A Unified Approach to Active Dual Supervision for Labeling Features and Examples

Josh Attenberg[1], Prem Melville[2], and Foster Provost[3]

[1] Polytechnic Institute of NYU, Brooklyn, NY 11201, josh@cis.poly.edu
[2] IBM Research, Yorktown Heights, NY 10598, pmelvil@us.ibm.com
[3] NYU Stern School of Business, New York, NY 10012, fprovost@stern.nyu.edu

Abstract. When faced with the task of building accurate classifiers, active learning is often a beneficial tool for minimizing the requisite costs of human annotation. Traditional active learning schemes query a human for labels on intelligently chosen examples. However, human effort can also be expended in collecting alternative forms of annotation. For example, one may attempt to learn a text classifier by labeling words associated with a class, instead of, or in addition to, documents. Learning from two different kinds of supervision adds a challenging dimension to the problem of active learning. In this paper, we present a unified approach to such active dual supervision: determining which feature or example a classifier is most likely to benefit from having labeled. Empirical results confirm that appropriately querying for both example and feature labels significantly reduces overall human effort—beyond what is possible through traditional one-dimensional active learning.

1 Introduction

Active learning has been often used to reduce the amount of supervision required for effective learning. Traditionally, active learning research has focused on querying an oracle for labels on potentially informative examples. However, labeling effort may be better spent on providing alternative forms of supervision. Consider, for example, the task of *sentiment detection*, where given a piece of text as input, the desired output is a label that indicates whether this text expresses a positive or negative opinion. This problem can be cast as a typical binary text classification task, where a learner is trained on a set of documents that have been labeled based on the sentiment expressed in them. Alternatively, one could provide *labeled features*: for example, in the domain of movie reviews, words that evoke positive sentiment (e.g., "mesmerizing", "thrilling", etc.) may be labeled positive, while words that evoke negative sentiment (e.g., "boring", "disappointing", etc.) may be labeled negative. Through this kind of annotation a human conveys prior linguistic experience with a word by a sentiment label that reflects the emotion that the word evokes. The general setting of learning from both labels on examples and features is referred to as *dual supervision*.

This setting arises more broadly in tasks where, in addition to labeled documents, it is possible to provide domain knowledge in the form of words or phrases [26] or more sophisticated linguistic features that associate strongly with a class. Recent work [5,23,12] has demonstrated that feature supervision can greatly reduce the number of labels required to build high-quality classifiers. In general, example and feature supervision are complementary, rather than redundant.

This difference in information naturally leads to the problem of *active dual supervision*, or, how best to query a human resource to collect document labels *and* feature labels simultaneously, with the objective of building the highest quality model at the lowest cost. Much of the literature on active learning has focused on example-only annotation for classification problems. Less attention has been devoted to simultaneously acquiring alternative forms of supervisory domain knowledge. An exception, Sindhwani et al. apply classical uncertainty and experimental design-based active learning schemes to select labels for examples and features separately [24]. The present paper makes the following significant improvements over this prior work:

- Sindhwani et al. [24], at each iteration, randomly acquire a label for either an example or feature, and then probe the corresponding active learner. Here, we propose a holistic approach to active dual supervision based on an Expected Utility (estimated risk minimization) framework—within which, by optimizing the trade-offs between the costs and benefits of the different types of acquisitions, we deterministically select the most informative examples or features for labeling.
- We provide an instantiation of this framework for a recently introduced generative approach to dual supervision, instead of the graph-based dual supervision models used by Sindhwani et al. This generative approach, Pooling Multinomials [12], is comparable in performance to graph-based approaches and does not rely on unlabeled data. This is important for the present work. The Pooling Multinomials approach used here can be trained rapidly in an online fashion, rendering the otherwise computationally complex Expected Utility framework tractable.

Empirical results show that not only are we effective at actively selecting features for labeling, but that our unified approach to active dual supervision is better than the active learning of either instances or features in isolation.[4]

2 Dual supervision

Most work in supervised learning has focused on learning from examples, each represented by a set of feature values and a class label. In dual supervision we consider an additional aspect: labels of features, which convey prior knowledge on associations of features to particular classes. This paper focuses solely on text classification and all features represent term-frequencies of words; therefore, we use *feature* and *word* interchangeably.

[4] A preliminary version of this work appeared in [15].

While the active learning schemes explored in this paper are broadly applicable to any learner that can support dual supervision, we choose to focus on active learning for the Pooling Multinomials classifier [12] described below.

2.1 Pooling Multinomials

We introduce the Pooling Multinomials classifier as an approach to incorporate prior lexical knowledge into supervised learning for improved text classification. In the context of sentiment analysis, such lexical knowledge is available as the prior sentiment-polarity of words, while for classification, this knowledge comes from a human's term/class associations. Pooling Multinomials classifies unlabeled examples just as in multinomial Naïve Bayes classification, by predicting the class with the maximum likelihood, given by $argmax_{c_j} P(c_j) \prod_i P(w_i|c_j)$; where $P(c_j)$ is the prior probability of class c_j, and $P(w_i|c_j)$ is the probability of word w_i appearing in a document of class c_j. In the absence of background knowledge about the class distribution, we estimate the class priors $P(c_j)$ solely from the training data. However, unlike regular Naïve Bayes, the conditional probabilities $P(w_i|c_j)$ are computed using both labeled examples and labeled features. Given two models built using labeled examples and labeled features, the multinomial parameters of such models can be aggregated through a convex combination, $P(w_i|c_j) = \alpha P_e(w_i|c_j) + (1-\alpha)P_f(w_i|c_j)$; where $P_e(w_i|c_j)$ and $P_f(w_i|c_j)$ represent the probability assigned by using the example labels and feature labels respectively, and α is the weight for combining these distributions. The weight indicates a level of confidence in each source of information, and Melville et al. [12] explore ways of automatically selecting this weight. However, in order to avoid confusion of our results with the choice of weight-selection mechanism, here we make the simplifying assumption that the two experts based on instance and feature labels are equally valuable, and as such set α to 0.5. The derivation and details of these models are not directly relevant to this paper, but can be found in [12].

Note that, though Pooling Multinomials is based on a Naïve Bayes generative model, it is a state-of-the-art approach for text classification. Our empirical results show that Pooling Multinomials outperforms linear SVMs, a popular technique for performing text classification that lacks a mechanism for handling feature labels. Melville et al. demonstrated that Pooling Multinomials performs better than alternative approaches to incorporating labeled features [12].

2.2 Experimental setup

We conduct experiments on four binary text classification data set. The movies data set (2,000 examples, 5,000 features), introduced by Pang et al. [16] poses the task of classifying sentiment in movie reviews as positive or negative. *Politics* (107 examples, 1500 features) is based on posts from political blogs that were labeled as expressing positive or negative sentiments towards presidential candidates [12]. The *Baseball* (1,988 examples, 1,500 features) and *Science* (20,000 examples, 1,500 features) data sets are drawn from the 20-newsgroups[5] text collection where the task is to assign messages into the newsgroup in which they appeared. When performing analysis on these data sets, we use a bag-of-words representation, where documents are represented by term frequencies of the most frequent terms across all documents. All subsequent experiments present results averaged over 10-folds of cross validation.

2.3 Learning from example vs. feature labels

Dual supervision makes it possible to learn from labeled examples and labeled features simultaneously. As in most supervised learning tasks, one would expect more labeled data of either form to lead to more accurate models. In this section we explore the influence of increased number of instance labels and feature labels independently, and also in tandem.

As with any active learning research, in order to study the effect of increasing number of labels we simulate a human oracle labeling data. In the case of examples this is straightforward, since all examples in these data sets have labels. However, in the case of features, we do not have a gold-standard set of feature labels. Ideally, we should have a human expert in the loop labeling each feature selected. However, such a manual process is not feasible for large scale, repeatable experiments. In order to simulate human responses to queries for feature labels, we construct a *feature oracle* in the following manner (as done in [5,24]). The information gain of words with respect to the known true class labels in the data set is computed using binary feature representations. Next, out of all available terms representing the data, the top $\sim \frac{1}{5}$ as ranked by information gain are assigned a label. This label is the class in which the word appears more frequently, corrected by the differences in base rate. The oracle returns a "don't know" response for the remaining words. As a result, this oracle simulates a human domain expert who is able to recognize and label the relevant task-specific words while being unable to assign a label to non-polar terms.

To demonstrate the basic value of dual supervision, Fig. 1 compares three schemes: Instances-then-features, Features-then-instances, and Passive Interleaving on the Movies data set. All three begin with a base set of training data, including labels for 10 randomly selected instances and 10 randomly selected features. As the name suggests, *Instances-then-features*, provides labels for randomly selected instances until all instances have been labeled, and then switches to labeling features. Similarly, *Features-then-instances* acquires labels for randomly selected features first and then switches to getting instance labels. In *Passive Interleaving* we probabilistically switch between issuing queries for randomly chosen instance and feature labels. In particular, at each step we choose to query for an instance with probability 0.36, otherwise we query for a feature label. The instance-query rate of 0.36 is selected based on the ratio of available instances (1,800) to available features (5,000) in the Movies set. For the learning curves presented in Fig. 1, the x-axis corresponds to the number of queries

[5] `http://archive.ics.uci.edu/ml/`

issued. As discussed earlier, in the case of features, the oracle may respond to a query with a class label or may issue a "don't know" response, indicating that no label is available. As such, the number of feature-queries on the x-axis does not correspond to the number of actual known feature labels. We would expect that on average 1 in 5 feature-label queries prompts a response from the feature oracle that results in a known feature label being provided.

At the end of the learning curves, each method has labels for all available instances and features; and as such, the last points of all three curves are identical. The results show that fixing the number of labeled features, and increasing the number of labeled instances steadily improves classification accuracy. This is what one would expect from traditional supervised learning curves. More interestingly, the results also indicate that we can fix the number of instances, and improve accuracy by labeling more features. Finally, results on Passive Interleaving show that though both feature labels and example labels are beneficial by themselves, dual supervision which exploits the interaction of examples and features does in fact benefit from acquiring both types of labels concurrently.

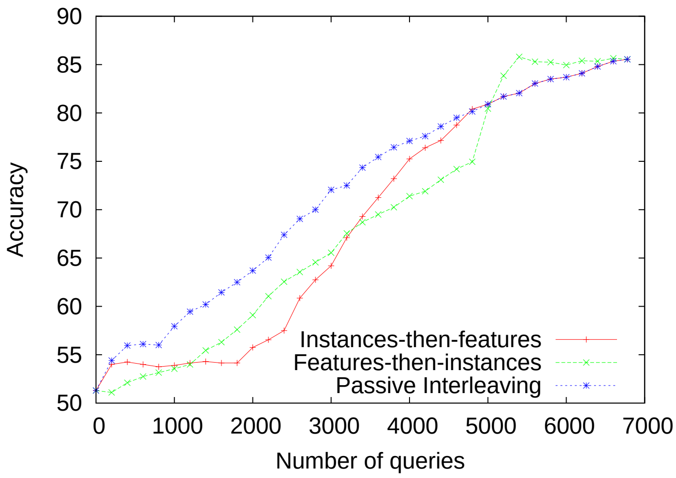


Fig. 1. Comparing the effect of instance and feature label acquisition in dual supervision.

For all results above, we are selecting instances and/or features to be labeled uniformly at random. Based on previous work in active learning one would expect that we can select instances to be labeled more efficiently, by having the learner decide which instances it is most likely to benefit from. The results in this section suggests that actively selecting features to be labeled may also be beneficial. Furthermore, the Passive Interleaving results suggest that an ideal active dual supervision scheme would actively select both instances and features for labeling. We begin by exploring active learning for feature labels in the next section, and then consider the simultaneous selection of instances and features in Sec. 4.

3 Acquiring feature labels

Traditional active learning has primarily focused on selecting unlabeled *instances* to be labeled. The dual-supervision setting adds an additional aspect to active learning where labels may be acquired for features as well. In this section we focus on the task of active learning applied only to feature-label acquisition.

3.1 Feature uncertainty vs. certainty

In the traditional active learning setting, Uncertainty Sampling has earned a reputation as an effective and intuitive technique for selecting instances to get labeled [8]. In this approach, labels are requested for instances for which the current model gives the highest degree of uncertainty—which for binary classification with 0/1 loss are those instances nearest to the current classification boundary. Despite its simplicity, Uncertainty Sampling is often quite effective in practice, and has therefore become a standard for comparison for active learning research. This raises the question of whether one can apply the same principle to feature-label acquisition: select unlabeled features that the current model is most uncertain about.

Much like instance uncertainty, feature uncertainty can be measured in different ways, depending on the underlying method used for dual supervision. Since Pooling Multinomials builds a multinomial Naïve Bayes model, we can directly use the model's conditional probabilities of each feature f given a class. For ease of exposition we refer to the two classes in binary classification as *postive* (+) and *negative* (-), without loss of generality. Given the probabilities of f belonging to the positive and negative class, $P(f|+)$ and $P(f|-)$, we compute the uncertainty for f using the absolute value of the log-odds ratio, i.e.,

$$abs\left(\log\left(\frac{P(f|+)}{P(f|-)}\right)\right) \qquad (1)$$

The smaller this value, the more uncertain the model is about the feature's class association. In every iteration of active learning we can select the features with the lowest certainty scores. We refer to this approach as *Feature Uncertainty*.

Though Uncertainty Sampling for features seems like an appealing notion, it may not lead to better models. If a classifier is uncertain about a feature, it may have insufficient information about this feature and may indeed benefit from learning its label. However, it is also quite likely that a feature has a low certainty score because it does not carry much discriminative information about the classes. In the context of sentiment detection, one would expect that neutral/non-polar words will appear to be uncertain words. For example, words such as "the" which are unlikely to help in discriminating between classes, are also likely to be considered the most uncertain. As we shortly report, on the *Movies* dataset, Feature Uncertainty ends up wasting queries on such words ending up with performance inferior to random feature queries. What works significantly better is an alternative strategy that acquires labels for features in *descending* order of the score in Eq 1. We refer to this approach as *Feature Certainty.* A similar approach provided the best results in previous work [24]. We further improve on these results by the method described below.

3.2 Expected feature utility

The intuition underlying the feature certainty heuristic is that it serves to confirm or correct the orientation of model probabilities on different words during the active learning process. One can argue that feature certainty is also suboptimal in that queries may be wasted simply confirming confident predictions, which is of limited utility to the model. An alternative to using a certainty-based heuristic, is to directly estimate the expected value of acquiring each feature label. Such Expected Utility (Estimated Risk Minimization) approaches have been applied successfully to traditional active learning [19], and to active feature-value acquisition [14]. In this section we describe how this Expected Utility framework can be adapted for feature-label acquisition.

At every step of active learning for features, the next feature selected for labeling is the one that will result in the highest estimated improvement in classifier performance. Since the true labels of the unlabeled features are unknown prior to acquisition, it is necessary to estimate the potential impact of every feature query for all possible outcomes.[6] Hence, the decision-theoretic optimal policy is to ask for feature labels which, once incorporated into the data, will result in the highest increase in classification performance in *expectation.*

If f_j is the label of the j-th feature, and q_j is the query for this feature's label, then the Expected Utility of a feature query q_j can be computed as:

$$EU(q_j) = \sum_{k=1}^{K} P(f_j = c_k)\mathcal{U}(f_j = c_k) \tag{2}$$

Where $P(f_j = c_k)$ is the probability that f_j will be labeled with class c_k, and $\mathcal{U}(f_j = c_k)$ is the utility to the model of knowing that f_j has the label c_k. In practice, the true values of these two quantities are unknown, and the main challenge of any Expected Utility approach is to accurately estimate these quantities from the data currently available.

A direct way to estimate the utility of a feature label is to measure expected classification accuracy. However, small changes in the probabilistic model that result from acquiring a single additional feature label may not be reflected by a change in accuracy. Therefore, we use a finer-grained measure of classifier performance, Log Gain, which is computed as follows. For a model induced from a training set T, let $\hat{P}(c_k|x_i)$ be the probability estimated by the model that instance x_i belongs to class c_k; and $\mathbb{I}$ is an indicator function such that $\mathbb{I}(c_k, x_i) = 1$ if c_k is the correct class for x_i and $\mathbb{I}(c_k, x_i) = 0$, otherwise. Log

[6] In the case of binary classification, the possible outcomes are a *positive* or *negative* label for a queried feature.

Gain is then defined as:

$$LG(x_i) = -\sum_{k=1}^{K} \mathbb{I}(c_k) \log \hat{P}(c_k|x_i) \tag{3}$$

Then the utility of a classifier, $\mathcal{U}$, can be measured by summing the Log Gain for all instances in the training set T. A lower value of Log Gain indicates a better classifier performance. We present an empirical comparison of different utility measures in Sec. 5.

In Eq. 2, apart from the measure of utility, we also do not know the true probability distribution of labels for the feature under consideration. This too can be estimated from the training data, by seeing how frequently the word appears in documents of each class. In the instance-based multinomial Naïve Bayes model we already collect these statistics in order to determine the conditional probability of a class given a word, i.e. $P(f_j|c_k)$. We can use these probabilities to get an estimate of the feature label distribution, $\hat{P}(f_j = c_k) = \frac{P(f_j|c_k)}{\sum_{k=1}^{K} P(f_j|c_k)}$.

Given the estimated values of the feature-label distribution and the utility of a particular feature query outcome, we can now estimate the Expected Utility of each unknown feature, selecting the features with the highest Expected Utility for labeling.

Though theoretically appealing, this approach can be computationally intensive if Expected Utility estimation is performed on all unknown features. In the worst case this requires building and evaluating models for each possible outcome of each unlabeled feature. In a setting with m features and K classes, this approach requires training $O(mK)$ classifiers. However, the complexity of the approach can be significantly alleviated by only applying Expected Utility evaluation to a sub-sample of all unlabeled features. Given the large number of features with no true class labels, selecting a sample of available features uniformly at random may be sub-optimal. Instead we select a sample of features based on Feature Certainty. In particular we select the top 100 unknown features that the current model is most certain about, and identify the features in this pool with the highest Expected Utility. We refer to this approach as *Feature Utility*. We use Feature Certainty to sub-sample the available feature queries, since this approach is more likely to select features for which the label is known by the oracle.

3.3 Active learning with feature labels

We ran experiments comparing the three different active learning approaches described above on the Movies data set. Here we begin with a model trained on a random selection of 10 labeled features and 100 labeled instances.

The experiments in this section focus only on the selection of *features* to be labeled— in each iteration of active learning we select the next 10 feature-label queries, based on Feature Uncertainty, Feature Certainty, or Feature Utility. As a baseline, we also compare to the performance of a model that selects features uniformly at random. Our results are presented in Fig. 2.

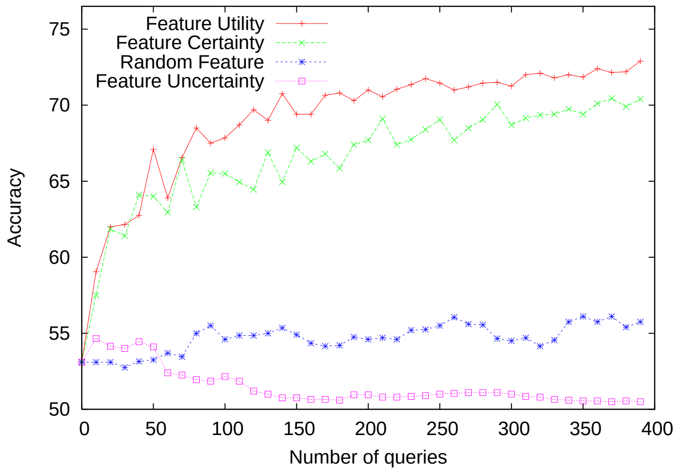


Fig. 2. Comparing different active learning approaches for acquiring feature labels.

The results show that Feature Uncertainty performs worse than random sampling; however, the converse approach of Feature Certainty does remarkably well. This is in line with out discussion above in Sec. 3.1 The results for Feature Utility show that estimating the expected impact of potential labels for features does in fact perform much better than feature certainty. The results confirm that despite our crude estimations in Eq. 2, Feature Utility is an effective approach to active learning of feature labels. Furthermore, we demonstrate that by applying the approach to only a small sub-sample of certain features, we are able to make this method computationally feasible to use in practice. Increasing the size of the sample of candidate feature queries is likely to improve performance, at the cost of increased time in selecting queries.

4 Active dual supervision

In the previous section we demonstrated that actively selecting informative features to be labeled performs significantly better than random selection. This conclusion is congruent with the rich body of work showing the benefits over random selction of actively selecting *instances* for labeling. Furthermore, we have demonstrated in Sec. 2 that randomly selecting both feature and instance labels in tandem is better than either in isolation. An ideal active scheme should be able to assess if an instance or feature would be more beneficial at each step, and select the most informative instance or feature for labeling.

Fortunately, the Expected Utility method is very flexible, capable of addressing both types of acquisition within a single framework. Since the measure of utility is independent of the type of supervision and only dependent on the resulting classifier, we can estimate the expected utility of different forms of acquisitions in the same manner. For instance, Saar-Tsechansky et al. [20] use such an approach to estimate the utility of acquiring class labels and feature values (not labels), within one unified framework. A similar technique is utilized here, yielding a holistic approach to active dual supervision, where the Expected

Utility of an instance or feature label query, q, can be computed as

$$EU(q) = \sum_{k=1}^{K} P(q = c_k) \frac{\mathcal{U}(q = c_k)}{\omega_q} \tag{4}$$

where ω_q is the cost of the query q, $P(q = c_k)$ is the probability of the instance or feature queried being labeled as class c_k, and utility $\mathcal{U}$ can be computed as in Eq. 3. By evaluating instances and features in the same units, and by measuring utility per unit cost of acquisition, such a framework facilitates explicit optimization of the trade-offs between the costs and benefits of the different types of acquisitions. For the sake of simplicity, we assume equal costs of acquisitions in this paper, i.e., $\omega_q = 1$. But in principle this framework can be used even if the cost of acquiring a feature label is different from the cost of acquiring an instance label. We refer to this combined instance and feature acquisition approach simply as Expected Utility. As before, we speed up query selection by first sub-sampling 100 features based on certainty and 100 instances at random, and then evaluate the Expected Utility on this candidate set.

Experiments are performed as before, comparing Expected Utility to Feature Utility and Passive Interleaving. Recall that Passive Interleaving corresponds to probabilistically interleaving queries for randomly chosen, not actively chosen, examples and features. For completeness, we also compare with an instance-only active learning method. Namely, we use Uncertainty Sampling [8], which has been shown to be a computationally efficient and effective approach in the literature. In particular, we select unlabeled examples to be labeled in order of decreasing uncertainty, measured in terms of the margin, as done in [13]. The margin on an unlabeled example is defined as the absolute difference between the class probabilities predicted by the classifier for the given example, i.e., $|P(+|x) - P(-|x)|$. We refer to the selection of instances based on this uncertainty as Instance Uncertainty, in order to distinguish it from Feature Uncertainty.

We compare the performance of any two methods, A and B, by computing the percentage reduction in classification error rate obtained by A over B at each acquisition phase and report the average reduction over all acquisition phases. We refer to this average as the *average percentage error reduction*. The reduction in error obtained with policy A over the error of policy B is considered to be significant if the errors produced by policy A are lower than the corresponding errors (i.e., at the same acquisition phase) produced by policy B, according to a paired t-test ($p < 0.05$) across all the acquisition phases [13]. In our experiments, we compare all active methods using Passive Interleaving as our baseline (B). Our results are summarized in Table 1, where statistically significant improvements over Passive Interleaving are shown in bold. We also present learning curves on three datasets in Fig. 3.

We observe that actively selecting instances or features for labeling is better than randomly selecting either. In general, effectively selecting features' labels, via Feature Utility, does even better than actively selecting only instances. However, in some cases, the advantage of actively selecting only one type of supervision is out-weighed by randomly selecting both instances and features in

Data Set	Instance Uncertainty	Feature Utility	Expected Utility
Movies	**9.90**	**31.18**	**36.52**
Science	−3.88	**13.05**	**25.24**
Baseball	−101.98	−2.59	**39.61**
Politics	−7.04	−7.16	1.48

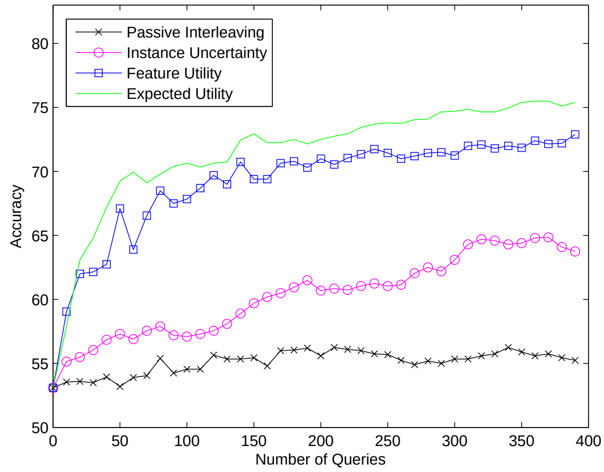
Table 1. Error reduction(%) of active learning approaches compared to Passive Interleaving.

tandem. This phenomenon can be seen for *Baseball* in Fig. 3(b), where Uncertainty Sampling is clearly less effective than Passive Interleaving; even Feature Utility's initial advantage is lost by randomly selecting some instance labels in Passive Interleaving. However, by estimating the benefit of each type of acquisition at each step, the holistic Expected Utility approach is able to outperform active learning on instances and features in isolation, as well as randomly interleaving instance-labels with feature-labels. The savings from using Expected Utility for active dual supervision can be quite substantial. For instance, this approach achieves an accuracy of 75% on *Movies* with only 350 queries, while Passive Interleaving requires 10 times the number of queries to reach the same performance. The average reduction in error using Expected Utility over Passive Interleaving ranges from a 1.5% on *Politics* to 40% on *Baseball*.

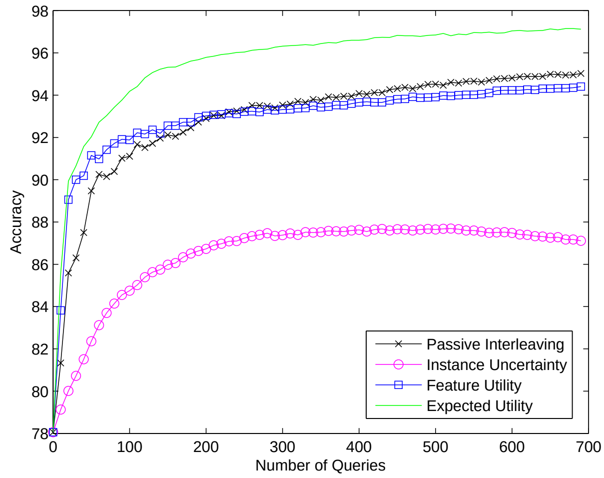
5 Choice of utility measure

Up to now, in order to evaluate the utility of a potential label, we have measured the average log gain (Eq. 3) calculated on the training set by a classifier trained with the inclusion of the associated labeling. However, there are alternative ways to measure utility. One obvious choice is to measure utility based on classification accuracy on the training set. Both log gain and accuracy are "supervised" utilities measures, meaning that they are computed on examples for which we have labels. In contrast, in previous work in the traditional instance labeling setting, Roy and McCallum [19] use two "unsupervised" measures, computed on the pool of unlabeled examples. Their motivating objective is different from our desire to estimate directly the expected improvement in generalization performance. Instead, they try to "select those examples which maximizes [sic] the sharpness of the learner's posterior belief about the unlabeled examples" [19]. Namely, they use entropy and $(1 - \arg\max_{c_k} \hat{P}(c_k|x))$, where $\hat{P}(c_k|x)$ is the resulting classifier's predicted probability for class c_k . We will refer to the latter measure as maximum posterior. Lower values of these measures correspond to higher utilities in their setting.

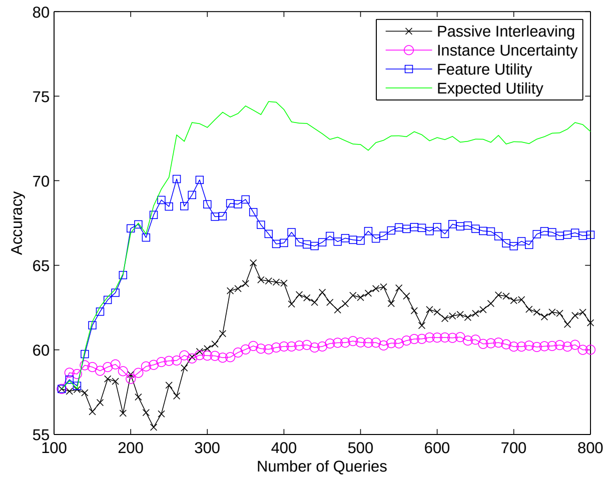
We ran experiments as before for Feature Utility and the combined Expected Utility on the Movies data set, comparing five different measures of utility: log gain, accuracy, and entropy estimated on the training set, as well entropy and maximum posterior on the unlabeled pool. The results for Expected Utility are presented in Fig. 4. The performance of entropy and maximum posterior on



(a) Movies



(b) Baseball



(c) Science

Fig. 3. Comparing Expected Utility to alternative label acquisition strategies.

the unlabeled pool, and accuracy are indistinguishable in the figure, and do not perform as well as the other measures. The results on Feature Utility (not shown) show a similar trend except that accuracy performs better than entropy and maximum posterior on the unlabeled pool, while still doing worse than log gain. By incorporating the predicted class probabilities, log gain is able to capture small changes in the classifier that may lead to an improvement that may not be reflected by a change in classification accuracy. Hence it tends to perform better than measuring utility based on accuracy. The unsupervised measures used by Roy and McCallum do not perform as well as they are focused on selecting examples that on average will make the predictions of the model the most certain on the not-yet-labeled examples. These results empirically support the use of log gain in the Expected Utility framework. For a deeper theoretical discussion of this measure see [20].

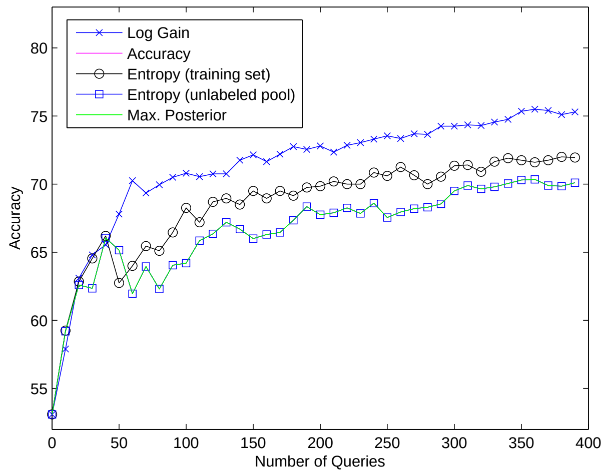


Fig. 4. Comparing different measures of utility. Accuracy, entropy (unlabeled) and max. posterior overlap.

6 Related work

Active learning in the context of dual supervision models is a new area of research with very little prior work, to the best of our knowledge. Most prior work in active learning has focused on pooled-based techniques, where examples from an unlabeled pool are selected for labeling [3]. In contrast, active feature-value acquisition [14] and *budgeted learning* [11] focus on estimating the value of acquiring missing features, but do not deal with the task of learning from feature *labels*. Raghavan and Allan [17] and Raghavan et al. [18] study the problem of *tandem learning* where they combine uncertainty sampling for instances along with co-occurence based interactive feature selection. Godbole et al. [7] propose notions of feature uncertainty and incorporate the acquired feature labels, into learning by creating one-term mini-documents. Druck et al. [6] perform active learning via feature labeling using several uncertainty reduction heuristics. Sindhwani et al. [24] also study the problem of active dual supervision, applied to a graph-based dual supervision method. They explore various heuristic approaches to active learning for instances and features separately. In order to interleave selections from both instances and features, they randomly probe an active instance learner or an active feature learner for the next query. In contrast, we take a holistic approach to active dual supervision, where by estimating the potential value of features and instances on the same scale, we select the type of acquisition that is most likely to benefit our classifier. While in principle the Expected Utility-based techniques presented here could be applied to any technique for dual supervision, Pooling Multinomials is well suited for our approach, since it can be implemented to be update-able (without retraining), making the computations in Expected Utility extremely efficient. It is a challenge to efficiently implement Expected Utility for the graph-based method used by Sindhwani et al., and this is a promising direction for future work.

In contemporaneous work, Attenberg et al. [1] propose active dual supervision as one possible solution to the cold start problem often faced by active learners in settings with high class skew. Additionally, they propose tasking human domain experts with seeking and finding useful feature values directly, as opposed to the query/respose approach seen here.

Learning from labeled examples and features via dual supervision is itself a new area of research. Sindhwani et al. [22] use a kernel-based framework to build dual supervision into co-clustering models. Sindhwani and Melville [23] apply similar ideas for graph-based sentiment analysis. Note that, dual supervision should not be confused with Co-Training [2], in which the description of examples can be divided into two distinct views i.e. disjoint feature sets. There have also been previous attempts at using only feature supervision, mostly along with unlabeled documents. Much of this work [21,25,10,4] has focused on using labeled features to generate *pseudo-labeled examples* that are then used with well-known models. In contrast, Druck et al. [5] constrain the outputs of a multinomial logistic regression model to match certain reference distributions associated with labeled features. In a similar vein, Liang et al. [9] learn from labeled examples and constrains on model predictions.

7 Conclusions and Future Work

This paper presents a unified framework for active dual supervision, where the relative benefit of each type of acquisition is assessed based on the expected improvement of the resulting classifier. We demonstrated that not only is combining example and feature labels beneficial for modeling, but that actively selecting the most informative examples and features for labeling can significantly reduce the burden of labeling such data. For simplicity, we did not consider the different costs of acquiring labels. Presumably labeling a feature versus labeling an instance could incur very different costs—which could be monetary costs or time taken for each annotation. The general Expected Utility framework we present can directly handle such cost-benefit trade-offs, and empirically validating this is an avenue for future work. Furthermore, the mixing of multinomials based on labeled features and labeled examples exerts a strong influence on the probability estimates produced, and therefore the choices made in active learning. Another direction for future work is the investigation of the mixing parameter, α, and its influence on active dual supervision.

Human oracles may be able to provide a much richer set of background information than can be expressed via individual token polarities. For instance, "yeah" and "right" may both be terms denoting a positive sentiment polarity, while "yeah, right" may be used with sarcasm with a negative connotation[7]. Extending models to incorporate higher order information from oracles is another good direction for future work.

The Expected Utility framework we propose is general, in that it can be applied to any learning algorithm that supports dual supervision. However, it is a significant challenge to devise methods to efficiently estimate the terms in Eq. 4 that are appropriate to the learner of choice. As shown in Sec. 5, even the choice of measure of utility is not obvious, and can make a significant difference in results. As such, adapting this framework to other learners, such as the graph-based approach in [24], is a challenging, but promising direction to explore.

Acknowledgments

We would like to thank Vikas Sindhwani for insightful discussions and help in preparing this document.

References

1. Attenberg, J., Melville, P., Provost, F.: Guided feature labeling for budget-sensitive learning under extreme class imbalance. In: BL-ICML '10: Workshop on Budgeted Learning (2010)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT (1998)
3. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine Learning 15(2), 201–221 (1994)
4. Dayanik, A., Lewis, D., Madigan, D., Menkov, V., Genkin, A.: Constructing informative prior distributions from domain knowledge in text classification. In: SIGIR (2006)
5. Druck, G., Mann, G., McCallum, A.: Learning from labeled features using generalized expectation criteria. In: SIGIR (2008)
6. Druck, G., Settles, B., McCallum, A.: Active learning by labeling features. In: EMNLP '09. pp. 81–90. Association for Computational Linguistics (2009)
7. Godbole, S., Harpale, A., Sarawagi, S., Chakrabarti, S.: Document classification through interactive supervision of document and term labels. In: PKDD (2004)

[7] We would like to thank an anonymous reviewer for this suggestion given to a preliminary version of this paper

8. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Proc. of 11th Intl. Conf. on Machine Learning (ICML-94) (Jul 1994)
9. Liang, P., Jordan, M.I., Klein, D.: Learning from measurements in exponential families. In: ICML (2009)
10. Liu, B., Li, X., Lee, W.S., Yu, P.: Text classification by labeling words. In: AAAI (2004)
11. Lizotte, D., Madani, O., Greiner, R.: Budgeted learning of naive-Bayes classifiers. In: UAI (2003)
12. Melville, P., Gryc, W., Lawrence, R.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: KDD (2009)
13. Melville, P., Mooney, R.J.: Diverse ensembles for active learning. In: Proc. of 21st Intl. Conf. on Machine Learning (ICML-2004) (2004)
14. Melville, P., Saar-Tsechansky, M., Provost, F., Mooney, R.: An expected utility approach to active feature-value acquisition. In: ICDM (2005)
15. Melville, P., Sindhwani, V.: Active dual supervision: Reducing the cost of annotating examples and features. In: NAACL HLT 2009 (2009)
16. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: EMNLP (2002)
17. Raghavan, H., Madani, O., Jones, R.: An interactive algorithm for asking and incorporating feature feedback into support vector machines. In: SIGIR (2007)
18. Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on features and instances. J. Mach. Learn. Res. 7 (2006)
19. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: ICML (2001)
20. Saar-Tsechansky, M., Melville, P., Provost, F.: Active feature-value acquisition. In: Management Science (2009)
21. Schapire, R.E., Rochery, M., Rahim, M.G., Gupta, N.: Incorporating prior knowledge into boosting. In: ICML (2002)
22. Sindhwani, V., Hu, J., Mojsilovic, A.: Regularized co-clustering with dual supervision. In: NIPS (2008)
23. Sindhwani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: ICDM (2008)
24. Sindhwani, V., Melville, P., Lawrence, R.: Uncertainty sampling and transductive experimental design for active dual supervision. In: ICML (2009)
25. Wu, X., Srihari, R.: Incorporating prior knowledge with weighted margin support vector machines. In: KDD (2004)
26. Zaidan, O.F., Eisner, J.: Modeling annotators: A generative approach to learning from annotator rationales. In: EMNLP (2008)