all high confidence points in the test set that are close to any discovered UU, thus attributing a utility to the uncertain status of non-queried points and even to points that have been confirmed by the oracle query to not be UU. Suppose that a single misclassified point exists in a region of high confident predictions, the utility would increase substantially for the surrounding points, despite not being UU. With these appearant issues, we aim to construct a utility-based query algorithm that more appropriately rewards the identification of high confidence misclassifications, and helps to identify regions of classifier overconfidence.

Methodology

We propose an alternative utility model based roughly on facility location optimization methods (Farahani and Hekmatfar, 2009). In the facility location problem a utility can be constructed that uses a greedy algorithm to minimize the cost, or maximize the reward, of building a series of new facilities in a supply chain, while also minimizing distances between clients to the nearest facility (Guha and Khuller, 1999; Arya et al., 2004). In the UU query setting, we can draw an analog to the selection of a point to query to the establishment of a facility at that location in the feature space; evaluating the reward for selecting the point, and the distance it stands from the surrounding unobserved points. We propose a facility location utility function as:

$$W(Q) = \sum_{x \in S} r\left(c_x\right) - \frac{1}{n} \sum_{x \in \mathbb{X}} \min_{q \in S} \left(d\left(x, q\right)\right)$$

where $r\left(c_x\right) = \log(1/(1 - c_x))$ is the reward function for finding an UU with confidence c_x, and $d(x, q)$ is the Euclidean distance between points x and q. We use the greedy algorithm that at each iteration selects q' with the maximum expected utility, as defined in Algorithm 1 below.

Algorithm 1 Greedy Facility Location Search

```
**Input:** Test set $\mathbb{X}$, prior $\hat{\phi}\left(x|Q = \emptyset\right)$, budget B
$Q = \{\}$ {inputs that have been queried}
$y_Q = \{\}$ {oracle defined labels}
**For:** $b = 1, 2, ..., B$ **do:**
$q' = \text{argmax}_{q' \notin Q} E\left[W\left(Q \cup q'\right)\right]$
$y_{q'} = OracleQuery(q')$
$Q \leftarrow Q \cup q'$
$S \leftarrow \{x | x \in Q \text{ and } y_x \neq M(x)\}$
$b \leftarrow b + 1$
**Return:** $Q$ **and** $y_q$
```

At each iterative step in Algorithm 1, we need to select the point that will maximize the expected gain in facility location utility, given probability estimates for point misclassification, such that $\hat{\phi}(q'|Q) = \hat{P}(y_{q'} \neq M(q')|Q)$. To find the expected gain in utility for each point, we evaluate the utility under the possibilities that the point is either misclassified or correctly classified. These possible utility outcomes are then averaged with weights equal to the estimated probability of each outcome. Thus the optimization step requires the solution of the following:

$$\underset{q' \notin Q}{argmax} E[W(Q \cup q')] =$$

$$\underset{q' \notin Q}{argmax} \left[\begin{array}{c} \hat{\phi}(q') \cdot \left[\sum\limits_{x \in S \cup q'} r\left(c_x\right) - \frac{1}{n} \sum\limits_{x \in \mathbb{X}} \min\limits_{q \in S \cup q'} \left(d\left(x, q\right)\right) \right] + \\ \left(1 - \hat{\phi}(q')\right) \cdot \left[\sum\limits_{x \in S} r\left(c_x\right) - \frac{1}{n} \sum\limits_{x \in \mathbb{X}} \min\limits_{q \in S} \left(d\left(x, q\right)\right) \right] \end{array} \right]$$

Note that $\left[\sum_{x \in S} r\left(c_x\right) - \frac{1}{n} \sum_{x \in \mathbb{X}} \min_{q \in S} \left(d\left(x, q\right)\right)\right]$ is constant for all considered points, but cannot simply be removed from the argmax solution because it is multiplied by an estimated probability that is unique to each point.

In addition to a change to the utility structure, we propose the use of model-based estimates for $\hat{\phi}(x)$. In this paper we demonstrate the use of logistic regression classifiers as alternatives to the cluster-based probabilities used in **Bansal and Welds (2018)** adaptive greedy algorithm.

There are a few characteristics to note in the design of the facility locations utility model. First, the rewards are only accumulated by finding UUs in the query set, which avoids the issue of placing value on points in the test set for simply having high confidence points. Whereas having a small average minimum distance between all test points to the closest observed UU is also seen as valuable for encouraging strong coverage by the query set, especially early in the query sequence.

The reward function is designed to impact the utility in a way that is consistent with a limiting factor being the budget for oracle queries. Viewed as a geometric distribution problem with a probability $\phi(x)$ of discovering a UU, we expect to need $1/\phi(x)$ queried points like point x before discovering the first UU (Casella and Berger, 2002). For heuristic insight into the reward behavior construction, if we assume that $\phi(x) = (1 - c_x)$, then our reward is a log-scaled count of the number of randomly selected points we would expect to query in order to find the UUs in our query set. We use the log scaling to avoid over-incentivizing the search for incredibly rare UUs, as we know there is a limited budget for oracle queries. The optimization step will provide the highest expected rewards for selecting the most overconfident points relative to the updated probability estimates, that is to say when $(1 - c_x) < \hat{\phi}(x)$. Note that unlike the UU definition, this construction does not depend on the arbitrary definition of a confidence threshold, τ, beyond which we search for misclassifications. The reward component of the facility locations utility should encourage the query set to select any points where the model is most overconfident. We define *overconfidence* as the difference between the confidence values given by the classifier and the actual rates of correct classification.

Experimental Evaluation

We evaluate our facility location utility model by applying Algorithm 1 to four empirical data sets: pang04, pang05, mcauley15 and kaggle14. These sets originating from the Pang and Lee (2004) movie review text classifier, Pang and Lee (2005) movie review text classifier, Mcauley et al. (2015) media review text classifier, and Kaggle (2014) dogs-vs-cats image classifier, respectively. In each case we fit a classifier, $M(x)$, to a biased training set, then generate predicted classes and confidence values for all observations in the test set. Then singular value decomposition was used to reduce the dimensionality such that the test set, $\mathbb{X} \in \mathbb{R}^2$. The sets and classifiers were chosen to maintain consistency with the data used to evaluate methods in the previous works by Lakkaraju et al. (2017) and Bansal and Weld (2018). The pang04, pang05 and Mcauley15 sets come directly from the generously shared repository accompanying the work Bansal and Weld (2018). The construction of the kaggle14 training set, test set and classifier....

Figure 2 displays the overconfidence of the models using a cubic-splines fitted between the indicator of correct classification an the confidence value in each test dataset to obtain an empirical estimates of the true rates of correct classification. We see that the models from the Pang04 and Pang05

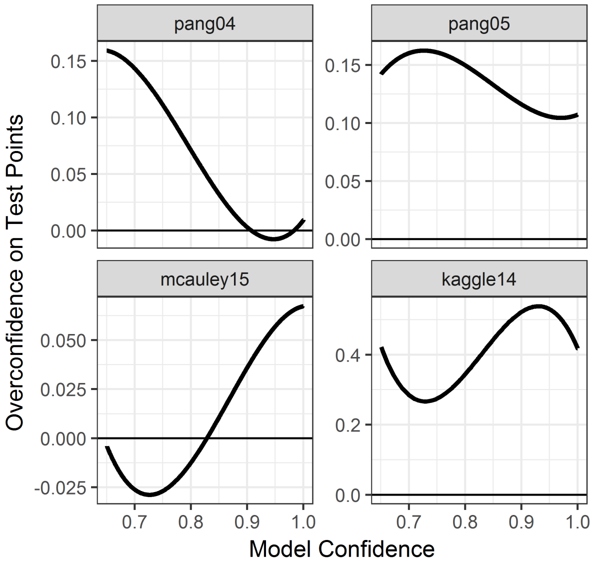

Figure 2: Observed overconfidence of the model for experimental data sets.

datasets are most overconfident for points with relatively low confidence values, thus we would expect the a simple sequential search of the most-uncertain points to provide high facility locations utility in these cases. Whereas the predictions for McCauley15 and Kaggle14 are most overconfident in the highest confidence range, thus the most-uncertain search should provide low facility locations utility in these cases. We see that these four data sets represent fairly different profiles of overconfidence, thus present good variety for evaluating characteristics of the facility locations utility model.

To evaluate the queries generated by the facility locations utility model we collect query results from running Algorithm 1 on repeated random samples from the test sets, thus allowing Monte Carlo estimates for utility characteristics. The optimization step was completed using both the logistic regression and the cluster-based construction of the probability estimates, $\phi(x)$. Figure 2 displays the Monte Carlo medians and 90% confidence envelopes for the utility gains, paired with a visualization of confidence ranges where the model most overestimates its accuracy. To allow meaningful comparison over different test sets, that performance is scaled relative to the tractable upper bound obtained by running the greedy algorithm with an omniscient knowledge of test misclassification.

In Figure 3, we see that the most-uncertain selection method that begins its search with points with confidence values just above τ=.65 provides the strongest utility for the pang04 case, as is desired, because these points are also shown to be most overconfident. The facility locations utility outcomes from using the logistic-based optimization were typically slightly lower. For the pang05 case where the overconfidence is skewed right but relatively ubiquitous, we again find very comparable utility between the logistic-based and the most-uncertain queries. For the mcauley15 where overconfidence is heavily skewed left, we now see strong facility location utility outcomes from the logistic-based optimization, and very low utility outcomes from the most-uncertain queries. In the last case of kaggle14 where the overconfidence is skewed left but relatively ubiquitous, ...

We also look to compare the queries gathered by the coverage-based utility algorithm and facility locations utility algorithm. Given that both rely on greedy selection relative to their own utility function, it does not make sense to compare their selections on the utility values directly. Instead we will look to using a standardized discovery ratio ...

dataset	Most Uncertain	Coverage-Based	Facility Locations
pang04	1.53 (1.33,1.71)	1.47 (1.23,1.67)	1.43 (1.14,1.69)
pang05	1.41 (1.21,1.56)	1.66 (1.35,2.06)	1.54 (1.16,2.08)
mcauley15	0.89 (0.71,1.08)	1.09 (0.83,1.61)	1.27 (0.76,8.08)
kaggle14	2.55 (2,3.02)	3.09 (2.33,4.26)	3.58 (2.52,4.99)

Discussion & Conclusions

Previous literature has defined unknown unknowns as any highly confident predictions that results in misclassification, but this definition ignores the unavoidable uncertainties of predictive modeling. It should be expected that classifier predictions are imperfect, this is why confidence statements exist! The actions we take as a result of the predictions should take into account the inherent uncertainty. However, in the case where the claimed confidence is overstated, we cannot properly mitigate the risk posed by misclassification. Unlike the previous works that propose utility functions that simply seek to uncover high confidence misclassifications, the facility locations utility that we propose is designed to seek out overconfident misclassifications. We have demonstrated the ability of our greedy algorithm to consistently obtain strong facility locations utility in four data scenarios with disparate overconfidence profiles.

The use of the logistic regression probability estimates for $\hat{\phi}(x)$ in the optimization step appears to be robust to the underlying structure overconfidence, providing a strong misclassification discovery ratio in the tested cases. This is important because in real-world applications we wouldnt know the overconfidence behavior a priori to our query search, so we require a versatile estimation method.

There are many avenues for future work related to the facility locations utility methods that we have presented. First, the facility locations utility model structure separates the discovery reward and coverage proximity components, which could allow separate rescaling to weight each component in line with the priorities of an application. Next, exploratory methods could be developed to evaluate what the query set tells us about the overconfidence of your model, perhaps interpreting the structure of the models used to predict $\hat{\phi}(x)$ to better understand what features are related to overconfidence. Lastly, there may be cases where it is impractical to collect a large enough oracle query set to refit the original classifier, but it may be sufficient to estimate the

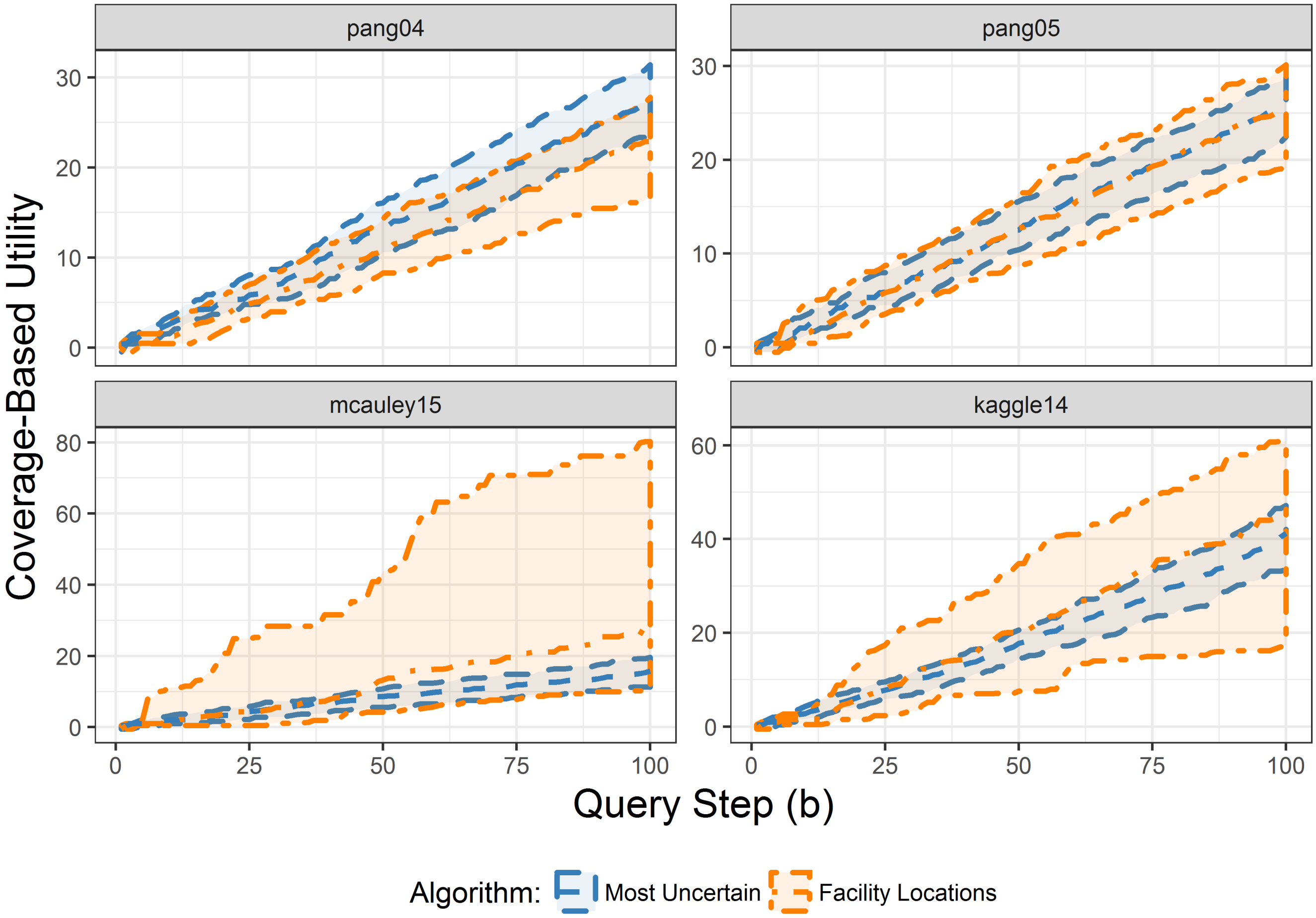

Figure 3: Faciltiy Locations Utility rewards finding overconfidence

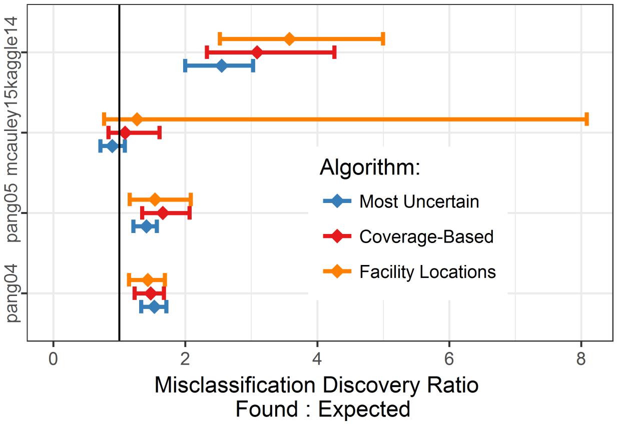

Figure 4: Misclassification Discovery Ratio.

original classifiers overconfidence and perform recalibration so that actions taken based on the predictions can include more appropriate risk mitigation.

References

Attenberg, J.; Ipeirotis, P.; and Provost, F. 2015. Beat the Machine. *Journal of Data and Information Quality* 6(1):1–17.

Bansal, G.; Weld, D. S.; and Allen, P. G. 2018. A Coverage-Based Utility Model for Identifying Unknown Unknowns. *Aaai 2018* 8.

Lakkaraju, H.; Kamar, E.; Caruana, R.; and Horvitz, E. 2016. Identifying Unknown Unknowns in the Open World: Representations and Policies for Guided Exploration.

Nushi, B.; Kamar, E.; Horvitz, E.; and Kossmann, D. 2016. On Human Intellect and Machine Failures: Troubleshooting Integrative Machine Learning Systems.

Patel, V.; Gopalan, R.; Li, R.; and Chellappa, R. 2014. Visual Domain Adaptation: An Overview of Recent Advances. 1–34.

Settles, B. 2010. Active Learning Literature Survey. *Machine Learning* 15(2):201–221.