Algorithms for Active Classifier Selection: Maximizing Recall with Precision Constraints

Paul N. Bennett*, David M. Chickering, Christopher Meek
Microsoft Research
Redmond, WA, USA
{pauben, dmax, meek}@microsoft.com

Xiaojin Zhu†
University of Wisconsin
Madison, WI, USA
jerryzhu@cs.wisc.edu

ABSTRACT

Software applications often use classification models to trigger specialized experiences for users. Search engines, for example, use query classifiers to trigger specialized "instant answer" experiences where information satisfying the user query is shown directly on the result page, and email applications use classification models to automatically move messages to a spam folder. When such applications have acceptable default (i.e., non-specialized) behavior, users are often more sensitive to failures in model precision than failures in model recall. In this paper, we consider model-selection algorithms for these precision-constrained scenarios. We develop adaptive model-selection algorithms to identify, using as few samples as possible, the best classifier from among a set of (precision) qualifying classifiers. We provide statistical correctness and sample complexity guarantees for our algorithms. We show with an empirical validation that our algorithms work well in practice.

Keywords

guaranteed precision; efficient evaluation; model selection

1. INTRODUCTION

Classification models are often subject to precision constraints in applications where they are used to trigger specialized user experiences. Search engines, for example, use query classifiers to trigger specialized "instant answer" experiences where information satisfying the user query is shown directly on the result page. These systems arbitrate among a large and ever-increasing number of specialized responses including: showing a stock quote after classifying the query as a stock symbol; returning a definition after classifying the query as having glossary intent; or displaying a map after classifying the query as having local intent. In all cases, the system has the option to suppress all specialized experiences and instead use the *default* behavior of providing only the "ten blue links" in the search results. When a reasonable default behavior is available, classifier precision is usually more important to users than classifier recall; in a study of search engine switching behavior, [7] demonstrated that dissatisfaction with the search results presented (DSAT Rate) is 3.25x more likely than coverage (failure to provide desired results) to cause users to switch search engines. Beyond instant answers in search engines, intelligent assistants (e.g., Google Now, Facebook's M, Apple's Siri, or Microsoft's Cortana) represent another class of specialized triggers that various applications can use to enhance user experience.

In principle we could choose classification models for the above applications by optimizing user utility under assumptions about the relative cost of false positives and false negatives. In practice, however, companies often wish to protect perceived brand quality and may take the more conservative approach of setting a strict constraint on model precision – then work to maximize recall subject to that precision constraint. In this paper, we assume that we have a set of candidate classification models and a precision constraint, and we develop adaptive model-selection algorithms that will select a high-recall model that satisfies that constraint using minimal labeling effort.

Model selection is a fundamental problem in machine learning that arises both when choosing thresholds for a single classifier and when choosing among alternative classifiers. Applications often require selection algorithms that balance between two or more competing dimensions of quality. Examples include: (1) choosing treatment dosage in clinical trials, where there is a trade-off between the drug benefit and the drug toxicity; (2) spam filtering, where there is a trade-off between having good-email in the spam folder and bad-email in the inbox; and (3) record matching, where there is a tradeoff between incorrectly matching records and the failure to match near-duplicate records.

In the information retrieval community, researchers typically frame such tradeoffs in terms of precision versus recall. Unfortunately, when the fraction of positive examples is small, accurately estimating recall requires a large number of samples. Due to these challenges, we focus instead on the tradeoff between precision and the alternative *reach* quality criterion, which is the number of true positives in a test set that are predicted positive by the classifier. Reach is sometimes called *cardinality* in the information retrieval community [10] and is a natural measure of the efficacy of a classifier on a test set. For example, in the case of a particular instant answer query classifier, it is the number of queries for which the instant answer experience correctly triggers.

*Authors alphabetical – all authors contributed equally.

†Work performed while at Microsoft Research.

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

 ISBN 978-1-4503-4675-7/17/02…$15.00

DOI: http://dx.doi.org/10.1145/3018661.3018730

In academic settings, the tradeoff between precision and recall is often measured using a single combination criterion such as an F_α-measure. As discussed above, however, model-selection algorithms are often required to limit the potential damage incurred by using the classifier (e.g., probability of dying in a clinical trial or probability of good-email in the spam folder for an email application), and in many cases a minimum threshold on the precision of the classifier is appropriate.

In this paper, we develop adaptive model-selection algorithms for identifying high-reach classifiers that have acceptable precision. We provide sample-complexity guarantees for our algorithms and statistical guarantees for the selected classifiers. We compare our algorithms on a set of web-page classification tasks and demonstrate our algorithms can significantly reduce the number of required samples.

2. PROBLEM SETTING

As mentioned above, for many classification settings, positive examples are rare and estimating recall with statistical certainty can be costly. Naïvely, selecting the best model from a set of models is a costly process where each good model must be certified and the winner chosen from among the good candidates. Our primary insight is to recast this common challenge as a problem of choosing from among all models the good ones (those matching a precision condition) and *maximizing* reach from among these good classifiers. Note that maximizing reach is equivalent to maximizing recall but it does not require that we estimate the denominator of recall which is constant across all classifiers. Since we do not need to estimate the number of positives in all data but only in those predicted to be positive by any one of the classifiers, this reduces the statistical power needed. Additionally, we can adapt bandit approaches to combine the steps of precision-qualification together with estimation of reach. Finally, we generalize our approach to allow a user-specified slack in the precision condition, tightness of the maximization of reach, or statistical confidence desired.

More formally, we assume a setting where a "user" wants a system that can select the best classifier from a set of m candidate classifiers $h_1, \ldots, h_m$. In practical settings, the user is typically an engineer with domain expertise and experience applying machine learning software packages to learn models, but who may not have deep experience in learning theory or the design of new machine learning methods. The user is thus assumed to have domain knowledge as to what level of precision is acceptable in the target domain. In some cases, the user may want to find an approximate best answer – this can be especially valuable early in the stages of interactive model development when the goal is to quickly choose among promising candidate models or feature sets for further exploration; in this setting labels are often sparse and the user would like to spend as little time as possible providing additional labels for evaluation.

We assume the availability of a large unlabeled test corpus. We can ask human oracles for the true label of any test item, but we want to do so sparingly since each incurs labeling cost. Imagine we apply each classifier h_i separately to the test corpus for $i = 1 \ldots m$. If we knew the true labels of each test item, we would be able to partition the corpus into (1) n_{1i} true positive items with true label + and predicted label +; (2) n_{2i} false negative items with true label + and predicted label -; (3) n_{3i} false positive items with true label - and predicted label +; (4) n_{4i} true negative items with true label - and predicted label -. The recall of h_i is $\frac{n_{1i}}{n_{1i}+n_{2i}}$ and the precision of h_i is $\frac{n_{1i}}{n_{1i}+n_{3i}}$.

Based on domain knowledge, the user specifies a desired precision threshold, PT. We say a classifier is a *good classifier* if it has precision at least PT. In the approximate selection case, the user may have a tolerance for precision dropping slightly below this threshold. The user can designate this tolerance using a slack variable for the precision threshold, γ where $0 \leq \gamma \leq PT$, and we call a classifier *acceptable* if it has a precision of at least $PT - \gamma$.

Among precision-acceptable classifiers, we want to select the one with the largest recall. One practical difficulty in practice is to estimate the denominator $n_{1i} + n_{2i}$, which can require excessive labeling. As described briefly above, note that the sum $n_{1i} + n_{2i}$ is the total number of true positive items in the test corpus and the sum is constant across all choices of i. Thus the numerator n_{1i} provides the same recall ranking among the classifiers with respect to the test corpus. Therefore, we can state our goal equivalently as maximizing n_{1i} subject to the precision acceptability constraint. We define n_{1i} as the *reach* of classifier h_i.

For convenience, the *trivial acceptable classifier*, α, is the classifier with acceptable precision that classifies every example as not in the class. This classifier is trivially acceptable because we define its precision to be 1 and its reach to be 0. We assume that in addition to the models provided by the user, the method can also produce the trivially acceptable classifier as an answer. We call a classifier optimal if it is the maximizer of reach among all good classifiers. Let R^* denote the reach of the optimal classifier; note that this is always defined because the trivial acceptable classifier is a candidate model. Similar to precision, we assume the user may be willing to tolerate a small percentage decrease in reach relative to the optimal classifier such that the reach of the selected model is at least $(1-\epsilon)R^*$. The user may specify this tolerance as a slack variable ϵ where $0 < \epsilon \leq 1$.

Finally, we assume that the user specifies a desired confidence $1-\delta$ in the result where $\delta \in (0,1)$ is another user-specified parameter. We take as the user's end goal to select with probability $1-\delta$ an acceptable classifier whose reach is at least $(1-\epsilon)R^*$.

More formally, let p_i and R_i denote the precision and reach of classifier i, respectively. Then given a set of models M of size m, and user-specified parameters of PT, γ, ϵ, and δ, the method returns a classifier $c \in M \cup \{\alpha\}$ such that the following two conditions hold with probability $1-\delta$:

$$p_c \geq PT - \gamma \quad (1)$$

$$R_c \geq (1-\epsilon) \max_{c' \in M \cup \{\alpha\} \text{ s.t. } p_{c'} \geq PT} R_{c'} \quad (2)$$

We call a model that meets these two constraints a (γ, ϵ) approximately optimal classifier. We next demonstrate how the solution to this problem can be cast as a novel multi-armed bandit problem before turning to the description of several algorithms to solve the problem.

3. PROBLEM APPROACH

Let the *predicted positive set* $PP_i = \{x : h_i(x) = 1\}$ be the subset of the test corpus on which h_i predicts positive. If we have enough human oracle resources to label $\cup_{i \in [m]} PP_i$ then we can achieve our goal exactly. In practice we will use adaptive sampling. For any classifier h_i, if we draw t_i items

$x_1, \ldots, x_{t_i}$ uniformly with replacement from its predicted positive region PP_i and ask the oracle for labels $y_1, \ldots, y_{t_i}$, we define an estimator for the precision as

$$\hat{p}_{t_i} = \frac{\sum_{j=1}^{t_i} \mathbb{1}(y_j = 1)}{t_i}. \quad (3)$$

This is a pure exploration multi-armed bandit problem, where each classifier is a bandit arm. We want to identify all arms whose precision is better than PT; furthermore, we want to identify the top arm in terms of its reach n_1 among those surviving arms. An interesting feature is that PP_i and PP_j may overlap. The act of pulling an arm may also pull some other arms – the arms are coupled. In more basic terms, this means that for a carefully designed procedure a true label that was obtained from a sampled predicted positive for one classifier can also be used to evaluate another classifier that predicts it to be positive. In particular we take advantage of this in our pooled-sampling-shared-label sampling approach (Algorithm 2) by uniformly sampling over the union of all of the predicted positives for all classifiers we are still evaluating. Since all predicted positives were available to sample with uniform probability, estimates for reach and precision (which depend on predicted positives only) can be updated for *all* the classifiers whose predicted positive sets were in the union used for sampling without resorting to more sophisticated reweighting using importance weights. We note that this coupling which exploits the shared relationship between arms is also a novel contribution to the bandit literature.

Obviously $\mathbb{E}[\hat{p}_{t_i}] = \frac{n_{1i}}{n_{1i}+n_{3i}}$ and Hoeffding's inequality holds for a fixed i and a fixed t_i. However, we need an anytime bound that holds simultaneously for all possible t_i and for all classifiers i. In other words, we want a function $U(t, \delta)$ such that given $\delta > 0$,

$$P\left(\forall i \in [m], \forall t_i, |\hat{p}_{t_i} - \frac{n_{1i}}{n_{1i}+n_{3i}}| \leq U(t_i, \delta)\right) \geq 1 - \delta. \quad (4)$$

Note that the above precision statement is equivalently a statement on the reach n_1:

$$P\left(\forall i \in [m], \forall t_i, |\hat{p}_{t_i}|PP_i| - n_{1i}| \leq |PP_i|U(t_i, \delta)\right) \geq 1-\delta. \quad (5)$$

Because these bounds are true for all time, we can define a stricter confidence bound at time t_i as the intersection of individual intervals up to this time:

$$LCB_{i,t_i} = \max_{j=1}^{t_i} \hat{p}_{i,j} - U(j, \delta) \quad (6)$$

$$UCB_{i,t_i} = \min_{j=1}^{t_i} \hat{p}_{i,j} + U(j, \delta). \quad (7)$$

Then, with large probability

$$p_i \in [LCB_{i,t_i}, UCB_{i,t_i}] \text{ for all } t_i. \quad (8)$$

There are several choices of $U(t, \delta)$. The simplest one is perhaps confidence-splitting among a finite T (e.g. $T = |\cup_{i\in[m]} PP_i|$) and m arms. That is, start with a cap T on the number of test items the oracle is willing to label. One choice is $T = |\cup_{i\in[m]} PP_i|$. We request a stricter confidence $\frac{\delta}{mT}$ for each fixed t and for each arm, so that the total failure probability is bounded by δ. In particular, with Hoeffding's bound we may define

$$U(t, \delta) = \sqrt{\frac{\log(2mT/\delta)}{2t}}. \quad (9)$$

With these bounds on precision and reach, we propose in the next section a novel algorithm to find the best classifier.

4. ALGORITHMS

Our goal is to choose an acceptable classifier that is guaranteed to have reach nearly as good or better than any good classifier, without needing to label too many samples. Algorithm 1 is a schematic algorithm for selecting precision-qualified high-reach models. It is schematic because we consider alternative implementations for the routine `SampleAndUpdateStatistics`. We say that an implementation of the `SampleAndUpdateStatistics` algorithm is *statistically valid* if it chooses at most one predicted positive example associated with one of the classifiers, obtains the label for this example from an oracle and performs statically valid updates of counts and bounds.

Data: The m classifiers have predicted positive sets $PP_1, \ldots, PP_m$, confidence δ, precision threshold PT, precision slack γ, slack ϵ, oracle budget T
Result: The index of a classifier or α if no qualifying or acceptable classifier is found.

$A = \{1 \ldots m\}$; `// Active set`
$PG = \{1 \ldots m\}$; `// Possibly Good set`
$RQ = \emptyset$; `// Reach Qualified set`
$h_1 = \ldots = h_m = s_1 = \ldots = s_m = S = 0$; `// counts`
$LCB_1 = \ldots = LCB_m = 0$; `// Lower Confidence Bound on precision`
$UCB_1 = \ldots = UCB_m = 1$; `// Upper Confidence Bound on precision`
while $PG \neq \emptyset$ *and* $RQ = \emptyset$ **do**
  $S++$;
  if $S > T$ **return** α;
  `SampleAndUpdateStatistics`$(A, h, s, UCB, LCB, \delta)$;
  $PA = \{i : UCB_i > PT - \gamma\}$; `// update Possibly Acceptable set`
  $PG = \{i : UCB_i > PT\}$; `// update Possibly Good set`
  $KA = \{i : LCB_i > PT - \gamma\}$; `// update Known Acceptable set`
  $KG = \{i : LCB_i > PT\}$; `//update Known Good set`

  `// Find Upper Bound on Good classifiers`
  `// Reach (`$UBGR$`)`
  for $i = 1 \ldots m$ **do**
    if $|PG| = 0 \vee (i \in PG \wedge |PG| = 1)$ **then** $UBGR_i = -\infty$;
    else $UBGR_i = \max_{j \in PG \setminus i}(UCB_j|PP_j|)$;
  end
  $RQ = \{i \in KA : LCB_i|PP_i| \geq (1 - \epsilon)\, UBGR_i\}$;
  `// Find greatest Lower Bound on Good classifiers Reach (`$LBGR$`)`
  if $|KG| = 0$ **then** $LBGR = -\infty$;
  else $LBGR = \max_{i \in KG}(LCB_i|PP_i|)$;
  `// Find Reach Disqualified classifiers`
  $RD = \{i : \max((1 - \epsilon)UCB_i|PP_i|, LCB_i|PP_i|) < LBGR\}$;
  $A = PG \setminus RD$; `// update active set`
end
if $|RQ| > 1$ **then** return best classifier in RQ;
elseif $|KA| > 0$ **then** return best classifier in KA;
else return α signifying the trivial acceptable classifier ($PP_0 = \emptyset$).;

Algorithm 1: Precision-qualified Reach Classifier Selection Algorithm

THEOREM 1 (CORRECTNESS CLAIM). *If Algorithm 1 using a statistically valid* `SampleAndUpdateStatistics` *algorithm selects a classifier then the selected classifier is acceptable and has greater than* $1-\epsilon$ *fraction of the reach of any good classifier with probability at least* $1-\delta$.

Proof sketch: The final selection criterion (the conditional after the while loop) guarantees that the algorithm will only return a known acceptable classifier (i.e., a classifier in the known acceptable set KA or the trivially acceptable classifier). This follows from the fact that the reach qualified set is a subset of KA (i.e., $RQ \subseteq KA$). The while loop will terminate only when there is a reach qualified classifier ($|RQ| > 0$) or there are no possibly good classifiers ($|PG| = 0$). In either case the reach criterion is satisfied (vacuously in the case that $|PG| = 0$). From the correctness of the anytime bounds used for determining these sets we obtain the probability guarantee. □

Theorem 1 provides a correctness guarantee for Algorithm 1 for any statistically valid implementation of `SampleAndUpdateStatistics(·)`. Ideally we would like to provide guarantees that the algorithm will terminate by selecting a classifier after obtaining a reasonable number of labels from the oracle. Such a result depends upon the specific details of the implementation of `SampleAndUpdateStatistics(·)`. The next results provide sample complexity bounds for two different implementations of `SampleAndUpdateStatistics(·)`. The two algorithms that we consider are Algorithm 2 (the `pooled-sampling-shared-label` algorithm) and the `round-robin` algorithm. In the `round-robin` algorithm, we iteratively choose a classifier c from among the currently active classifiers, randomly sample a predicted positive from PP_c, obtain the oracle label for the sampled example, and, finally, update the counts and bounds for classifier c. Note that the round-robin algorithm uses the same formula for UCB and LCB as the `pooled-sampling-shared-label` algorithm. The `round-robin` algorithm differs from the `pooled-sampling-shared-label` algorithm in both the approach to sampling examples and their use. In particular for round robin the sampled examples are exclusively used to update the performance estimates for a single classifier. It is useful for the proofs below to note that the precision bounds we use are *uniform bounds*, that is, the difference between the upper and lower confidence bounds does not depend on the actual precision but rather depends only on the total number of samples used to compute the estimated precision except in the case that the bounds are truncated by the natural upper and lower bounds on precision.

THEOREM 2 (SAMPLE COMPLEXITY FOR `round-robin`). *For T sufficiently large, Algorithm 1 using the* `round-robin` *algorithm selects a classifier using fewer than* $m \times \max(n_\epsilon, n_\gamma)$ *calls to the label oracle.*

Proof sketch: Define $n_\gamma = n_\gamma(\delta, T, PT, \gamma, m)$ to be the number of samples required for the difference between the upper and lower bounds for precision to be less than γ (e.g., $\gamma > UCB_i - LCB_i$). Because we are using uniform bounds, the same number of samples is required for each classifier and does not depend on the sampled labels. After labeling n_γ samples for a classifier, we will either know that the classifier is in KA or is not in PG. If the classifier is not in PG then the classifier will be removed from the active set of classifiers.

The definition of RD also ensures that one only throws out classifiers that are dominated in terms of their reach. Note that we define the set RD with respect to KG to ensure that we do not eliminate a good classifier based on a witness that is later thrown out due to its upper bound precision falling below PT. This is critical to ensure that the algorithm will, with high probability, return a classifier in the event that there is a good classifier.

Finally, we define $n_\epsilon = n_\epsilon(\delta, T, PT, \gamma, m, \epsilon)$ to be the number of samples required for the difference between the upper and lower bound for precision to be smaller than $\epsilon(PT - \gamma)$ (e.g., $\epsilon(PT - \gamma) \geq UCB_i - LCB_i$).

Assume that each active classifier has $\max(n_\gamma, n_\epsilon)$ labeled samples. If there are no active classifiers then the algorithm will terminate as $PG = \emptyset$. Otherwise, let s be the classifier with the highest upper bound reach of any active classifiers. From the fact that we have drawn at least n_γ samples we know that $UCB_s > PT - \gamma$. It follows from this and the fact that we have drawn at least n_ϵ samples that $LCB_s|PP_s| > (1-\epsilon)\,UCB_s|PP_s|$. From the fact that s was chosen to have the maximal upper bound reach of all active classifiers, it follows that for each active classifiers c it is the case that $(1-\epsilon)\,UCB_s|PP_s| \geq (1-\epsilon)\,UCB_c|PP_c|$. It follows that s must be in RQ and, in this case, the algorithm terminates. □

Data: The set of active classifiers A that have predicted positive sets $PP_1, \ldots, PP_m$, the current vector of counts for classifiers t, s, the current vector of upper and lower bounds for precision UCB, LCB, and δ that governs the statistical guarantee provided by the bounds.
Result: An updated set of counts and updated upper and lower precision bounds for classifiers.
$x \sim unif(\cup_{i \in A} PP_i)$; $y = oracle(x)$;
for *each* $i \in A$ *such that* $x \in PP_i$ **do**
  s_i++;
  $h_i = h_i + \mathbb{1}(y == 1)$;
  $\hat{p}_i = h_i / s_i$;
  $U_i = \sqrt{\frac{\log(2mT/\delta)}{2s_i}}$;
  $LCB_i = \max(LCB_i, \hat{p}_i - U_i)$;
  $UCB_i = \min(UCB_i, \hat{p}_i + U_i)$;
end

Algorithm 2: The `pooled-sampling-shared-label` algorithm which is one potential implementation of a `SampleAndUpdateStatistics`(·) algorithm using pooled sampling and a shared-label update. The algorithm samples from the set of predicted positives for active classifiers and updates the counts for all classifiers containing the sampled example.

THEOREM 3 (SAMPLE COMPLEXITY `pooled-sampling-shared-label`). *For T sufficiently large, Algorithm 1 using the* `pooled-sampling-shared-label` *algorithm selects a classifier using, in expectation, fewer than* $m \times \max(n'_\epsilon, n_\gamma)$ *calls to the label oracle.*

Proof sketch: In expectation, pooled-sampling will sample a predicted positive from the classifier with the largest predicted positive set with probability at least $1/m$. Similar to the analysis above, after $m \times \max(n_\gamma, n_\epsilon)$ samples, the classifier with the largest predicted positive set will either fail to be in PG or be in the set KA. In this case, however, it is harder to guarantee that this classifier is in RQ due to the potential for imbalanced sampling. To remedy this we require n'_ϵ rather than n_ϵ labels.

Consider the case in which we compare a classifier lp with high predicted positives and low precision and a classifier hp with low predicted positives and high precision. In this case, we can have $LCB_{hp}|PP_{hp}| < (1-\epsilon)UCB_{lp}|PP_{lp}|$ and $LCB_{lp}|PP_{lp}| < (1-\epsilon)UCB_{hp}|PP_{hp}|$ which means that if both of these classifiers are active (in A) then neither of them can be in RQ. The reason that this can happen is that when $|PP_{hp}| \ll |PP_{lp}|$ we are likely to draw many more samples from lp than from hp and this imbalance means that while the reach bounds for lp are reasonably tight those for hp are not. Let's consider a specific example of this type in which hp has perfect precision with test set reach of 10^x and lp has predicted positive set of size 10^y and precision 10^{x-y} for $y > x$. These two models have identical reach. Unfortunately, as y grows the variance in our reach estimate grows and it become harder and harder for either to become a member of RQ. We have a bound on the precision. In particular, we know that $10^{x-y} > PT - \gamma$. One simple way to obtain a bound for shared sampling is to use the ratio of the sized of predicted positive sets. In particular, let classifier $a_{\max}$ be the acceptable classifier with the largest predicted positive set and let classifier $a_{\min}$ be the acceptable classifier (possibly a good classifier) with the smallest predicted positive set. If we define n'_ϵ to be $n_\epsilon \frac{|PP_{a_{\max}}|}{|PP_{a_{\min}}|}$ then, after we sample mn'_ϵ samples using our pooled-sampling we will, in expectation, label n_ϵ predicted positives from classifier $a_{\min}$. In this case, if we choose the active classifier with maximal upper bound reach it will be in RQ.

In order to return a classifier with high probability, our results require that T is sufficiently large. We note that the value of T required is a function of PT, ϵ, γ, m, and δ. If desired, one can compute the minimum value of T by solving a fixed-point equation. □

Note that Theorem 2 shows the number of samples needed for round robin is linear in the number of classifiers. For Theorem 3 we show a similar statement for pooled sampling with shared labels although the bound is somewhat looser. Despite this, in the empirical analysis in Section 5, we will see that the performance in practice of pooled sampling with shared labels is often much better than round robin.

5. EXPERIMENTS

To study the empirical performance of our algorithm we evaluate several variants of the algorithm that give rise to natural baselines in the setting of model selection for topical text classifiers for web pages. Since our algorithms apply to any classification setting, we chose to focus on topic classification since it is both well studied and it allows us to select a publicly available dataset for reproducibility.

5.1 Data

To evaluate the model selection properties of the algorithms, we use all 15 topical top-level categories of the *Open Directory Project* (ODP) web hierarchy.[1] We selected ODP as a representative set of topical classification tasks whose top-level categories vary in both frequency of positives (from 1.8% for Home to 15.5% for Arts) as well as classification performance – two varying qualities that might be expected to impact model selection. For each of these topics we independently trained a bag-of-words binary logistic regression model on the content crawled from URLs in the hierarchy. Any URL belonging to the topic or to a descendant of the topic is considered a positive example for that topic while any URL which does not belong to the topic or its descendants is considered a negative. One typical model selection problem is choosing what threshold should be used when deploying a classifier for a particular performance constraint. We use this as the setting for our classifiers and implicitly generate thresholds by considering a set of 9 classifiers for each topic. Although we note that our algorithm applies to other more general model selection problems this helps us ensure that we have classifiers at a variety of precision and reach levels while at the same time studying a common application setting for model selection. The 9 classifiers for a particular topic are generated by ranking the entire test set by the prediction of the classifier and then predicting the top n to be positive. The minimal value of n considered was $n = 50$ and n was doubled thereafter. That is, the 9 classifiers for a topic would predict the top $n = \{50, 100, 200, 400, 800, 1600, 3200, 6400, 12800\}$ were positive.[2] For each topic, the algorithms select which threshold would yield the best reach subject to the precision constraints.

5.2 Algorithms

We investigated several variants of our algorithm that correspond to natural baselines and help illustrate the key aspects of our algorithm. We vary the choice of the SampleAndUpdateStatistics algorithm between a round-robin algorithm and the pooled-sampling-shared-label algorithm (Algorithm 2). This enables us to determined the impact on reduced labeling cost obtained through the pooled sampling.

We also vary the choice between using model elimination and no model elimination. Model elimination means that models are eliminated after each sample if possible because either their precision is outside of constraints or their reach is below another precision-qualified candidate. No model elimination means that the set of active classifiers A is always equal to $\{1 \dots m\}$. This helps us determine to what extent reduction in labeling cost is due to identifying models to eliminate early.

We also consider using early stopping or not. Early stopping stops sampling labels as soon as a candidate that can be guaranteed to meet all constraints is found whereas no early stopping always exhausts the full budget. This variant enables us to more directly measure how quickly we can determine when to stop in comparison to the same (or a similar algorithm) using bounds based on the same total budget.

Finally, we consider how costly it is to require statistical guarantees, and we employ a guessing variant that selects the current maximizer of reach with acceptable precision even if a statistical guarantee cannot be given for it. While in general most production systems would desire a statistical guarantee, this enables understanding how often a heuristic is right that spends a given budget and then guesses if statistical significance has not been reached.

We now give a full description for each algorithm as well as an abbreviation to refer to it in brief. Our primary "full" algorithm is variant 7 below. Given these design choices we then have the following algorithms:

[1] www.dmoz.org

[2] Note that once a model is selected the prediction threshold value for a particular classifier can be generated by taking the average score of the n and $n+1$ example for the value of n for that classifier.

1. RR-NES-NME-NG: Round Robin, No Early Stopping, No Model Elimination, No Guesses
   Round robin is the most basic and straightforward baseline to consider when sampling. The algorithm rotates through the classifiers and independently samples a predicted positive for the current classifier and updates the performance estimates for that classifier. The algorithm continues until the labeling budget is exhausted and then selects a model if one can be selected while providing statistical guarantee or otherwise outputs α. This baseline indicates what the straightforward approach can do while maintaining a statistical guarantee.

2. RR-NES-NME-G: Round Robin, No Early Stopping, No Model Elimination, allow Guesses
   This approach is like the previous, but a model that is identified as the likely maximizer of reach with acceptable precision can be selected even if a statistical guarantee cannot be made. It represents the setting where a fixed budget will be spent on labeling and then a choice would be made based on available data.

3. RR-ES-NME-NG: Round Robin, Early Stopping, No Model Elimination, No Guesses
   Like approach 1, this approach applies independent sampling but adds the ability to stop early if a model is found for which precision and reach can be statistically guaranteed before the labeling budget is exhausted.

4. RR-ES-NME-G: Round Robin, Early Stopping, No Model Elimination, allow Guesses
   Like the previous approach (3) but the best guess can be made for a model if the labeling budget is exhausted and no statistical guarantee can be made.

5. RR-ES-ME-NG: Round Robin, Early Stopping, Model Elimination, No Guesses
   Like approach 3, this approach applies independent sampling and early stopping but also adds the ability to eliminate models whose precision is bound to be outside of acceptable or whose reach is known to be outside of the reach bounds relative to a classifier whose precision is now bound to be acceptable. This approach is meant to be the most competitive baseline with our full model (described below).

6. RR-ES-ME-G: Round Robin, Early Stopping, Model Elimination, allow Guesses
   Like the previous approach (5) but the best guess can be made for a model if the labeling budget is exhausted and no statistical guarantee can be made.

7. PSSL-ES-ME-NG: Pooled Sample with Shared Label, Early Stopping, Model Elimination, No Guesses
   This is essentially our "full" algorithm. It always provides a statistical guarantee if a model is selected and indicates the lack of a guarantee by returning α.

8. PSSL-ES-ME-G: Pooled Sample with Shared Label, Early Stopping, Model Elimination, Allow Guesses
   This is the version of our "full" algorithm that may not always provide a statistical guarantee if the budget is exhausted before a model is selected.

5.3 Experimental Methodology

To simulate a realistic setting we set the precision and reach constraints to what we felt to be similar to operational settings based on experience. We set the precision threshold, $PT = 0.90$ to indicate the model must meet a high precision threshold. We set the precision slack to $\gamma = 0.1$. Taking the precision slack of 0.1 together with the precision threshold, this would be representative of scenarios where the desired classifier has precision 0.80 or better. For the confidence, $1 - \delta$, we vary $\delta \in \{0.05, 0.1, 0.2, 0.95\}$ where lower values of δ indicates higher confidence is desired. We focus on $\delta = 0.05$ since this value is often used for statistical significance. For the reach slack we set $\epsilon = 0.1$ since we assume a setting where an approximate maximizer suffices.[3] We then set the upper budget on labeling to $T = 5000$ to indicate a scenario where the user is willing to expend labeling budget for the right answer but still prefers to minimize total labeling budget. Finally, to account for randomness in the sampling, we re-run each procedure with different seeds to produce 250 different runs and report the averages and percentages over these runs.

5.4 Performance Measures

In terms of performance we focus on label efficiency and how often the algorithm produces an "acceptable answer". A selected model is an acceptable answer if its true precision and reach when evaluated over the full test corpus meets the requested constraints. If the selection algorithm produces the answer α (the trivially acceptable classifier), this is only considered acceptable when there is no classifier that meets the requested precision constraint.

5.5 Results & Discussion

Figure 1 presents the labeling cost of each method on the y-axis as a function of the confidence $(1 - \delta)$ on the x-axis.[4] At all levels of confidence, pooled sampling with shared labels (PSSL) requires less than half as many labels as the other approaches – often it requires less than 20% of the naïve round robin approach. Furthermore, as expected it requires a decreasing amount of labels as the requested confidence in the guarantee decreases (i.e., moving toward the right on the x-axis). In contrast, the other methods either nearly or completely exhaust the full labeling budget in the majority of topics and settings.

Labeling budget does not express the full story, however, because we also wish to know that the algorithms select an acceptable answer when they terminate. The percentage of times an acceptable answer is selected out of all 250 random restarts is shown in Figure 2. We see that PSSL always produces acceptable answers with guarantees. We see that the round robin methods can usually but not always *guess* an acceptable answer but vary rarely produce a statistical guarantee. While it may seem encouraging that they guess an acceptable answer, the reader should keep in mind that they typically have used 2x-5x more labels as well!

[3]Space prevents us from presenting results varying the slack variables as well although we have conducted these experiments and found our method performs well across a variety of settings.

[4]Only the "no guesses" variants are displayed for Figure 1 since for these graphs performance between the "guess" and "no guesses" variants are the same.

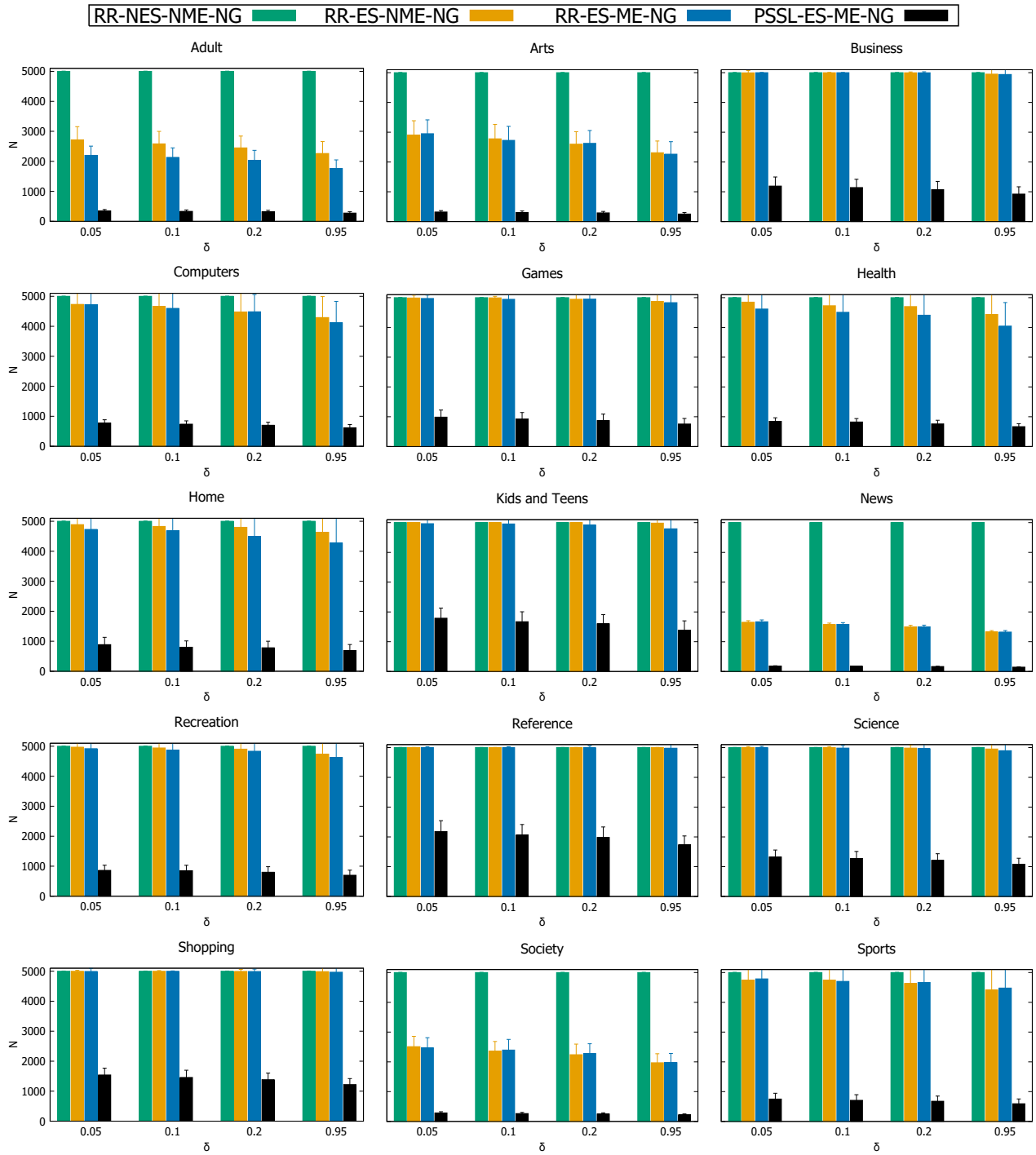

Figure 1: Number of labels needed on average vs δ (= 1 - confidence) over 250 repetitions in ODP top-level classes. As δ increases (and the confidence requested decreases) our method requires few labels.

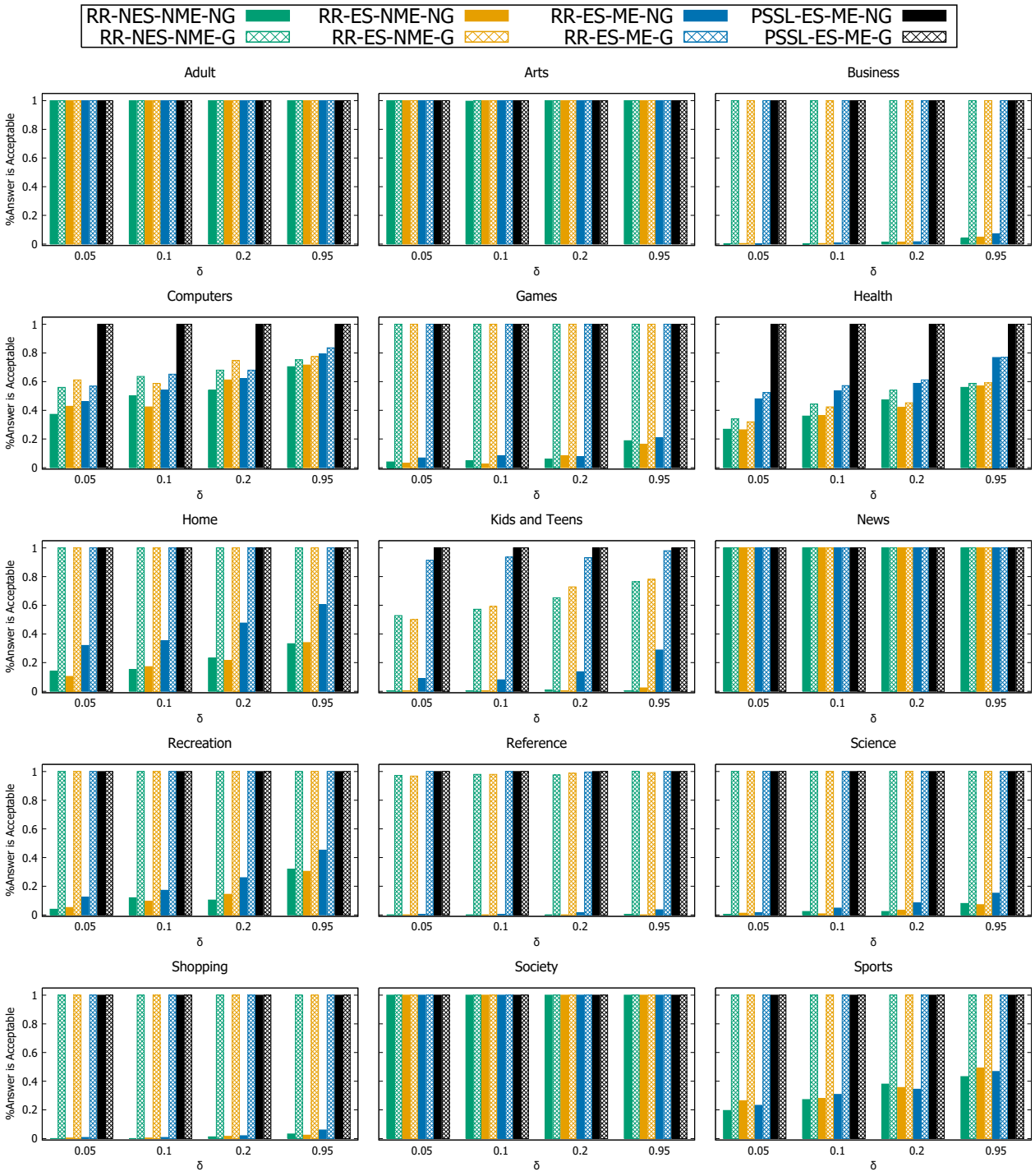

Figure 2: Percentage of acceptable answers vs δ over 250 repetitions in ODP top-level classes. Our method both guesses (black pattern fill) an acceptable answer in all runs and makes statistical guarantees (black solid) in all runs. While most other methods guess an acceptable answer (pattern fill) they cannot make statistical guarantees (solid fills) except for weak confidences (increasing δ) and typically require 2x-5x more labels.

6. RELATED WORK

There is a body of work on adaptively evaluating the performance of classifiers. For instance, Bennett & Carvalho [3] use adaptive importance sampling to reduce the number of samples required to estimate the precision of classifiers. Similarly, Sawade *et al.* [9] use an active evaluation approach for estimating the F_α-measure (an unthresholded tradeoff between precision and recall) of a given model.

From the perspective of model selection, there is wide agreement that, for many use scenarios, the selection criteria for classifiers requires a tradeoff. This can be seen in the use of receiver-operator-characteristic (ROC) curves and precision-recall curves. We focus on the precision-recall tradeoff because this tradeoff has advantages for skewed data [5]. As we argue, reach may often be more suitable than recall (see [10] for additional supporting arguments).

Similar to our paper, Arasu *et al.* [1] consider optimizing reach subject to a precision constraint for the problem of record matching. Their algorithm, unlike ours, makes a monotonicity assumption on the precision and recall of the classifier over its parameter space. Furthermore, it can require a large number of samples [2] and does not provide statistical guarantees. Bellare *et al.* [2] use a black-box approach leveraging existing active learning algorithms to choose models on the basis of recall given a precision constraint for the problem of entity matching. Because they use a black-box approach, the guarantees provided are not as strong as those that we present in this paper and apply to recall rather than reach.

More broadly, there is a large body of work on adaptive design and adaptive estimation. This includes work on Bayesian clinical trials [4], adaptive dose-finding designs [8], and active learning. In several respects, our work is similar in spirit to and draws inspiration from the work of Even-Dar *et al.* [6] who provide bounds on the sample complexity of finding ϵ-optimal solutions (solutions that are within ϵ of the best arm). The parameter ϵ can be thought of as a "slack value" that enables the authors to provide sample complexity bounds that do not depend upon the statistical properties of the arms of the bandit. Thus, the required number of samples only depends upon the desired guarantees and the choice of slack variable. In several of our algorithms, we leverage a similar approach, but we use a *relative* rather than additive error (for reach in our case) that makes more sense for our application. In addition, we add a second "slack variable" to handle our precision threshold. This enables us to obtain sample guarantee for several of our algorithms that, analogous to [6], do not depend upon the statistical properties of the classifiers provided.

7. CONCLUSION

In this work, we described how the problem of selecting a high recall classifier subject to precision constraints can be reduced to the problem of selecting a high *reach* classifier with the same precision constraints. The switch from maximizing recall to reach is a subtly important one that relies on the observation that the denominator of recall is constant across all models being evaluated and that it is this denominator which greatly increases the cost of accurately estimating recall. We then develop several algorithms for efficiently (in terms of label cost) selecting a high reach classifier that has acceptable precision. Our framework is flexible and enables the user to specify both hard bounds and slack on the precision and reach conditions. Furthermore, we provide statistical guarantees for the selected classifier and sample complexity guarantees for our algorithms. Empirical comparison of our algorithm to other approaches demonstrate that our algorithms can significantly reduce the number of samples required to select classifiers while providing guarantees on the selected model's classification performance.

References

[1] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 783–794, New York, NY, USA, 2010. ACM.

[2] K. Bellare, S. Iyengar, A. Parameswaran, and V. Rastogi. Active sampling for entity matching with guarantees. *ACM Trans. Knowl. Discov. Data*, 7(3):12:1–12:24, Sept. 2013.

[3] P. N. Bennett and V. R. Carvalho. Online stratified sampling: evaluating classifiers at web-scale. In *In CIKM*, pages 1581–1584, 2010.

[4] B. P. Carlin, J. B. Kadane, and A. E. Gelfand. Approaches for optimal sequential decision analysis in clinical trials. *Biometrics*, 54(3):964–975, 1998.

[5] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[6] E. Even-Dar, S. Mannor, and Y. Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, COLT '02, pages 255–270, London, UK, UK, 2002. Springer-Verlag.

[7] Q. Guo, R. W. White, Y. Zhang, B. Anderson, and S. Dumais. Why searchers switch: Understanding and predicting engine switching rationales. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 334–344, 2011.

[8] A. Hirakawa, N. A. Wages, H. Sato, and S. Matsui. A comparative study of adaptive dose-finding designs for phase I oncology trials of combination therapies. *Statistics in Medicine*, 34:3194–3213, 2015.

[9] C. Sawade, N. Landwehr, and T. Scheffer. Active estimation of f-measures. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2083–2091. Curran Associates, Inc., 2010.

[10] J. Zobel, A. Moffat, and L. A. Park. Against recall: Is it persistence, cardinality, density, coverage, or totality? *SIGIR Forum*, 43(1):3–8, June 2009.