Practical Learning from One-Sided Feedback

D. Sculley
Department of Computer Science, Tufts University
161 College Ave., Medford, MA USA
dsculley@cs.tufts.edu

ABSTRACT

In many data mining applications, online labeling feedback is only available for examples which were predicted to belong to the positive class. Such applications include spam filtering in the case where users never check emails marked "spam", document retrieval where users cannot give relevance feedback on unretrieved documents, and online advertising where user behavior cannot be observed for unshown advertisements. One-sided feedback can cripple the performance of classical mistake-driven online learners such as Perceptron. Previous work under the Apple Tasting framework showed how to transform standard online learners into successful learners from one sided feedback. However, we find in practice that this transformation may request more labels than necessary to achieve strong performance. In this paper, we employ two active learning methods which reduce the number of labels requested in practice. One method is the use of Label Efficient active learning. The other method, somewhat surprisingly, is the use of margin-based learners without modification, which we show combines implicit active learning and a greedy strategy to managing the exploration exploitation tradeoff. Experimental results show that these methods can be significantly more effective in practice than those using the Apple Tasting transformation, even on minority class problems.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: General; H.2.8 [**Information Systems**]: Data Mining

General Terms

Algorithms, Experimentation

Keywords

online learning, streaming data, active learning, apple tasting, data mining

KDD'07, August 12–15, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-609-7/07/0008 ...$5.00.

1. INTRODUCTION

The problem of learning from one-sided feedback was introduced by Helmbold, Littlestone, and Long [9], who described this as the the Apple Tasting problem: the problem of learning to identify sweet apples from visual cues. Of course, an apple taster gets no feedback from those apples it rejects, only from those that it actually chooses to taste. This is a variant of the standard online learning framework. In one-sided feedback, the learner only receives feedback when it predicts a positive label for the given example. That is, the only way a learner can see the true label of an example is to predict that it is a member of the positive class.

Data mining streams with one-sided feedback is an important problem with a range of practical applications. For example, an email filter may send messages to either a *spam* folder for bad email or a *ham* folder for good email. If a lazy user never checks the spam folder, as is often the case, then the learner will only receive feedback on messages that were predicted to be ham. There are a wide variety of other applications with one-sided feedback. In a document retrieval setting, users cannot give relevance feedback for documents not shown [17]. In an online advertisement setting, user reaction can only be learned for advertisements actually displayed [1]. Oil prospectors only learn if there is, indeed, oil in a predicted location after they finish drilling, and learn nothing about locations not drilled.

The problem of learning from one-sided feedback defeats several classical online learning algorithms, such as Perceptron [15] and Winnow [13]. These mistake-driven algorithms suffer in this scenario, especially in the presence of noise, as the online updates tend to sacrifice recall for precision and may recognize very few positives. Helmbold, Littlestone and Long showed how to convert any standard online learner, including these mistake-driven methods, into an apple tasting algorithm by randomly sampling from those examples predicted to be in the negative class [9] with resultant mistake bounds. However, this method samples uniformly from the predicted negatives, and thus does not necessarily request labels for the most informative examples.

Contributions.

We propose that practical data mining on one-sided feedback streams is best done with active learning methods. We show that Label Efficient active learners perform well from one-sided feedback, requesting fewer labels than the Apple Tasting methods. We also show, somewhat surprisingly, that margin-based learners such as Online Support Vector Machines (SVMs) and Perceptron with Margins both learn

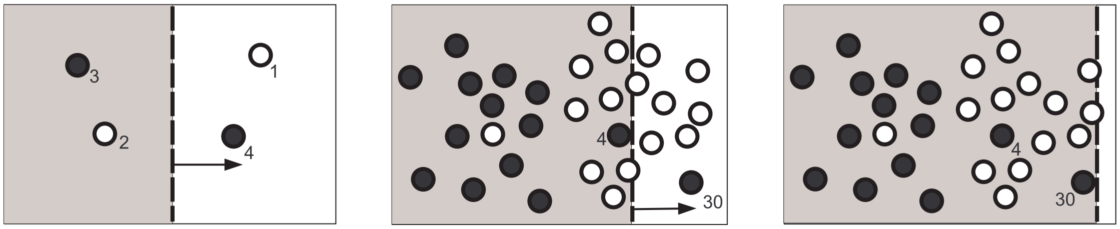


Figure 1: One Sided Feedback Breaks Perceptron. Here, white dots are positive examples, the black dots are negatives, the dashed line is the prediction hyperplane, and the shaded area predicts negative. Examples 1, 2, and 3 each cause no updates: 1 and 3 are correct, and no feedback is given on 2. Examples 4 and 30 are the only examples causing updates, *ratcheting* the hyperplane until no positives are correctly identified.

effectively from one-sided feedback *without modification.* In the one-sided feedback scenario, it turns out that margin-based learners implicitly use an active learning strategy and a greedy search solution to the exploration/exploitation tradeoff. Our experiments show that both types of active methods can achieve high levels of performance with many fewer labels than the Apple Tasting solution, and that the margin-based methods are often the most effective.

The remainder of this paper proceeds as follows. Section 2 gives preliminary background on one-sided feedback and reviews the Apple Tasting transformation with an eye towards possible improvements for practical use. In Section 3, we discuss the application of Label Efficient active learners to one-sided feedback problems. In Section 4, we show that in many cases margin-based methods can learn effectively from one-sided feedback without transformation due to implicit uncertainty sampling and a greedy approach to the exploration/exploitation tradeoff. Section 5 covers difficulties posed for learning from one-sided feedback in minority-class distributions. Experimental results are in Section 6, and the final section contains our conclusions.

2. PRELIMINARIES AND BACKGROUND

We are concerned with the problem of online learning from one-sided feedback, first described as the Apple Tasting problem [9]. We assume a distribution D on a space of examples $X = R^d$, and each example $\mathbf{x}_i$ has an associated label y_i. There is a learner L with a hypothesis function $h(\cdot) : R^d \mapsto \{-1, 1\}$ predicting the label of a given example. The learner is allowed to update its hypothesis when it is shown an example and label pair $(\mathbf{x}_i, y_i)$. There is an oracle T that returns a (possibly noisy) label y_i for a given $\mathbf{x}_i$.

Learning proceeds in a (potentially unbounded) number of rounds, $\{t_1, ..., t_{max}\}$. Given D, L, T, for each round t_i:

- An example $\mathbf{x}_i$ is drawn from D.
- L guesses a label $h(\mathbf{x}_i)$ for $\mathbf{x}_i$.
- If $h(\mathbf{x}_i) = 1$ then oracle T returns a (possibly noisy) label y'_i and L may update its hypothesis using $(\mathbf{x}_i, y'_i)$.
- However, if $h(\mathbf{x}_i) = -1$, then y'_i is never revealed to L.

In this paper, we assume that the cost of requesting a label for a truly negative example is equivalent to the cost of misclassifying a negative example, while the cost of requesting a label for a truly positive example is zero. This is equivalent to saying that the only way to request a label for a given example is to predict that it is a positive.

2.1 Breaking Classical Learners

To illustrate the issues surrounding the one-sided feedback problem, we first show that noisy one-sided feedback can break classical mistake-driven online learners such as Perceptron [15] and Winnow [13]. These learners update their hypotheses only on mistakes. However, under the one-sided feedback scenario, they are never told about mistakes made when they predict a negative label, so no updates can occur. The only mistakes they will update on are those for which they predicted a positive label for a negative example.

Updating on one-sided errors creates a *ratcheting* effect shown in Figure 1. Once the hyperplane has been shifted towards the positive side, it can never be shifted back. If the noise rate $p > 0$, then the hypothesis will converge to one which predicts every a negative label for every example. (Proof omitted.) Even if there is no noise, if the data are not linearly separable in the feature space then the hypothesis will converge to one which never mistakes a negative for a positive. (Proof omitted.) This can cause recall levels for the positive class to suffer greatly, as we show in our experiments. Even if the data are separable, a poor initial hypothesis may never be improved. Thus, purely mistake driven learners are unsuitable for one-sided feedback.

2.2 An Apple Tasting Solution

Helmbold *et al.* proposed a solution to the one-sided feedback problem and analyzed it theoretically using the mistake bound model from learning theory [9]. They showed that if a learner can be forced to make a maximum of either M_p mistakes on positive examples or M_n mistakes on negatives from full feedback from a given (noiseless) distribution, then it can be transformed into a learner making at most $M_p - M_n + 2\sqrt{TM_n}$ mistakes from one-sided feedback on that distribution. These mistake conditions can be met for Perceptron or Winnow by setting an initial bias.

Their solution (the “Apple Tasting method”, hereafter), relies on occasional random sampling from those examples which are predicted to have negative labels. When an example is sampled, a label request is made to the oracle by flipping the predicted label from -1 to 1. A label request is made on step i when $h(\mathbf{x}_i) = -1$ with probability $p = \sqrt{(1 + m_n)/i}$, where m_n is the number of mistakes found so far among the examples for which labels have been specifically requested. [9] Intuitively, this method samples the learner’s error rate to determine how much exploration is needed. As $\frac{1+m_n}{i}$ grows, more labels are requested because the observed error rate is high. When this estimate of the error rate decreases, fewer labels are requested.

```
Given: $\beta > 0$, data set $X = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$:

Initialize: $\mathbf{w} := 0$, $K = 0$
For each $\mathbf{x}_i \in X$ do:
        Compute $f(\mathbf{x}_i) =< \mathbf{w}, \mathbf{x}_i >$
        Classify $\mathbf{x}_i$ using $y'_i = sign(f(\mathbf{x}_i))$
        Draw Bernoulli random variable $Z_i \in 0, 1$
            with parameter $\frac{b_i}{b_i + |f(\mathbf{x}_i)|}$
            where $b_i = \beta\sqrt{1 + K}$
        If $Z_i = 1$ then
            Request label $y_i$
            If $y'_i \neq y_i$
                Update $\mathbf{w}$
                K = K + 1
```

Figure 2: Pseudo-code for Label Efficient active learner.

2.3 Improving on Apple Tasting

While this Apple Tasting transformation offers a robust solution to the problem of learning from one-sided feedback, we note that there are areas of possible improvement for practical use, using active learning and examining different exploration/exploitation strategies.

Active Learning.

The Apple Tasting method samples from the negative predictions in a uniform manner, without taking into account the certainty of the prediction. Although uniform sampling enables theoretical guarantees of correctness for purely separable data [9], it is not always the most efficient way to learn a good hypothesis in practice. Active learning methods attempt to choose *informative* examples to learn from. Uncertainty sampling is one such method, in which examples are chosen based on how uncertain the current hypothesis is about their label [12]. Other active learning methods include Query by Committee, in which disagreement among possible learners is cause for sampling [7]; choosing examples based on how much they would reduce the current version space [3]; and estimating how much the example would reduce training error if its label were known [16].

We propose that active learning can improve on the Apple Tasting bounds in practice on one-sided feedback problems. The methods we explore in this paper are based on uncertainty sampling, which is computationally efficient. Label Efficient learners use uncertainty sampling methods to adjust the probability that a label will be requested for a predicted negative. We also show that margin-based learners implicitly use a fixed form of uncertainty sampling to request labels. Exploring other active learning methods in one-sided feedback problems remains for future work.

Exploration/Exploitation Tradeoff.

The Apple Tasting solution strikes a particular balance between *exploration* and *exploitation*, to use terminology from reinforcement learning [21], by requesting more labels when the estimated error rate is high. Exploration of the data space allows the learner to acquire new knowledge and better estimate the optimal hypothesis – however, this exploration may incur cost associated with label requests. Exploiting previous knowledge carries no exploration cost, but may incur misclassification cost if the hypothesis is faulty. Determining the optimal balance between exploration and exploitation *a priori* for an arbitrary task is an open problem, and different approaches may be better for different situations. The Label Efficient learner uses a strategy similar to that of Apple Tasting, by attempting to explore more when observed error rate is high. Margin-based learners implicitly use a greedy exploration strategy that we describe in Section 4, that can request many fewer labels in practice but offers no theoretical guarantees.

Categorizing the Learners.

The methods we explore in this paper can be organized as follows. Classical Perceptron, with no active learning and no exploration, fails on one-sided learning. Apple tasting adds exploration to solve this problem, but without active learning may request more labels than necessary. Label Efficient methods request fewer labels and maintain theoretical guarantees. Margin-based methods use a greedy exploration to further reduce the number of needed labels in many cases, but at the sacrifice of theoretical guarantees. The following sections examine these last two learners in more detail.

Learner	Active	Exploration
Perceptron	no	none
Apple Tasting	no	error-rate driven
Label Efficient	yes	error-rate driven
Margin-Based	(implicit) yes	greedy

3. LABEL EFFICIENT ONLINE LEARNING

An online active learner must decide whether or not to request a label for a given example at that particular time step, without knowledge of the future or the ability to reconsider at a future point. When labels are costly, this creates resource allocation issues. The Label Efficient problem is to learn well with few label requests. Although this problem was posed in the standard online learning setting [8], it has natural application with one-sided feedback where sampling from the negative predictions carries cost, and sampling from positive predictions is essentially free. To our knowledge, this is the first use of label efficient learners on one-sided feedback problems.

Cesa-Bianchi *et al.* proposed a label efficient active learner based on the perceptron algorithm (see Figure 2 for pseudo-code, simplified for the case of normalized example vectors) and give bounds on the expected number of mistakes and the expected number of label requests for linearly separable data [2]. The method adapts to the number of known mistakes seen so far, and samples more frequently when higher error rates are observed. Furthermore, unlike other label efficient learners that have been proposed, this method does not require the user to specify a maximum number of examples to label, and instead manages the exploration/exploitation problem adaptively, given an initial setting of parameter β. Finally, this active method takes uncertainty into account and is more likely to sample points that lie close to the classification hyperplane.

This method can be applied in the case of one-sided feedback. Here, requesting a label for a given example forces the learner to predict a positive label for a given example. Label requests are made on all positive examples. In terms of the pseudo-code, $Z_i = 1$ whenever $y'_i = 1$. The method

```
Given: $C$, DATA SET $X = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$:

Initialize: $\mathbf{w} := 0$, $b := 0$, SEENDATA := { }
For each $\mathbf{x}_i \in X$ do:
        COMPUTE $f(\mathbf{x}_i) =< \mathbf{w}, \mathbf{x}_i > +b$
        CLASSIFY $\mathbf{x}_i$ USING $sign(f(\mathbf{x}_i))$
        ADD $\mathbf{x}_i$ TO SEENDATA
        If $y_i f(\mathbf{x}_i) < 1$
            FIND NEW OPTIMAL $\mathbf{w}', b'$ OVER SEENDATA,
               USING $\mathbf{w}, b$ AS SEED HYPOTHESIS.
            $\mathbf{w} := \mathbf{w}'$, $b = b'$
```

Figure 3: Pseudo code for Online SVM.

was originally analyzed in terms of the classical Perceptron algorithm; we apply it to other linear classifiers as well.

4. MARGIN-BASED LEARNERS

One claim of this paper is that margin-based learners can be effective learners from one-sided feedback. In this section, we review the essentials of two margin-based learners, Online SVMs and Perceptron with Margins. We then demonstrate how margin updates enable learning from one-sided feedback, revealing implicit uncertainty sampling and a greedy exploration/exploitation tradeoff strategy. This section concludes with an examination of conditions that can cause this greedy strategy to fail.

4.1 Two Learners

Here, we review two margin-based learners: Online SVMs and Perceptron with Margins. Both of these methods are linear classifiers that update their linear hypothesis not only on mistakes, but also on correctly classified examples that lie close to the classification hyperplane, enabling learning from one-sided feedback.

Online SVMs.

SVMs offer a statisticly robust method to classification. (See the introductory text by Schölkopf and Smola [19] for a complete discussion.) Briefly, an SVM computes a classification hyperplane which maximizes the margin between two classes by minimizing the following objective function:

$$\tau(\mathbf{w}, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

with the constraints:

$$\forall i = 1..n : y_i(< \mathbf{w}, \mathbf{x}_i > +b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

Here, w is the weight vector of the hypothesis, with bias b: minimizing the $||\mathbf{w}||^2$ creates the maximum possible margin. Each misclassified example $\mathbf{x}_i$ carries cost ξ_i. The parameter C sets the balance between the goals of minimizing misclassification cost and maximizing the margin.

Any standard SVM operating in a batch learning mode may be converted to an Online SVM using the simple wrapper pseudo-code shown in Figure 3. We implemented the Online SVM used in our experiments using Platt's SMO, an iterative solver [14].

Notice that the Online SVM stores every example, but only requires updates when it has made a mistaken prediction, or when a correct example lies within the margins – that is, when $|f(\mathbf{x}_i)| < 1$. Online SVMs do not need to update on any correctly classified example outside the margins due to the Karush-Kuhn-Tucker conditions, which show that such examples cannot become support vectors until a mistake is observed [19]. By the same conditions, any example lying within the margins causes an update and will move the classification hyperplane. This is a key point which enables Online SVMs to learn from one-sided feedback, as we discuss later in this section.

```
Given: $m$, $\eta$, DATA SET $X = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$:

Initialize: $\mathbf{w} := 0$
For each $\mathbf{x}_i \in X$ do:
        COMPUTE $f(\mathbf{x}_i) =< \mathbf{w}, \mathbf{x}_i >$
        CLASSIFY $\mathbf{x}_i$ USING $sign(f(\mathbf{x}_i))$
        If $y_i f(\mathbf{x}_i) < \frac{m}{2}$
            $\mathbf{w} := \mathbf{w} + y_i \eta \mathbf{x}_i$
```

Figure 5: Pseudo code for Perceptron with Margins.

Perceptron with Margins.

While classical Perceptron seeks only to minimize training error [15], the Perceptron with Margins attempts to create a margin m similar to that of SVMs. Unlike SVMs, however, Perceptron with Margins is not guaranteed to find the *maximum* margin. However, for linearly separable data there are lower bounds on the size of the margin [11].

Like Online SVM, the Perceptron with Margins updates its hypothesis both on mistaken predictions and on correctly predicted examples that lie within the margin of the classification hyperplane (see Figure 5 for pseudo-code [11]). Note that classical Perceptron is equivalent to Perceptron with Margins using parameter $m = 0$, as classical Perceptron only updates on mistakes. This is the critical distinction that allows Perceptron with Margins to learn from one-sided feedback, while classical Perceptron fails.

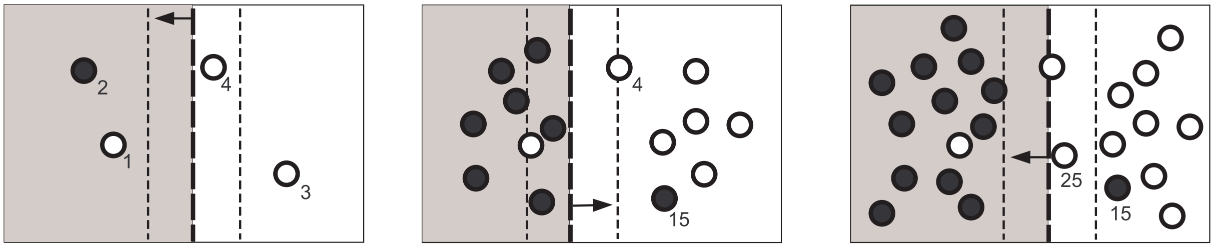
4.2 Margin-Based Pushes and Pulls

At first, it may seem counter-intuitive that any learner can learn effectively from one-sided feedback without modification. We now show the intuition driving the finding that margin-based learners can indeed learn in this scenario with no modification.

Recall that classical learners such as Perceptron are subject to *ratcheting* because they can only recognize one kind of mistake in the one-sided feedback scenario. Margin-based learners are resistant to ratcheting as they can update their classification in both directions. As before, misclassified negatives still cause updates moving the hyperplane more towards the positive, correctly classified negatives have no effect and misclassified positives have no effect as no feedback is given. Furthermore, correctly classified positives that lie outside the margins also cause no update to occur.

The key difference is: *margin based learners update their hypothesis on correctly classified examples that lie within the margin.* (See Figure 4.) The hyperplane may be pulled towards the positive by misclassified negatives and pushed towards the negative by positive examples classified within the margins. These hypothesis updates are *not* irreversible, and the hyperplane can converge to a good hypothesis.

Figure 4: Margin-Based Pushes and Pulls. Examples 1, 2, and 3 cause no updates, as before. But Examples 4 and 25, each correctly classified but within the margins, push the hyperplane towards the negative. Example 15, a misclassified negative, pulls the hyperplane towards the positive.



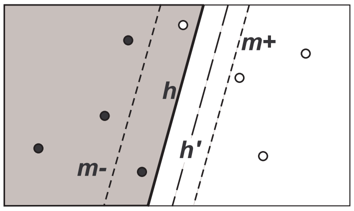


Figure 6: Implicit Uncertainty Sampling for Perceptron with Margins. The margin-based learner with hypothesis h and margins m_+ and m_- learning from one-sided feedback reduces to an active learner with hypothesis h' and margins m_+ and h using uncertainty sampling in the region between h and h'.

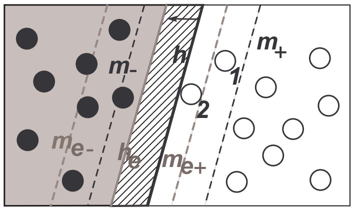


Figure 7: Exploration and Exploitation. If the initial hypothesis is h, then examples 1 and 2 cause margin updates pushing h_e out towards m_-, but not beyond it unless an example is found to lie between h and h_e.

4.3 Margins, One-Sided Feedback, and Active Learning

Here, we demonstrate the claim that margin-based methods that are applied to one-sided feedback problems implicitly use active learning to sample from negative predictions.

Reduction 1. *A margin-based learner L with margin m learning from one-sided feedback reduces to an active margin-based learner L' with margin $\frac{m}{2}$. The sampling rule for this active learner is to perform uncertainty sampling on any example $\mathbf{x}_i$ for which the prediction $f'(\mathbf{x}_i) \geq \frac{-m}{4}$.*

Assume the learner has classification hyperplane h defined by its weight vector $\mathbf{w}$, with margin planes m_+ on the positive side and m_- on the negative side, and that the distance between each margin and the hyperplane is $\frac{m}{2}$ (see Figure 6). Now consider the hyperplane h', which lies halfway between h and m_+. We can view h' as a classification hyperplane for learner L', with margins m_+ on the positive side and h on the negative side, each at a distance of $\frac{m}{4}$ from h'. Because h' is translated distance $\frac{m}{4}$ from h, L' will score each $\mathbf{x}_i$ with $f'(\mathbf{x}_i) = f(\mathbf{x}_i) + \frac{m}{4}$.

Active learner L' requests the label y_i for any example $\mathbf{x}_i$ found to lie between h and h' – that is, for any example such that $f'(\mathbf{x}_i) \geq -\frac{m}{4}$. This label y_i is always available, because $\mathbf{x}_i$ lies to the positive side of h and one-sided feedback will be provided to L. L' requests labels for its predicted negatives that lie close to its classification hyperplane: a simple form of uncertainty sampling [18]. Furthermore, L' requests labels for all examples that lie to the positive side of h', and such labels are also always available to L under the one-sided feedback scenario. Thus, L' performs uncertainty sampling on any $\mathbf{x}_i$ such that $f'(\mathbf{x}_i) \geq -\frac{m}{4}$.

As a margin-based learner, L' updates on mistakes or examples found within the margins of h', computing a new hypothesis h''. L then adopts a new hypothesis from L' as follows. If L' is a Perceptron with Margins, the new hypothesis h for L will be $h = h'' - \frac{m}{4}$. If L' is an Online SVM, the new hypothesis will simply be $h = h''$ because the SVM will optimize over all labeled examples that have been seen to that point, regardless of its starting hypothesis. Thus, L reduces to L'.

4.4 Exploring and Exploiting

One of the primary problems in online active learning is resource allocation [8], often referred to as the exploration/exploitation tradeoff [21]. It is difficult to determine *a priori* the best balance between sampling (which may incur labeling cost) and prediction without sampling (which may incur misclassification cost) for arbitrary distributions [21]. The Apple Tasting [9] and Label Efficient [2] methods both attempt to strike a balance by estimating the error rate. However, determining the error rate incurs sampling cost, and the upper bounds computed for these methods may not be tight in practice. For example, the current hypothesis may be good but many labels may need to be requested before the error rate confirms this. Margin-based methods use a greedy approach to balancing this tradeoff that can require many fewer labels in practice. However, performance for this greedy method cannot be guaranteed due to the possibility of malicious or pathological distributions.

The greedy strategy is the following. When h is a hyperplane consistent with all seen labeled data, the learner L only requests labels for examples that it predicts to be positive. This is a conservative strategy emphasizing exploitation, and incurring zero labeling cost. Note that at

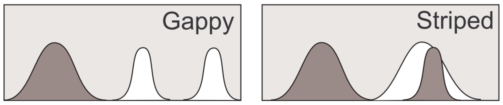


Figure 8: Pathological Distributions.

this time, examples on the negative side of m_- are strongly believed to be negatives, and those between h and m_- are suspected to be negative.

When a new example $\mathbf{x}_i$ is found to lie within the margins, between h and m_+, the learner is willing to explore, and the hyperplane is shifted through margin-updates, to create a new hyperplane h_e, with margins m_{e+} and m_{e-} (see Figure 7). Assuming a moderate learning rate (that is, $\eta \leq \frac{m}{2}$ for Perceptron with Margins or non-extreme values of C for Online SVMs), h_e will lie somewhere between h and m_-, causing the learner to sample from the suspected negatives (from the perspective of h), but never from examples strongly believed to be negatives. Each new positive example between h and the new m_{e+} will push h_e closer to m_-. However, each such update will shrink the gap between h and m_{e+}, until $h = m_{e+}$. At this point, h_e can be located no further toward negative than m_-.

Note that h is still consistent with all the seen, labeled data at this point (although h is no longer maximum margin). The only thing that will cause an update now is misclassifying a negative, or finding a positive between h and h_e. Either of these cases would show that the original h is no longer consistent with the seen data, and L must recompute a new h and start again with the conservative strategy.

Thus, unlike the Apple Tasting or Label Efficient strategies, margin-based learners do not sample from all predicted negatives with positive probability – only from those that are close to h, and only when there have been sufficiently many positives found between h and m_+ to encourage further exploration. One way to view this method is as a greedy strategy for finding support vectors with few label queries – very few negatives are sampled unnecessarily.

4.5 Pathological Distributions

Before moving on in this discussion, some caveats are in order as this greedy exploration strategy can be stalled or defeated by certain pathological distributions. Linearly separable distributions that include large gaps may cause margin-based methods to cease making progress. This will occur whenever the probability of of an example landing in the space between h and h_e is zero. For many interesting distributions, this only occurs in the margin between the two classes. However, it is possible to have such a gap within a single class (see Figure 8), which will have the same effect when the side of the gap is greater than $\frac{m}{2}$. Gappy distributions may be dealt with by increasing the margin size. Note that this provides a second intuition for the failure of classical Perceptron: when $m = 0$, every distribution is a gappy distribution.

In some cases, a distribution may not be linearly separable in the feature space, but we may still wish to find a hypothesis that minimizes loss as with the soft margin SVM. In many of these cases, the margin-based methods will be successful, as the greedy exploration will continue in the limit so long as the expected loss per example from examples lying between h and m_+ is less than some set threshold. However there is the possibility of striped distributions (see Figure 8) that can cause the greedy exploration to fail. A stripe is a region of at least width $\frac{m}{2}$ where the expected cost of applying the surrounding area's class label to examples from that region is greater than the cost of applying the opposing label. As with gaps, stripes may be dealt with by increasing the margin size in some cases, or by adjusting misclassification costs, or by finding a transformation of the feature space that renders the data linearly separable (removing the stripes).

Another possible failure of margin-based learners can be caused by malicious orderings of the examples. In a noisy domain, it is possible for an adversary to select a sequence of incorrectly labeled examples that will cause ratcheting by one-sided learners. Such malicious orderings may be possible in certain one-sided feedback applications such as optimal placement of banner advertisements.

Finally, when the learning rates are set too high, the learner may overshoot the positive class after it has misclassified a negative (or a series of negatives). If the resulting hyperplane is placed beyond the positive class, classifying everything as a negative, then the learner will be ratcheted and no further learning will take place.

5. MINORITY CLASS PROBLEMS

Learning from one-sided feedback is particularly challenging in when the positive class is a minority in the distribution. This is a challenging problem for active learning in general. It has been shown that for non-homogeneous distributions, such as minority-class distributions, a linear classifier such as Perceptron using active learning can require as many labels to achieve a given error rate as the same method without active learning [6]. That is, active learning may give no benefit in these situations. This is unfortunate, as minority class problems are common in practical data mining.

Thus, active learning solutions to the one-sided feedback problem necessarily suffer in the case of minority class distributions. Furthermore, Apple Tasting and Label Efficient methods that attempt to sample error rate may have difficulty because the measured error may be low even when the entire minority class is misclassified. Margin-based methods may have similar difficulty, where a long sequence of observed negatives may cause ratcheting in the hyperplane. These issues may be ameliorated by assigning different misclassification costs to the two classes.

In our experiments with minority class one-sided feedback, we test two standard methods of cost weighting: example weighting, and threshold biasing.

Example Weighting.

One method of assigning misclassification costs is to weight examples from the minority class more heavily. In batch settings, this may be done by oversampling from the minority class, but this may lead to inefficiency with SVMs as the computational cost of this method increases with the size of the data set. One method of example cost weighting with SVMs is to use different values of the C parameter for each class [19], allowing errors of each class to be weighted differently in the loss function. We can modify the Perceptron update rule to consider costs as well, using the update

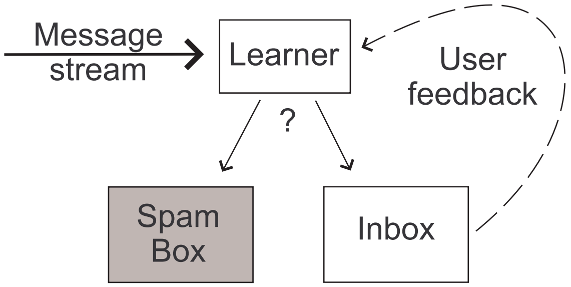


Figure 9: Spam Filtering Feedback.

$\mathbf{w} := \mathbf{w} + y_i \eta \mathbf{x}_i c_{y_i}$ on each mistake or margin update. Here, c_{y_i} is the cost of making a mistake on an example with class label y_i.

Threshold Biasing.

A second method of assigning misclassification costs with linear classifiers is to add an additional biasing constant to the classification threshold, effectively translating the hyperplane. This allows the learner to trade errors of one class for another. As has been thematic in this paper, in one-sided learning biasing the hyperplane towards the negative also has the effect of creating additional uncertainty sampling, and forces the learner to request labels from predicted negatives lying near the unbiased hyperplane. This gives additional robustness to the problem of one-sided learning from minority-class distributions.

6. EXPERIMENTS

In this section, we report experiments in one-sided feedback scenarios from two domains: spam filtering with relatively even class distributions, and document classification with minority class distributions. These results give strong support for the use of active learning for one-sided feedback.

6.1 Spam Filtering from One-Sided Feedback

Spam filtering is a natural application of online learning from one-sided feedback. In a typical email system, a learning-based filter is applied to distinguish between wanted emails, and unwanted emails known as *spam*. The messages predicted to be wanted by the user (sometimes referred to as *ham* emails) are sent to the user's inbox, while the predicted spam are sent to a spam box. The online learner relies on user feedback to update its hypothesis. However, in many situations a user may never check the emails that are sent to the spam box – no feedback is given on any message predicted to be spam (see Figure 9). This is a different scenario than is typically considered in the evaluation of spam filtering methods, which normally assume that feedback will be given on every message regardless of its classification [5]. However, the high accuracy that many machine learning methods achieve in this full-feedback scenario may criticized as being overly optimistic. While the full-feedback scenario represents the ideal user from the perspective of the learner, the one-sided feedback scenario represents the ideal learning scenario from the perspective of the user.

Experimental Setup.

We construct an online learning task with one-sided feedback as follows. On each round, a learner is shown a message and asked to predict its label from $\{-1, 1\}$ for spam and ham respectively. When a positive label is predicted, the true label is revealed to the learner and it may update its hypothesis. If the learner wishes to sample the label for a message it predicts to be spam, it directs that message into the inbox by predicting a positive label. Thus, label requests have the same cost as false positives.

We map spam messages to feature vectors using the methodology from [20], which has produced state of the art results with spam data. We use a 4-mer feature space, which consists of the set of all (possibly overlapping) contiguous character strings of length 4. The first 3000 characters of each email (including header, body, and any attachments) are mapped to sparse binary vectors in this feature space. These vectors are then normalized with the Euclidean norm to correct for differences in message length. This feature space is used for all test on spam data.

We tested three methods, Online SVMs, Perceptron with Margins, and classical Perceptron (which is equivalent to Perceptron with Margins where $m = 0$.) For Online SVMs, we used the Relaxed Online SVM described in [20], which has been shown to give nearly identical performance to the simple Online SVM on spam data with greatly reduced cost for large data sets. We set parameters $C = 100$ and buffer size 1000, updating on all margin errors. For Perceptron with Margins, we used $m = 2$ as a default parameter, and learning rate 1. For classical Perceptron, we used the same learning rate and $m = 0$.

We tested these methods on one-sided feedback in three ways: in their unmodified version, with the addition of Label Efficient sampling, and with the addition of Apple Tasting sampling. The results reported for the Label Efficient methods were for optimal values of the parameter $\beta > 0$ found in coarse grained trials on tuning data. Both Apple Testing and Label Efficient methods are randomized algorithms; we report average results over 10 trials with each. We tested other sampling strategies, including threshold biasing, ϵ-greedy, and the Softmax variant, but do not report these results as they were not competitive on these data sets. Finally, we report results on these data sets using full feedback for comparison.

Data Sets.

We use the two largest publicly available labeled data sets of spam and ham email, which are the `trec05p-1` data set of 92,189 messages with a 43% ham rate [5] and the `trec06p` data set of 37,822 messages with a 34% ham rate [4]. Both of these benchmark corpora have a canonical ordering for online learning which we use for repeatability. In preliminary tests and where parameter tuning was needed, we used a separate corpus, the smaller publicly available `spamassassin_corpus` of 6032 examples.

Evaluation Metrics.

Evaluating the performance of spam filtering methods is typically done by measuring the area under the ROC curve [5], which accounts for potentially uneven misclassification costs by assuming the ability to freely vary the classification threshold. In the one-sided feedback scenario, threshold modification after the fact is problematic, as the predicted class has implications on what feedback was available during learning. Thus, we evaluate performance using precision, recall, and the F-measure from the classifier's actual thresh-

Table 1: Results for Email Spam filtering. We report F1 score, Recall, Precision, number of False Negatives (lost ham) and number of False Positives (spam in inbox) for with one-sided feedback. We report results with full feedback for comparison.

trec05p-1	F1	Rec.	Prec.	# FN	# FP
Online SVM					
Margins	0.993	0.996	0.990	141	396
Label Efficient	0.991	0.997	0.985	135	609
With AppleTaste	0.989	0.996	0.981	145	757
Full Fb.	0.996	0.997	0.995	121	200
Pcptrn. Mgn.					
Margins	0.990	0.998	0.983	68	697
Label Efficient	0.989	0.998	0.980	68	785
With AppleTaste	0.984	0.998	0.970	91	1211
Full Fb.	0.994	0.996	0.992	153	318
Perceptron					
No Margins	0.516	0.348	0.999	25686	2
Label Efficient	0.960	0.964	0.956	1430	1748
With AppleTaste	0.930	0.920	0.940	3143	2330
Full Fb.	0.991	0.991	0.991	343	343

trec06p	F1	Rec.	Prec.	# FN	# FP
Online SVM					
Margins	0.988	0.996	0.981	51	253
Label Efficient	0.984	0.995	0.973	67	362
With AppleTaste	0.976	0.994	0.958	75	557
Full Fb.	0.993	0.994	0.992	81	100
Pcptrn. Mgn.					
Margins	0.984	0.998	0.970	32	397
Label Efficient	0.983	0.997	0.968	35	423
With AppleTaste	0.975	0.997	0.953	44	627
Full Fb.	0.993	0.995	0.991	69	111
Perceptron					
No Margins	0.005	0.003	0.945	12875	2
Label Efficient	0.938	0.943	0.933	738	876
With AppleTaste	0.891	0.886	0.897	1475	1309
Full Fb.	0.986	0.987	0.986	174	180

old. However, as a sanity check, we did calculate the ROC curve areas for each of the results, and these results agreed with the trends reported in this section.

Precision (P), recall (R), and the F_α measure are defined in terms of true positives (ham in the inbox), false positives (spam in the inbox), true negatives (spam in the spam box) and false negatives (ham in the spam box). When a learner makes a label requests, it is counted as a false positive when the example is a negative. Label requests for positive examples are counted as true positives. The measures are computed as follows:

$$P = \frac{TP}{TP+FP} \qquad R = \frac{TP}{TP+FN} \qquad F_\alpha = \frac{(1+\alpha)(P \cdot R)}{\alpha P + R}$$

The F_α measure gives a single number summary of classifier performance, where the parameter α determines how much weight to assign to precision and recall. We report the F1 measure results as a conservative view; we computed scores for F2 through F4 and found they emphasized our reported trends.

Results.

The results for experiments on both data sets are reported in Table 1. In general, they show a clear win for the margin-based methods, which had the highest F1 scores and lowest false positive rate, which is equivalent to making the fewest label requests. A spam filter using the unmodified Online SVM or Perceptron with Margins would place roughly half as much spam in the user's inbox as the Apple Tasting methods, while making roughly the same amount (or fewer) of misclassifications on ham. These results were not far from those achieved with full feedback.

As expected, classical Perceptron algorithm was defeated by the one-sided feedback scenario. Of the three methods of fixing classical Perceptron, the addition of margins (turning classical Perceptron to Perceptron with Margins) was more effective than either the Label Efficient or Apple Tasting strategies with classical Perceptron on all one-sided feedback tests. This supports the notion that the margin-based greedy exploration strategy is effective with class distributions.

Finally, the Label Efficient method out-performed the Apple Tasting method on all trials. Recall that the Label Efficient method also contains implicit use of uncertainty sampling, while Apple Tasting relies on uniform sampling of the negative predictions. This supports the claim that active learning is a good strategy for one-sided learning.

6.2 Minority Class Problems in Document Classification

In the previous experiments, the margin-based methods performed nearly as well from one-sided feedback as from full-feedback. This was true for the case in which the class distribution was roughly even. We now turn to the case in which the positive class is a minority class. As discussed in Section 5, the minority class case can pose problems for learning from one-sided feedback. We examine two methods of cost weighting to address this issue: example weighting and threshold biasing. Our initial tests showed that cost weighting is necessary with all of the methods on minority class problems.

For motivation, we consider the scenario in which a user wishes to be given documents of a specified class from a document stream. The user is only able to provide feedback on those documents that she actually sees, and she wishes to see as few irrelevant documents as possible while missing very few relevant ones. This is a special case of learning from *relevance feedback*, and has wide application in fields such as information retrieval, and data mining of text streams such as news articles or communications monitoring. In such scenarios, there are often many fewer relevant documents than irrelevant ones, creating a minority class distribution.

Data Sets.

We explore this scenario using the data from the Reuters-21578 data set for text classification, a standard benchmark data set [10]. We used the Mod-Apte split, and used the 9603 documents in the training data set in their given order to create an online learning task. The smaller test data set was used for initial tests and parameter tuning. We created binary classification tasks for each of the ten most common document classes to create a set of one-against-rest classification tasks. We used word stemming, removed stop words, and used normalized binary word-based feature

Table 2: Threshold Biasing on Minority-Class Document Classification. Results show best F1 score for each method over all parameter values.

Online SVM	Marg. Only	Apple T.	Label Eff.
earn (30%)	**0.943**	0.892	0.941
acq (17%)	**0.741**	0.538	0.723
money-fx (5.6%)	0.475	0.390	**0.611**
grain (4.5%)	**0.778**	0.454	0.763
crude (4.0%)	**0.723**	0.372	0.692
trade (3.8%)	0.335	0.325	**0.351**
interest (3.6%)	**0.518**	0.291	0.485
wheat (2.2%)	**0.716**	0.282	0.699
ship (2.0%)	0.287	0.220	**0.291**
corn (1.9%)	**0.538**	0.149	0.504
Avg. F1	0.606	0.410	0.608

Percpt. Margin.	Marg. Only	Apple T.	Label Eff.
earn (30%)	**0.934**	0.902	0.933
acq (17%)	**0.836**	0.734	0.814
money-fx (5.6%)	0.379	**0.410**	0.322
grain (4.5%)	**0.733**	0.362	0.728
crude (4.0%)	**0.690**	0.304	0.682
trade (3.8%)	0.005	**0.232**	0.215
interest (3.6%)	0.207	0.258	**0.516**
wheat (2.2%)	**0.617**	0.179	0.582
ship (2.0%)	0.010	**0.112**	0.010
corn (1.9%)	0.477	0.108	**0.480**
Avg. F1	0.488	0.360	0.528

Table 3: Example Weighting on Minority-Class Document Classification. Results show best F1 score for each method over all parameter values.

Online SVM	Marg. Only	Apple T.	Label Eff.
earn (30%)	**0.939**	0.923	0.937
acq (17%)	0.663	**0.781**	0.658
money-fx (5.6%)	0.328	0.471	**0.658**
grain (4.5%)	**0.656**	0.515	0.640
crude (4.0%)	**0.729**	0.433	0.706
trade (3.8%)	**0.437**	0.287	0.183
interest (3.6%)	0.460	0.482	**0.658**
wheat (2.2%)	**0.565**	0.375	0.560
ship (2.0%)	0.168	**0.181**	0.165
corn (1.9%)	**0.507**	0.204	0.499
Avg. F1	0.554	0.460	0.561

Percpt. Margin.	Marg. Only	Apple T.	Label Eff.
earn (30%)	**0.932**	0.899	0.932
acq (17%)	**0.834**	0.734	**0.834**
money-fx (5.6%)	0.574	0.547	**0.662**
grain (4.5%)	**0.735**	0.559	0.720
crude (4.0%)	**0.687**	0.539	0.670
trade (3.8%)	0.646	0.443	**0.650**
interest (3.6%)	**0.606**	0.433	0.570
wheat (2.2%)	**0.690**	0.421	0.634
ship (2.0%)	**0.512**	0.367	0.503
corn (1.9%)	0.586	0.389	**0.587**
Avg. F1	**0.680**	0.536	**0.680**

vectors. We removed any documents that did not contain at least one word in the text body. We also seeded each data set by moving the first occurrence of a positive example to the beginning of the online learning task. The data sets all contained minority positive class distributions, ranging from 30% positive down to 1.9% positive.

Experimental Setup.

As before, we tested Online SVMs and Perceptron with Margins in their unmodified forms, with Apple Tasting, and with the Label Efficient method. We used two methods of cost weighting: example weighting and threshold biasing. For each weighting method, we tested a spectrum of weighting parameter values and report the best $F1$ score for each learning and weighting method. The threshold biasing values we tested were $\{0, 0.008, 0.016, 0.032, 0.064, 0.128, 0.25, .75, 1\}$. For SVM example weighting, we fixed C_{pos} at 100 (consistent with initial on the separate tuning data), and adjusted C_{neg} from 0.001 to 128 in rough power of two increments to create the relative cost weighting. For Perceptron with Margins, we used the weighted Perceptron update rule described earlier, and held c_{neg} at 1 while testing c_{pos} in powers of two from 1 to 128. The β parameter for the Label Efficient method was set to 0.0001, set by initial trials on the separate tuning data.

Results.

The results reported in Tables 2 and 3 do not show a single clear winner across all trials. However, they do reveal some important general trends. First, minority-class problems clearly present difficulties for all approaches to one-sided feedback. The strongest performance was on the largest minority classes, `earn` and `acq`, while the weakest performances were on several of the smallest classes. Second, the margin-based and Label Efficient methods both out-perform the Apple Tasting methods on a wide majority of trials, with both Online SVMs and Perceptron with Margins. Third, the margin-based methods out-performed both Apple Tasting and Label Efficient methods on a majority of tests. However, on some tests, the margin-based methods were not able to learn effectively, suffering ratcheting as on the `ship` and `trade` results with threshold biasing for Perceptron with Margins. These few failures are typified by the extreme minority class distribution. Fourth, we note that threshold biasing was the more effective cost weighting method with Online SVMs, but example weighting was more effective with Perceptron with Margins. These results show that different data tasks may require different approaches for best results under one-sided learning from skewed data sets, but that margin-based methods would be a strong first choice.

7. CONCLUSIONS

The goal of this paper was to show that active learning improves performance from one-sided feedback. We have shown that the Label Efficient active learner can be applied to one-sided learning with good results. We have also showed that margin-based learners are active learners in the one-sided feedback scenario, and can exceed the performance of both the Apple Tasting and Label Efficient methods under many circumstances. One interesting ancillary contribution is the notion that margin-based learners may be used in conjunction with Apple Tasting or Label Efficient methods for additional robustness to pathological cases.

The Label Efficient method is well designed for active learning in an online setting, and adapts naturally to the problem of one-sided feedback. While margin-based methods are not generally considered to be active learners, we have shown that they do perform implicit active learning using uncertainty sampling in the one-sided feedback scenario. The margin-based methods gave best performance on spam data, approaching that of the full-feedback scenario. This performance was possible because the data contains relatively even class distribution and is quite linearly separable. Performance on minority class problems was more mixed, with both margin-based methods and Label Efficient methods performing well. The greedy exploration strategy used by margin-based methods only failed on extreme minority class distributions, and even then only with certain cost weighting schemes.

There are numerous practical data mining applications that suffer from the problem of one-sided feedback. We have empirically demonstrated the utility of these methods in two important tasks. In particular, the result that near state-of-the-art spam filtering is possible from one-sided feedback has important implications for large scale email systems. Future applications involving one-sided feedback may range from geo-statistical data mining to discover ore and oil deposits to personalized online news agents that learn effectively from one-sided feedback about the relevance of shown articles.

While we have concerned ourselves with linearly separable problems in this paper, these approaches can easily be adapted to the non-linearly separable case through feature transformation using kernel methods. Lastly, future work in one-sided feedback includes the interesting idea of one-sided feedback and regression. Is it possible to learn not just a classification function, but a full regression function effectively from feedback only on examples predicted to be above a given score threshold? This may have considerable impact in a variety of fields, including online advertising.

8. REFERENCES

[1] N. Abe and T. Kamba. A web marketing system with automatic pricing. *Comput. Networks*, 33(1-6):775–788, 2000.

[2] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research*, 7:1205–1230, 2006.

[3] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Mach. Learn.*, 15(2):201–221, 1994.

[4] G. V. Cormack. TREC 2006 spam track overview. In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

[5] G. V. Cormack and T. R. Lynam. TREC 2005 spam track overview. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005.

[6] S. Dasgupta. Analysis of a greedy active learning strategy. *NIPS: Advances in Neural Information Processing Systems*, 2004.

[7] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[8] D. Helmbold and S. Panizza. Some label efficient learning results. In *COLT '97: Proceedings of the tenth annual conference on Computational learning theory*, pages 218–230, 1997.

[9] D. P. Helmbold, N. Littlestone, and P. M. Long. Apple tasting. *Inf. Comput.*, 161(2):85–139, 2000.

[10] S. Hettich and S. D. Bay. The UCI KDD archive. Technical report, 1999.

[11] W. Krauth and M. Mézard. Learning algorithms with optimal stability in neural networks. *Journal of Physics A*, 20(11):745–752, 1987.

[12] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, 1994.

[13] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, 1988.

[14] J. Platt. Sequenital minimal optimization: A fast algorithm for training support vect or machines. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[15] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[16] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 441–448, 2001.

[17] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Readings in information retrieval*, pages 355–364, 1997.

[18] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 839–846, 2000.

[19] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimizati on, and Beyond*. MIT Press, 2001.

[20] D. Sculley and G. Wachman. Relaxed online support vector machines for spam filtering. In *SIGIR '07: The Thirtieth Annual Conference on Research and Development in Information Retrieval Proceedings*, 2007.

[21] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.