Online Active Inference and Learning

Josh Attenberg
Polytechnic Institute of NYU
Brooklyn, NY
josh@cis.poly.edu

Foster Provost
NYU Stern School of Business
New York, NY
fprovost@stern.nyu.edu

ABSTRACT

We present a generalized framework for *active inference*, the selective acquisition of labels for cases at prediction time in lieu of using the estimated labels of a predictive model. We develop techniques within this framework for classifying in an online setting, for example, for classifying the stream of web pages where online advertisements are being served. Stream applications present novel complications because (i) at the time of label acquisition, we don't know the set of instances that we will eventually see, (ii) instances repeat based on some unknown (and possibly skewed) distribution. We combine ideas from decision theory, cost-sensitive learning, and online density estimation. We also introduce a method for on-line estimation of the utility distribution, which allows us to manage the budget over the stream. The resulting model tells which instances to label so that by the end of each budget period, the budget is best spent (in expectation). The main results show that: (1) our proposed approach to active inference on streams can indeed reduce error costs substantially over alternative approaches, (2) more sophisticated online estimations achieve larger reductions in error. We next discuss simultaneously conducting active inference and active *learning*. We show that our expected-utility active inference strategy also selects good examples for learning. We close by pointing out that our utility-distribution estimation strategy can also be applied to convert pool-based active learning techniques into budget-sensitive online active learning techniques.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—data mining; I.2.6 [**Artificial Intelligence**]: Learning—induction; I.5.1 [**Pattern Recognition**]: Models—statistics

General Terms: Algorithms, Design, Human Factors

Keywords: active inference, machine learning, active learning, on-line advertising, micro-outsourcing

The authors thank AdSafe Media for data and support. Foster Provost thanks NEC for a Faculty Fellowship.

KDD'11, August 21–24, 2011, San Diego, California, USA.

1. INTRODUCTION

When making decisions under uncertainty with data-driven models, we often have the option of directing (costly) human resources to help improve the process, for example by hand-labeling carefully selected data instances. Active learning methods try to select the instances to label that will best improve the modeling for a given cost. In this paper we study a complementary problem. *Active inference* involves carefully selecting instances to label at the time of use of a predictive model ("prediction time" or "inference time").

In many applications it is desirable to incur some cost to improve decision making by bypassing the model's decisions in favor of a human's. This choice may be a result of the model's uncertainty about a particular case, or alternatively, because assigning a label to a certain case will impart some improvement on other decisions. As a first contribution of this paper, we present a general framework for such *active inference*. This framework has prior approaches for inference-time label acquisition as special cases. The main generalizations of prior settings are threefold: (i) instances can be drawn with replacement from some distribution;[1] (ii) there may be a limited budget for labeling,[2] and (iii) the set of instances that will eventually be seen may not be known when a particular labeling decision needs to be made.

As our main application we will consider learning and inference on data streams, a problem setting which can exhibit all three of these characteristics. As a motivating example consider the problem of building classifiers for "safe" online advertising: helping advertisers to control the content adjacent to their advertisements. Certain categories of objectionable content such as hate speech and pornography are at odds with the carefully crafted images associated with most brands. Given a stream of impression opportunities, a safe advertising system is tasked with classifying each page as acceptable or objectionable, with the goal of preventing ads from occurring on objectionable pages. Figure 1 presents a typical distribution over web page occurrences in a production safe advertising system. In such a highly skewed setting, a few pages appear extremely frequently in the impression stream. However, most pages occur very infrequently. Given the extreme sensitivity to objectionable content, as well as to large numbers of falsely blocked good web pages, it is clear that not all pages should warrant equal effort—some pages may be sufficiently sensitive or frequent to have their own

[1]In practice, this distribution may be far from uniform.
[2]The budget constrains the number of examples that may be labeled.

hard-set ground-truth labels (within some labeling budget). However, at any point in time the set of pages to-be-seen is unknown, a scenario that is especially likely upon the introduction of a new impression stream to the advertising system. Note that these properties are not exclusive to the safe advertising problem: similar difficulties may be faced by classification problems in web search, spam detection, online ad targeting, error detection in complex systems, and others, where instances for prediction appear in streams with repetition.

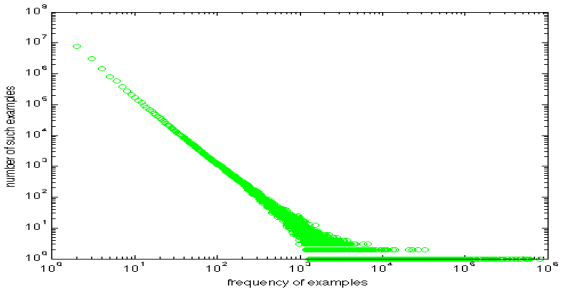


Figure 1: A histogram of impression frequencies of web pages sampled from a single day's ad traffic. Such skewed distributions on page views are typical in online advertising.

Ideally, one would like to apply the limited labeling budget to the most "useful" examples appearing in a stream. However, this task is complicated by the necessity of performing various estimates, and combining these into a higher-level estimation. The estimates include the expected per-example benefit from acquiring a ground truth label and using this label as a substitute for the model's predictions, and the expected number of times the example will appear in the future. More subtly, on top of this we also have to estimate the distribution of *utilities* that we will see, so we can plan how to spend the budget. The resulting model tells which instances to label so that by the end of the budget period, the budget has been best spend (in expectation).

We present a solution to this active inference problem using a framework for online utility-distribution estimation ("UDE") and test this method on eight different classification problems from a real streaming classification application. The main results show that: (1) active inference on streams can indeed reduce error cost substantially over not doing the online estimation, and (2) more sophisticated online estimation provides more reduction in error.

To close the paper we discuss relationships with active learning: What if you need to learn the model at the same time you are doing the active inference? In our original workshop position paper on active inference [1], we conjectured that a technique like the one we present and analyze below (which we had not yet designed completely or implemented) not only would be an effective active inference strategy, but also would be a potentially effective online active *learning* strategy for the online setting, and therefore that it should provide a strong baseline against which more sophisticated AI+AL strategies can be compared. We now provide initial supporting evidence.

As a final, suggestive, contribution we also discuss how the overall online utility estimation framework provides a new way to look at *online* active learning: it can be used to take any active learner that selects examples based on scoring each example in a pool (as most active learners do), and convert it to an online or stream-based active learner.

2. ACTIVE INFERENCE

Similar to work on active learning, this paper assumes that at a cost we can acquire accurate label information on selected cases. The difference is that here we acquire labels at prediction time, enabling the acquisition of "ground truth" labels as a supplement or substitute to the predictions of error-prone statistical models. The objective of this "active inference" is to reduce the total cost incurred by the predictive system. This section provides a framework for active inference for classification tasks. This framework generalizes prior work on prediction time label acquisition, including active inference for collective classification and techniques for classification with a reject option. Later, we develop techniques within this framework for performing online active inference (from streams) as another special case with its own unique characteristics and complications.

At first it might seem that prediction-time acquisition of training labels would not make sense: if labels are available for the instances being processed, then why perform potentially error-prone statistical inference in the first place? While "ground truth" labels are likely to be preferable to statistical prediction, we cannot ignore the cost/benefit context. Acquisition of ground truth labels may be more costly than simply making a model-based prediction, and there may be a limited budget for labeling.

The classification task can be formalized as: given a set of n discrete classes, $c^j \in \mathcal{C}, j = 1 \ldots n$, instances, x_i, are drawn from some distribution, each with an associated class label $y_i = c \in \mathcal{C}$, drawn according to some $p(y = c|x)$. A classifier estimates this hidden class value, y_i, ideally in a way that minimizes some cost function, $\text{cost}(c^k|c^j)$—the penalty for predicting $y_i = c^k$ when in fact the true label is c^j [14]. Consider a model capable of predicting the posterior probability distribution over $\mathcal{C}$ conditioned on a given x_i, $\hat{p}(y_i = c|x_i) = f(x_i)$.[3] Based on this estimated posterior distribution, one can choose a particular classification $\hat{y}_i$ which will minimize an *expected* misclassification cost (or *loss*):

$$L(x_i, \hat{y}_i) = \sum_{c' \in \mathcal{C}} \hat{p}(y_i = c'|x_i)\text{cost}(\hat{y}_i|c')$$

More generally, an instance x_i may be presented repeatedly. Let $\phi(x_i)$ denote the number of times a given example x_i appears during a particular period of prediction time, the embodiment a sequence of draws *with replacement* from some distribution $p(x_i)$.[4] For a set of instances $\mathbb{T}$ that would be encountered by a classifier during this time period, the

[3]For instance in the case of our motivating example, given a web page, estimate the probability that page is objectionable.

[4]For this paper we will simplify and consider x_i to be a unique case that repeats verbatim in the stream, such as a particular web page, a particular query string in web search classification, a particular spam email, etc. This is different, for example, from two distinct web pages i and j that have the same feature representation. On one hand, this unique-case condition can be enforced trivially whenever an identifier or key exists (or can be generated) for the instance, such as the web page's URL or the hash value of the text of an email, which can be included in x_i. On the other

expected misclassification cost is:

$$\mathcal{L}_{\mathbb{T}} = \sum_{x_i \in \mathbb{T}} \phi(x_i) \min_c \sum_j \hat{p}(c^j|x_i)\text{cost}(c|c^j) \quad (1)$$

Note that for the pool-based setting typical in the literature (e.g., in most cross-validation experiments) $\phi(x_i) = 1$ for all x_i and can thus be ignored.[5]

Given a budget B for acquiring instance labels during a given time period, a predefined cost function, $\text{cost}(c^k|c^j)$ and a given cost structure, $q(x)$, revealing the expense of acquiring the label for each instance at prediction time, the objective of a general active inference strategy is to select a set of examples for which to acquire labels, $\mathcal{A}$, such that the expected *total* classification cost is minimized, adhering to the budget constraints:

$$\mathcal{A} = \arg\min_{\mathcal{A}' \subset \mathbb{T}} \mathcal{L}_{\mathbb{T} \setminus \mathcal{A}'} + \sum_{x_i \in \mathcal{A}'} q(x_i) \quad (2)$$

$$\text{s.t. } B \geq \sum_{x_i \in \mathcal{A}'} q(x_i)$$

The idea that we may defer the prediction task for certain cases to a human expert has been studied extensively as "classification with a reject option." This setting tends to focus on the balance between the expected misclassification cost, $L(x_i, \hat{y}_i)$, and the cost associated with "rejecting" the inferred classification, $q(x_i)$.

In the simplest case, imagine an unlimited budget for label acquisition, where no repetition occurs, and the labels of all examples are independent. Further assume symmetric error costs; w.l.o.g., $\text{cost}(c^k|c^j) = 1$ when $c^k \neq c^j$, with a cost of 0 otherwise. In this case, it is straightforward to show [10, 9] that the optimal "reject" policy, the set $\mathcal{A}$ offering a minimum reject rate for a given expected error probability (or, equivalently, minimizing the expected error probability for a given reject rate), is given by:

$$\mathcal{A} = \left\{ x_i \| \min_c \hat{p}(y_i = c|x_i) > q(x_i) \right\} \quad (3)$$

As would be expected, very large query costs tend to obviate the usefulness of the reject option; indeed the reject option would never be exercised when $q(x_i) > \frac{1}{2}$. (It it optimal to always query the oracle when $q(x_i) = 0$.) Of course the uniform misclassification costs assumed above are seldom realistic. Extending the reject rule of Chow to the case of asymmetric misclassification costs, Herbei and Wegkamp [20] show that the optimal $\mathcal{A}$ is given by:

$$\mathcal{A} = \left\{ x_i \| \min_{\hat{y}} L(x_i, \hat{y}_i) > q(x_i) \right\} \quad (4)$$

To our knowledge, classification with a reject option has not been extended to the stream setting. Observing examples sequentially presents several challenges to an active inference strategy. Instances may repeat, thereby multiplying otherwise small individual expected losses into significant cumulative penalties. Perhaps more difficult, the set of possible examples is unknown, and the distribution from which instances are drawn must also be estimated. Furthermore, a finite budget must be managed over time; it is suboptimal simply to use the thresholds described above, because the budget may be exhausted early, on low-margin cases. The development we present below might be seen as an on-line, stream-based, budget-sensitive version of the reject option. However, the repeating of examples might stretch the "reject" idea: rather than rejecting cases that (we believe) will show up in the future, we *invest in them*.

The active inference framework presented in Equation 2 also is a generalization of a framework presented previously [6] in the context of *collective* active inference [22, 6, 7, 8]. When examples are interrelated (e.g., in a network), collective inference may take advantage of relational autocorrelation in the labels to improve predictive performance beyond that achieved by treating the instances as i.i.d.[6] Since with collective inference, inferred labels affect each other, errors in inference can propagate. Thus, if one has a budget for human labeling at inference time, it may be spent on carefully selecting the examples to label such that the collective generalization performance is maximally improved [22].

In the active inference model we have presented so far (more will come), the generalization over the model of [6] is the addition of $\phi(x)$. In many applications we repeatedly make decisions about some instances; whether or not we take $\phi(x)$ into account can substantially change the cases we would want to label, when the distribution is not uniform. We are not extending the prior active inference methods in this paper; the on-line, stream setting that we introduce next requires a different set of techniques from the collective, network setting. A very interesting line of future work would be to find solutions for the combined setting. For example, the web pages that arrive in a stream are indeed interlinked in a network that exhibits relational autocorrelation in the class variable (objectionable pages are more likely to be linked to other objectionable pages, cf., [12]).

3. ACTIVE INFERENCE ON DATA STREAMS

A main contribution of this paper is the extension of active inference to the data stream setting. Here, active inference has a unique set of challenges that are beyond the capabilities of current methods. The primary differentiating factor of online active inference is that examples are typically drawn *with replacement* from some (unknown) process, $p(x)$.

In practical applications like web classification, $p(x)$ often exhibits a highly skewed, power-law-like distribution (cf., Figure 1). Due to this instance repetition, misclassification costs multiply; particular examples with a small individual expected loss may becoming quite costly over time. In the pool-based setting typically discussed in the machine learning literature, $\phi(x)$ would be known and fixed. In such cases, the strategy presented in Equation 2 may be applied directly.

However, the primary use case we are concerned with in this paper is when $p(x)$ is unknown—such as in the context of classifying a stream—requiring estimating it from

hand, this simplification presents limitations. For example, we may want to consider a webpage to be the same webpage even if there are minor modifications to the text that cause the feature vector to change slightly. Similarly, in other applications, there may be an entity that repeats, but with different feature vectors (e.g., different transactions from an entity that is either fraudulent or not), and we want to make decisions at the entity level, not the transaction level [15]. We leave the generalization of the setting to future work.

[5]Or, given the unusual case of a multiset, a pool of examples that exhibits repetition, $\phi(x)$ can be computed exactly by simply counting occurrences in the data.

[6]This networked inference setting is related to work on selecting a subset of sensors to activate in a sensor network [17].

the data and continually updating it during the inference process to ensure/maintain accuracy. We describe how we will do this in Section 3.2, leaving a thorough evaluation of online density estimation and its interplay with active inference for future work.

It is important to note that the notion of "density" presented by having to estimate $p(x)$ is different from the notion of "density" employed in "density-senstive" active learning (e.g., [26, 21]). In the former case (this paper), $p(x_i)$ represents the probability of "drawing" x_i from the data-generating process (d.g.p.). In the active learning work, the "density" corresponds to the likelihood of drawing from the d.g.p. *other* examples similar to x_i, so that choosing x_i for learning will be worthwhile. We are not aware of active learning work that explicitly considers $p(x)$. In turn, in this paper we do not consider how the more general geometry of the problem space might be taken into account for active inference. It is an open question as to what problem characteristics would induce the rate of incidence of examples (e.g., in a stream), $p(x)$, to be correlated to the similarity between the examples themselves.

A second complication of the stream setting is that the (reduction in) cost associated with a particular x_i must be extrapolated into the future, and appropriately discounted. A third, related complication is that we will need a framework relating the budget to the stream (and to the discounting). Do we have a fixed budget for the future (foreseeable or not)? Do we have a budget-per-unit time? This of course will be application dependent. For the development of the rest of the paper, we deal with these two related complications by assuming that we are given a budget for a given time period (or a budget-per-unit-time), and that we can ignore discounting: either because we really are most concerned with this immediate time period (a "square-wave" discount function), or because the discounting affects $p(x)$ uniformly, so weighting by $p(x)$ implicitly deals with the discounting. In our experience, having a budget for a particular time period is a typical application setting. For example, a business may budget so many dollars per month for human labeling of web pages. Next month there will be a new (possibly different) budget. We assume for the rest of this development that we know enough about the rate of seeing examples over the budget period that we can directly translate $\hat{p}(x_i)$ to $\hat{\phi}(x_i)$, the estimated frequency of seeing example x_i over the budget period.

A fourth and more insidious complication is that in the stream setting we do not actually know $\mathbb{T}$, the set of x_i's that we will see over a particular time period, nor even the total set of (realisable) x_i's that we might actually see. If x_i is a web page described by a bag-of-terms representation (for example), we certainly don't expect to see every possible x_i. Thus it is awkward, and may be ineffective, to treat $\mathcal{A}$ simply as a set of examples (as we could in the pool setting discussed briefly above). We would like to take the more general notion of $\mathcal{A}$ being a decision strategy that will incorporate $\hat{\phi}(x_i)$ and $\hat{p}(y_i = c|x_i)$ to produce a decision whenever an x_i presents itself: should we spend some of our budget to acquire its label? We now discuss this in greater depth.

3.1 Utility Distribution Estimation

Given the cost structure presented above, the estimated expected benefit of acquiring y_i for instance x_i is

$$\hat{\mathcal{U}}(x_i) = \hat{\phi}(x_i)L(x_i, \hat{y}_i) - q(x_i) \tag{5}$$

As with online estimation of $\hat{p}(x)$, we can add another layer of estimation, and treat $\hat{\mathcal{U}}(x_i)$ as a random variable upon which we can induce a density estimate. Let $\hat{\psi}(\hat{\mathcal{U}})$ be our estimated probability (density) function over the different possible expected utilities for the various x_i drawn in accordance with $p(x)$.

Now we can formulate a proposed general active inference acquisition strategy: label all x_i for which $\hat{\mathcal{U}}(x_i) \geq \tau$, that is, label all instances with a sufficiently high estimated utility. This threshold should be set in such a way as to exhaust the budget per epoch, in expectation:

$$\tau = \arg\max_{\tau'} \int_{\tau'}^{\infty} \hat{\psi}(\hat{\mathcal{U}}) d\hat{\mathcal{U}} \tag{6}$$

constrained such that $\int_x p_1(x)\mathbb{I}_{\tau'}(x)q(x) \leq B$. Here $p_1(x)$ is the probability of seeing the argument at least once. If we expect N observations during the the budget period, we could estimate $p_1(x) = 1 - (1 - p(x))^N$, making the simplifying assumption that each subsequent draw is conditionally independent from other observations in the stream, given $p(x)$. Here $\mathbb{I}_{\tau'}(x)$ is 1 whenever $\hat{\mathcal{U}} \geq \tau'$, and 0 otherwise. Although Equation 6 could be simplified to choosing the minimum τ after each observation, leaving the rest implicit, the presented form illustrates the notion of a distribution of utilities of observed examples, from which we would like to choose the upper tail. We call such a utility-thresholding strategy *online utility-distribution estimation*. We will call this active inference strategy based upon online utility-distribution estimation *Active Inference with Utility Distribution Estimation*, or simply AI-UDE.

3.2 Online Density Estimation

Returning briefly to the problem of estimating $\hat{p}(x)$ on the fly, let us note that while appropriately choosing an estimatation model for $\hat{p}(x)$ is certainly critical to the performance of an active inference strategy, in this paper we leave a thorough evaluation of online density estimators and their associated influence on active inference for future work. Instead, we present two techniques as a baseline. Hopefully it will be clear that improved density estimation will only make the active inference techniques better. The two techniques considered in this paper are:

Dirichlet Multinomial (DMN) The simplest density estimation technique for this problem is a maximum likelihood estimation for the multinomial distribution over the instances. As this will over-estimate the probability on those cases that we observed more frequently just by chance, we instead use the posterior of a Dirichlet-multinomial distribution with a uniform prior. Specifically, we use the typical "add-α" smoothing: $p_\alpha(x_i) = \frac{f_{x_i}+\alpha}{\sum_j f_{x_j}+N\alpha}$, where N is the number of examples seen. For all experiments in this paper, we set $\alpha = 1$.

Good-Turing (SGT) The main drawback of methods like DMN is that they assume to know $\mathbb{T}$ at any point in time. As described above, we don't know what instances we will see, nor the total space of realisable instances. Especially with long-tail distributions, these instances could make up a substantial portion of the probability mass. Multinomial methods will thus overestimate the actual $p(x_i)$ and $\phi(x_i)$ for the already seen x_i's.

Good-Turing methods [18] explicitly account for the unobserved probability mass when estimating the probabil-

ity/frequency for each example observed at least once. Specifically, $\hat{\phi}(x_i) = (f_{x_i} + 1)\frac{E(n_{f_{x_i}+1})}{E(n_{f_{x_i}})}$, where n_{f_x} represents the number of examples observed f_x times; the height of the corresponding bin in a histogram. Good-Turing techniques are a family of smoothing estimators for computing estimating the quantity $E(n_{C(\cdot)})$.

Here we follow the Simple Good Turing (SGT) method [11] where to estimate $E(n_{C(\cdot)})$ an interpolation is performed on the empirical histogram, fitting a least-squares regression on the log-log scale. For a complete explanation, see [16].

4. EXPERIMENTAL SETUP

To demonstrate the effectiveness of active inference we compared methods on real-world stream data from the domain of safe online advertising. These data sets consist of placement opportunities (web pages) for ads observed in a stream. For experimentation, the pages were labeled along four distinct "safety" categories: adult content, hate speech, inappropriate alcohol content, and content related to illegal drugs. Each category poses a distinct learning and inference challenge, all with differing levels of skew with base rates ranging from $10,000:1$ to $1,000,000:1$ in the data stream.

Taking the cross product of four classification tasks with two data streams, we consider eight distinct brand safety sub-problems. The use case considered thus far involves the application of an existing (pre-trained) predictive model to the problem of example selection for inference. To facilitate this in an experimental setting, each category utilized a model trained using a separate, held-out, labeled data set, ensuring consistency across all active inference methods considered. In each case, all methods are given a burn-in period of $200,000$ unlabeled observations, allowing the density estimators to get some rough idea of the nature of the distribution under consideration.[7] After this burn-in, the density estimators continue to "learn," incorporating each new observation into a refined estimator for $\hat{\phi}(x)$.[8] Note that the burn-in period of $200,000$ is somewhat arbitrary: a larger or smaller burn-in does not alter the qualitative conclusions presented herein. However, the rates of divergence for the techniques compared varies. A cost ratio of $10,000:1$ is used, capturing the very severe consequences of having advertisements appear adjacent to inappropriate content. The data were drawn from two distinct time periods:

September 2010. A stream of $77,932,679$ ad impressions were taken from ad traffic on a single day in September, 2010, out of which $448,282$ distinct web pages were sampled (uniformly at random). All impressions from this sample set of web pages are considered.

January 2011. A stream of $189,096,009$ ad impressions were taken from a single day in January 2011. From this stream, $466,858$ unique web pages were sampled and the occurrences of these pages in the impression stream were recorded.

Budget-sensitive stream-based experiments require a setting more complex than traditional cross-validation experiments. Preliminary experiments suggest that the qualitative results are not sensitive to the particular choices made, but we have not yet conducted a large-scale sensitivity analysis. The budget is allocated in epochs of $25,000$ observations; each active inference technique is given a budget for selecting twenty-five examples for labeling per epoch. We assume the query cost $q(x_i)$ to be constant and small enough that the budget can be exhausted for all techniques (to avoid that additional complexity). For each technique, if an instance is encountered subsequently in the stream, it is classified directly using the already-purchased ground-truth label.[9] For examples without ground-truth labels, the model's predicted class distribution is used via the cost-sensitive calculation described in Section 2. In all strategies, if for any reason (e.g., poor utility-distribution estimation) the budget is not expended during a particular epoch, it is added to the budget of the next epoch. This is done to eliminate any disadvantage a particular strategy may have due to unused budget, isolating the quality of the selections themselves in the evaluation. However, all optimizations are performed on a per-epoch basis—effectively ignoring the impact of such remainders when performing selections.

Class probability estimation is performed with logistic regression trained using stochastic gradient descent and feature hashing [28]. These choices were based on efficiency during training and induction, critical given the massive numbers of experimental runs performed in this work, and the ability to naturally learn sequentially, a property which we will leverage in Section 6. Smaller-scale experiments indicate that the main findings are independent of the type of model induction technique used (indeed the chosen learning technique is quite competitive). Note, given appropriately selected training data, predictive models seem to perform well on this paper's classification tasks; e.g., for adult content AUC is around 0.97 (and it can be much higher with significantly larger training sets). The devil's detail is that the tremendous class skew still can leave precision wanting.

We compare the following active inference strategies.

Random. Randomly select k examples for labeling during each epoch. If the number of observations to be seen in each budget epoch can be approximated in advance, this strategy simply selects k instances uniformly.

First. Select the first not-yet-labeled examples encountered during each epoch until the budget is depleted. Under certain distributions, First may be a very strong competitor: it maximizes the opportunity for each acquired label to be utilized for direct inference. In addition, the examples appearing first may be those examples with the greatest frequency (or not, depending on the skew of the distribution $p(x)$).

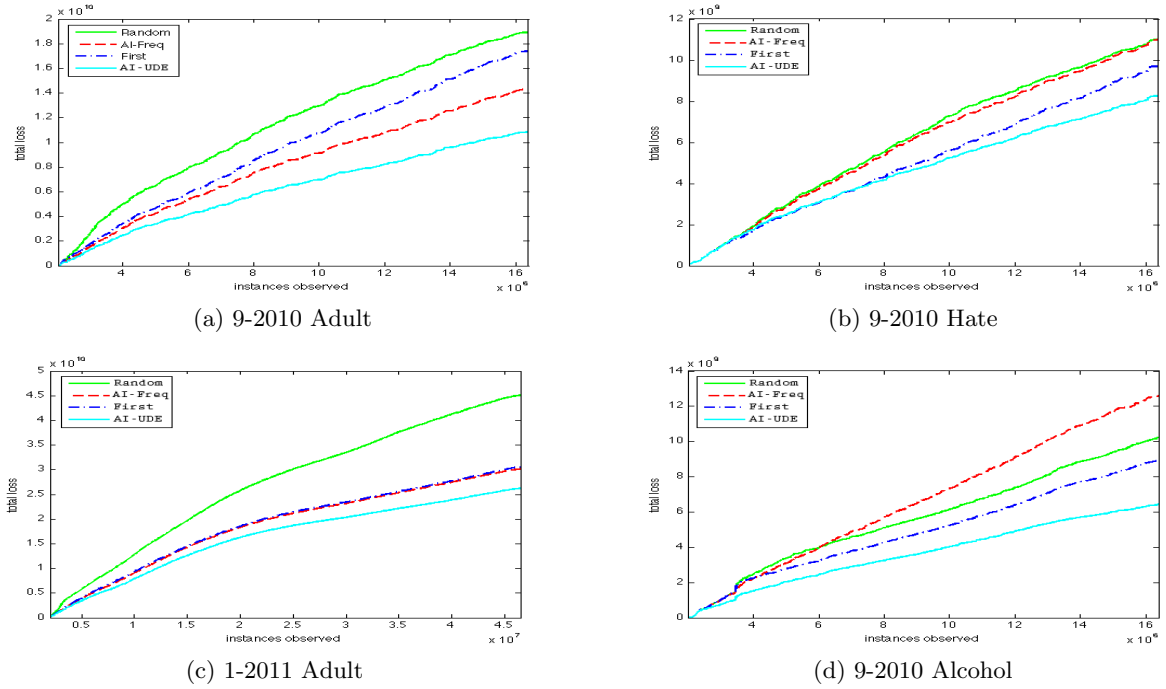
Active Inference with Utility Distribution Estimation (AI-UDE). Here we use the full expected utility-distribution estimation based on our proposed framework: $\mathcal{U}(x_i) = \phi(x_i)L(x_i, \hat{y}_i) - q(x_i)$, including cost-sensitive expected utility, online density estimation, and the online utility-distribution estimation proposed in Equations 2 and 6. Unless noted otherwise, for all AI techniques density estimation is performed using DMN with $\alpha = 1$; we return to SGT later.

AI-Frequency (AI-Freq). Here we turn off the loss por-

[7] In this case, a long tailed distribution such as the one presented in Figure 1.

[8] Rather than storing the examples themselves, unique identifiers (in the case of the ad safety problem, a URL) are used to maintain frequency information. The same identifiers are used to look up stored labels from the active inference process, giving the proposed process a small memory footprint.

[9] Additionally, these labels could optionally be used to supplement the training data available to the model, potentially reducing the future error rate. We explore this scenario in Section 6.



(a) 9-2010 Adult

(b) 9-2010 Hate

(c) 1-2011 Adult

(d) 9-2010 Alcohol

Figure 2: Active inference comparison on four datasets, representing the qualitative behavior on all the stream datasets.

tion of the UDE calculation, setting $\mathcal{U}(x_i) = \phi(x_i)$. While labeling the most frequent examples is simple in a pool-based multiset, such selection is non-trivial in the stream setting. For example, even if I get enough data to estimate $\phi(x_i)$ well for a particular x_i, how do I know whether this is going to be one of the *most frequent* x's? Fortunately, our online utility-distribution estimation framework given by Equation 6 solves this problem as well.

AI-Loss Only (AI-Loss). Here the frequency $\phi(x)$ is ignored for the purpose of utility-distribution estimation, using only $\mathcal{U}(x_i) = L(x_i, \hat{y}_i) - q(x_i)$. This represents the extension of prior strategies that do not use frequency estimation, one of the main novel contributions of the AI-UDE approach. This can be considered an application of cost-sensitive classification with a reject option to the setting where the example space is only partially observed, utilizing the online UDE framework of Equation 6.

5. RESULTS

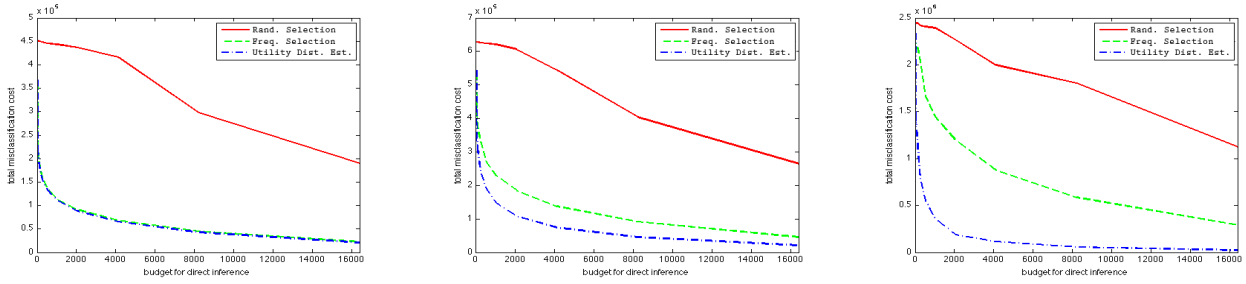
Figure 2 presents experimental results on four domains; these represent the qualitative behavior over all eight. With few exceptions (e.g., Figure 2(c)), the active inference strategies utilizing online utility-distribution estimation outperform those that do not. This suggests that there is benefit to estimating the distribution on the utility space for budgetary planning.

The full-blown utility-distribution estimation strategy (AI-UDE) dominates in all eight cases, offering the most promising results among all the strategies. In some cases AI-UDE yields a reduction in misclassification cost of 20% to 30% in comparison to the next-best strategy. Thus even the suboptimal (and likely inaccurate) online density estimators are sufficient to provide useful selections for these problems.

Considering only $\phi(x)$ in utility estimates (AI-Freq) still provides very promising loss results. Even without the rest of the expected utility calculation, being able to estimate $\phi(x)$ online—and importantly reason about its overall distribution online (Equation 6)—can provide for a competitive active inference strategy. This seems to be due to the long-tailed distributions, where some instances are much more frequent than others, and on these even very small errors multiply quickly. However, AI-Freq is not as consistent as AI-UDE; sometimes selection based solely on the frequency and frequency-distribution estimations is less effective than competing strategies. This may be due to high skew observed in the conditional distribution $p(y|x)$ in addition to that observed in $p(x)$. Such a skew may result in AI-Freq selecting only negative examples ($c = 0$), the likely "default" classification, which would therefore result in labeling many examples the model would already classify correctly.

We left the results from AI-Loss out of the figures. This strategy, meant to represent an online extension of prior methods for prediction-time label acquisition, was not competitive with the other selection heuristics considered and severely reduced the interpretability of the graphs. The poor performance seems to be due to the strictly loss-based approach having a tendency to choose outliers, singletons in terms of $\phi(x)$. (These often seem to be the cases for which the model is most uncertain, and therefore for which the frequency-ignorant expected utility of labeling them is highest.) In the stream, labeling singleton instances offers minimal reduction in misclassification cost. This suggests caution should be taken in direct applications of reject inference to data streams with repeated instances.

To provide some sensitivity analysis, we varied the budget and misclassification cost in a narrower experimental setting. Specifically, we selected a subset of 35,000 pages extracted from the real stream, with an 80 to 1 class skew for adult content. Over ten folds of cross-validation, a power law distribution ($\alpha = 2$) was induced on the testing portion of each fold in order to simulate the actual skewed $p(x)$. Then for the experiments that follow we assume that $\phi(x)$ is estimated perfectly. Thus we can focus on how much value the frequency and the loss components add, without being



(a) 10 to 1 (b) 100 to 1 (c) 1,000 to 1

Figure 3: Total incurred costs for different instance selection strategies for different given budgets, B

confounded by errors in the online density estimation (more on that later).[10] Here we assume that the per-query labeling cost $q(x_i) = 1$ for all x_i.

Figure 3 compares the total loss of Random, AI-Freq, and AI-UDE while varying the total available budget (along the horizontal axes) and varying the false-positive error cost (across the three panes) from 10 to 100 to 1,000 times the false-negative error cost.

From these experiments we see two things. First, no matter what the budget, active inference provides a substantial reduction in total loss. Second, given this power-law distribution, it is only for the larger cost ratios that the utility component adds value; for the 10 : 1 cost ratio AI-Freq and AI-UDE are essentially equivalent. It should be clear that this is just a demonstration: for a uniform distribution, the utility component obviously would dominate for any non-trivial cost ratio (assuming the probability estimation is good enough). On the power-law distribution, when the costs are high enough, both components make substantial contributions, and the full-blown AI-UDE gives a much larger loss reduction than AI-Freq.

6. ACTIVE INFERENCE AND LEARNING

What we have presented so far is only half the story. While many stream-based active inference settings may begin with a satisfactory model in order to perform statistical predictions,[11] in other applications we need to learn the model simultaneously with making predictions. In addition, as long as we are acquiring labels, we may want to improve the model. Countless prior papers on active learning (AL) have demonstrated the benefit of carefully choosing instances to label for training. In some applications there may be separate budgets for active inference and for labeling for training. However, even then the active inference selections will produce more potential training data. Therefore, we should consider how to allocate a single budget so as to get the best overall performance, taking into account both (active) inference and (active) learning. A full development being well beyond the scope of this paper, we present some interesting insights into the interplay of these two label demands, and demonstrate the striking generality of the online utility-distribution estimation strategy presented in Equation 6.

Similar to our setting, online active learning is concerned with selecting instances for labeling from a stream, where the current model is applied to the subsequent stream. Helmbold and Panizza [19] first looked at the tradeoffs between the cost of errors and the costs of labels in online active learning. Subsequently there have been several proposed techniques for "label efficient" techniques including b-sampling [5], and a logistic confidence model [24]. Online active learning research does not address the use of labels for direct inference, nor is repetition in the instance stream accounted for explicitly.[12] There also has been much work in "pool-based" active learning [25]. Many (not all) AL techniques formulate a usefulness score for instances, and then select or sample instances based on this score. The most commonly used family of AL techniques measures usefulness in terms of model uncertainty, with the score ranging from the similarity of the estimated probabilities of the two most likely classes, to the entropy or variance of an ensemble of constituent models, to the distance from a separating hyperplane. Roughly: get more labels on instances near the decision boundary. (Note the similarity to reject inference described above.)

Initially it appears that for our setting only online active learning approaches are appropriate, because the pool-based approaches presume that you score all the examples in the pool first in order to select among them. However, the utility term, $\mathcal{U}(x)$ from Equation 6 could instead denote any AL usefulness score! Then the online utility-distribution estimation procedure we introduced and applied for active inference above will instead estimate the distribution of active learning usefulness scores across the stream. Thus, it can be used to apply arbitrary score-based active learning techniques to the online/stream setting.

In fact, consider AI-UDE presented above. The full-blown AI-UDE can be thought of as generalized uncertainty sampling (cf. [23]) for streams with different error costs, repetition, unknown $p(x)$, and the need for budget management.

Proposition: *The active inference strategy of selecting the instance x_i with largest $\hat{\psi}(x_i)$ selects the same instance as uncertainty sampling under conditions of uniform (estimated) instance frequency, uniform query cost, and uniform error cost.*

Proof: The proof proceeds simply by unwinding the derivation above in Section 2 (See [1] for details). □

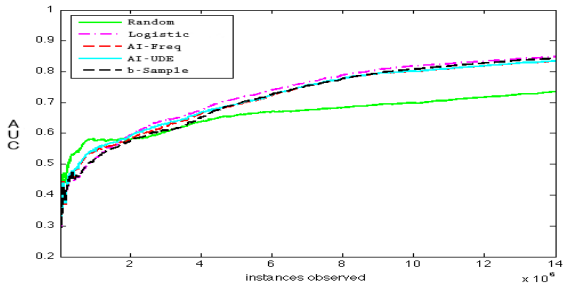
The generalization to different error costs generalizes uncertainty sampling in the same way reject inference was generalized (see Section 2). The other generalizations make intuitive sense in our setting as well: prefer to spend the labeling budget on instances, ceteris paribus, if they would be more costly to get wrong, if labeling them is particularly

[10] Note that by assuming we know $\phi(x)$ we've factored out the key components of the stream; this now is pool-based active inference where there may be replication in the pool.

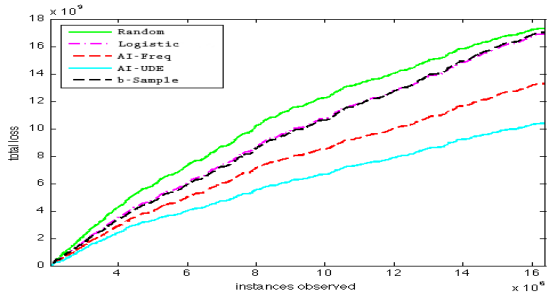
[11] For instance, in safe advertising an objectionability model may already exist when encountering new advertising partner, with a very different impression stream.

[12] This only scratches the surface of online active learning. For example, recently, Beygelzimer et al. [4] made a significant theoretical and empirical advance in importance-weighted active learning.

cheap, and/or if they are particularly likely to "reappear." To our knowledge no prior work has weighted uncertainty sampling by $p(x)$ or, equivalently, $\phi(x)$ (cf., the discussion above on density-sensitive AL).



(a) AUC of Training Only



(b) Loss Experienced by Learning and Induction

Figure 4: A comparison of active learning strategies with our proposed active inference strategies

Figure 4(a) compares the ROC area (AUC) of *learning* with the online AI strategy AI-UDE and the online AL strategies b-sampling and logistic sampling (discussed above). This experiment is performed on the Sept. 2010 adult classification data set. As before using $200,000$ unlabeled examples to burn-in a DMN density estimator, a completely untrained model is deployed and trained only on those instances selected via each acquisition strategy.[13] In line with our conjecture, the AI-UDE strategy indeed chooses examples that accelerate learning over random selection, as one would expect with an effective active learning technique. Surprisingly, the dedicated online active learning techniques provide only marginal additional improvement (at best). This provides suggestive evidence in support of our conjecture that AI-UDE is a solid candidate for active *learning* in stream settings.

Finally, let's put everything together. If AI-UDE did so well for active learning, perhaps the online AL strategies will do very well for active inference. Figure 4(b) presents the same label acquisition strategies applied to the same problem—starting as in the AUC results with the untrained model. However, now we see the effect on *total misclassification loss* if each strategy is used for AI and AL (otherwise, the same setup as the main results presented earlier). Unlike with the AUC results, here the AI and AL strategies do not perform comparably. For simultaneous AI+AL, AI-UDE dominates. This is clearly due to the focus on active inference, since we saw that the underlying models are comparable (at least in terms of AUC). These results are suggestive that active inference techniques like AI-UDE should be considered more broadly for reducing overall loss in stream applications needing online active learning.

[13]The overall performance figures are fairly low for this application—the best methods achieving an AUC of approximately 0.8 after an accumulation of $14,000$ labeled examples from over 14 million observations. This domain, with such extreme class skew, needs much larger training sets or special example selection regimes to get the best possible accuracies; typical active learning strategies have trouble finding positive examples [3, 2].

7. CONCLUSIONS & LIMITATIONS

This paper addresses the problem of prediction-time label acquisition, presenting a framework that generalizes both classification with a reject option and existing work in active inference. Within this framework, we presented the problem of online active inference, a real problem faced in applications such as our running example of web-page classification for safe advertising. The results on stream data from several safe-advertising problems demonstrates that online active inference has the potential to reduce the cost of inference substantially under skewed error costs and example frequency distributions. A key to success is the online estimation of two different quantities: (i) the frequency distribution and (ii) the distribution of utilities that will be encountered.

The expected-loss-minimizing active inference strategy we introduced turns out to be a generalization of traditional uncertainty sampling, as discussed in Section 6, and is quite effective when one also must (actively) *learn* the predictive model simultaneously with using it for inference. Furthermore, our framework for online estimation of the utility distribution provides the tooling to extend many batch-mode active learning strategies into the online setting.

While this work presents several advances in the development of prediction-time label acquisition techniques, there are a number of limitations that provide a fallow field for future research. Based on what we have seen so far, the largest potential for improvement in the active inference results may be from improved online density estimation. We used DMN extensively in the experimental portion of this work due to its speed of estimation for the very large-scale runs conducted for the main results. However, smaller-scale experiments show SGT yielding superior density estimation at the cost of speed. We see this in Figure 5, which shows Random, AI-UDE with DMN, AI-UDE with SGT, and AI-UDE with perfect ("oracle") estimation of $\phi(x)$. The current density estimation seems to get about half the total possible improvement in performance, and we see that SGT improves over DMN, but not substantially. Further improvements may be realized via techniques more sophisticated than SGT, or possibly by conditional density estimation (i.e., taking x into account). Note that the oracle estimation case probably is an unachievable "floor" on performance, since we have to observe the stream of data to do the online estimation.

Relatedly, there is an exploration/exploitation tradeoff in utility-distribution estimation. Because we are estimating $p(x_i)$ on the fly, it may be the case that for a particular x_i we first use the model to classify it for a while, and then eventually acquire its label. When are we certain enough to label, rather than wait "one more" time? The current models make this decision implicitly as they learn $p(x)$, but

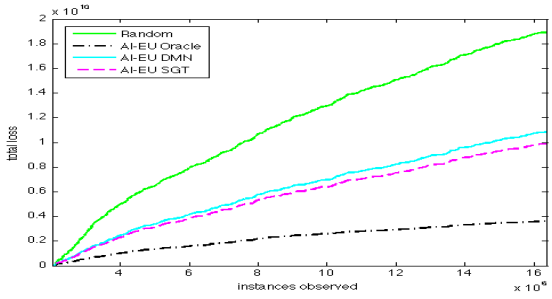


Figure 5: Comparing the AI-UDE strategy with different density estimation techniques, including perfect estimation

perhaps the decision could be improved with a confidence-interval strategy like that of [13]. More generally, the budget needs to be managed over the stream, trading off several competing desires. Labeling pages early both maximizes the value of those particular labels and maximizes the value to model induction. Labeling later allows better estimation of $p(x)$, and therefore may increase the value of the active inference.

Relatedly, this work assumes labels are made available instantly upon request. Incorporating a delay between requests and data acquisition is an important direction for future work, and a matter of practical importance. For example, it may be that waiting t time units only ends up shifting the curve up, because t is short enough that the expected additional loss will essentially be constant. Furthermore, this is just one possible budget framework.

We assume that $p(x)$ is static. Many realistic settings have a dynamic $p(x)$: new instances appear not only because they simply are rare, but because they actually are new. The dynamics of this change can be abrupt, with instances rising rapidly in popularity: e.g., new popular web pages or new popular search terms (e.g., "Egypt riots").

We have made the usual assumption that labeling is error-free. However, in reality for the applications we are considering the labeling will be done by error-prone humans, for example via a micro-outsourcing system, and the active inference procedure should take that into account. For example repeated labeling [27] becomes a strategy that must be considered. This adds wonderful complexity to active inference. There no longer is a clear switch from model-based inference to human-based inference. Now we need to consider the fusion of different evidence, acquired at different costs, at different times. The model's estimation could be seen as just another labeling source; for certain examples it may even be more accurate than an average human labeler.

Despite these limitations, we hope that the techniques and results presented in this paper have made significant progress toward the development of systems that manage labeling budgets as well as possible in real-world, stream-based, online prediction systems.

8. REFERENCES

[1] J. Attenberg and F. Provost. Active inference and learning for classifying streams. In *BL-ICML '10: Workshop on Budgeted Learning*, 2010.

[2] J. Attenberg and F. Provost. Inactive Learning? Difficulties Employing Active Learning in Practice. *SIGKDD Explorations*, 12(2):36–41, 2010.

[3] J. Attenberg and F. Provost. Why label when you can search? Strategies for applying human resources to build classification models under extreme class imbalance. In *KDD*, 2010.

[4] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proc of the 26th Intl Conf on Machine Learning*, ICML '09, 2009.

[5] N. C. Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. In *J. Mach. Learn. Res.*, volume 7, pages 1205–1230, 2006.

[6] M. Bilgic and L. Getoor. Effective label acquisition for collective classification. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 43–51. ACM, 2008.

[7] M. Bilgic and L. Getoor. Reflect and correct: A misclassification prediction approach to active inference. *ACM Trans. Knowl. Discov. Data*, 3, December 2009.

[8] M. Bilgic and L. Getoor. Active Inference for Collective Classification. In *AAAI*, 2010.

[9] C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, January 1970.

[10] C. K. Chow. An Optimum Character Recognition System Using Decision Functions. *IEEE Transactions on Electronic Computers*, (4):247–254, December 1957.

[11] K. Church and W. Gale. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech & Language*, 5(1):19–54, January 1991.

[12] B. D. Davison. Topical locality in the web. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 272–279, New York, NY, USA, 2000. ACM.

[13] P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 259–268, New York, NY, USA, 2009. ACM.

[14] C. Elkan. The Foundations of Cost-Sensitive Learning. In *IJCAI*, pages 973–978, 2001.

[15] T. Fawcett and F. Provost. Activity monitoring: Noticing interesting changes in behavior. In *In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, 1999.

[16] W. A. Gale. Good-Turing smoothing without tears. *Journal of Quantitative Linguistics*, 2, 1995.

[17] D. Golovin, M. Faulkner, and A. Krause. Online distributed sensor selection. In *Information Processing in Sensor Networks*, pages 220–231, 2010.

[18] I. J. Good and G. H. Toulmin. The number of new species, and the increase in population coverage, when a sample is increased. *Biometrika*, 43(1-2):45–63, June 1956.

[19] D. Helmbold and S. Panizza. Some label efficient learning results. In *COLT '97: Proceedings of the tenth annual conference on Computational learning theory*. ACM, 1997.

[20] R. Herbei and M. H. Wegkamp. Classification with reject option. *Can J Statistics*, 34(4):709–721, 2006.

[21] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *ICML*, 2004.

[22] M. J. Rattigan, M. Maier, and D. Jensen. Exploiting Network Structure for Active Inference in Collective Classification. Technical Report 07-22, University of Massachusetts Amherst, 2007.

[23] M. Saar-Tsechansky and F. Provost. Decision-Centric active learning of Binary-Outcome models. *INFORMATION SYSTEMS RESEARCH*, 18(1):4–22, Mar. 2007.

[24] D. Sculley. Online active learning methods for fast label-efficient spam filtering. In *Fourth Conf. on Email and AntiSpam*, 2007.

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

[26] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1070–1079, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[27] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD '08*, 2008.

[28] K. Weinberger, A. Dasgupta, J. Attenberg, J. Langford, and A. Smola. Feature hashing for large scale multitask learning. In *ICML '09*, 2009.