2013 IEEE 13th International Conference on Data Mining

Most-Surely vs. Least-Surely Uncertain

Manali Sharma* and Mustafa Bilgic†
Computer Science Department
Illinois Institute of Technology, Chicago, IL, USA
* *msharm11@hawk.iit.edu*
† *mbilgic@iit.edu*

***Abstract*—Active learning methods aim to choose the most informative instances to effectively learn a good classifier. Uncertainty sampling, arguably the most frequently utilized active learning strategy, selects instances which are uncertain according to the model. In this paper, we propose a framework that distinguishes between two types of uncertainties: a model is uncertain about an instance due to strong and conflicting evidence (most-surely uncertain) vs. a model is uncertain about an instance because it does not have conclusive evidence (least-surely uncertain). We show that making a distinction between these uncertainties makes a huge difference to the performance of active learning. We provide a mathematical formulation to distinguish between these uncertainties for naive Bayes, logistic regression and support vector machines and empirically evaluate our methods on several real-world datasets.**

***Keywords*-Active learning; uncertainty sampling**

I. Introduction

Active learning methods aim to learn the correct classification function by selecting the most informative instances according to the model and seeking labels for those instances from an expert [1]. The goal of active learning is to learn the classification function using the most cost-effective instances to reduce the time, effort and cost of the expert.

Many active learning methods have been developed in the past two decades. Uncertainty sampling is arguably the most frequently utilized method. This method is popular due to its simplicity and success. In this paper, we introduce a novel way to determine the cause of uncertainty of a model and use it to improve the label efficiency of active learning.

An underlying model's uncertainty can arise due to at least two reasons. (i) The model can be uncertain due to presence of strong, but conflicting evidence for each class. For example, in document classification, while some words in the document strongly pull the class label in one direction, other words strongly pull the class label in the opposite direction. In medical diagnosis, while some lab test results strongly suggest one disease, few others strongly suggest another disease. We call this type of uncertainty as *most-surely uncertain*. (ii) The model can be uncertain due to presence of weak evidence for each class. For example, in document classification, none of the words provide strong evidence for either class. In medical diagnosis, none of the lab test results provide a conclusive evidence for any disease. We call this type of uncertainty as *least-surely uncertain*.

Figure 1 depicts this phenomenon for binary classification. For *most-surely uncertain*, the attribute values pull strongly in opposing directions while for *least-surely uncertain*, none of the attribute values provide strong evidence for either class. In both cases, the underlying model is uncertain about the instances but the cause of uncertainty is different. It is worth noting that these uncertainties exist only with respect to the underlying model, which is trained on a relatively small training set in an active learning setting.

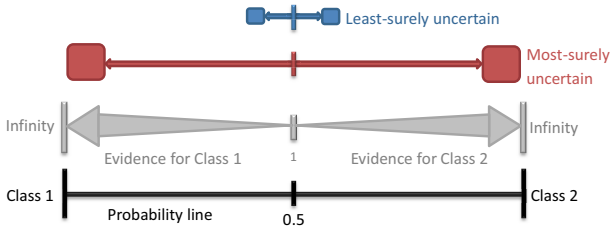

Figure 1. Most-surely vs. least-surely uncertain. A model's uncertainty for an instance is represented by the probability line. The *most-surely uncertain* represents a model's uncertainty on an instance due to strong evidence for each class, whereas *least-surely uncertain* represents a model's uncertainty on an instance due to weak evidence for each class. The values of evidence for each class range from 1 to infinity.

We introduce a formulation to distinguish between these two types of uncertainties. In particular, we provide formulations for naive Bayes, logistic regression and support vector machines. Through empirical evaluations on several real-world datasets, we show that distinguishing between these two types of uncertainties (*most-surely uncertain* and *least-surely uncertain*) makes a huge difference to the performance of active learning. We show that *most-surely uncertain* provides the most benefit for learning, drastically outperforming the regular uncertainty sampling.

The rest of the paper is organized as follows. First, we provide background on active learning and uncertainty sampling in Section II. Then, in Section III, we provide our problem formulation. In Section IV, we provide details of our experiments, datasets and evaluation strategies. In Section V, we provide the results of our experiments. Finally, we discuss related work in Section VI, present future work in Section VII and conclude in Section VIII.

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.15

II. Background

In this section, we first briefly describe active learning and then explain uncertainty sampling in detail. We assume that we are given a dataset $\mathcal{D}$ of instances consisting of attribute vector and label pairs $\langle x, y\rangle$. Each $x \in \mathcal{X}$ is described as a vector of f attributes $x \triangleq \langle a_1, a_2, \cdots, a_f\rangle$, each of which can be real-valued or discrete, whereas each $y \in \mathcal{Y}$ is discrete-valued $\mathcal{Y} \triangleq \{y_1, y_2, \cdots, y_l\}$. A small subset $\mathcal{L} \subset \mathcal{D}$ is the labeled set where the labels are known: $\mathcal{L} = \{\langle x, y\rangle\}$. The rest $\mathcal{U} = \mathcal{D}\backslash\mathcal{L}$ consists of the unlabeled instances whose labels are unknown: $\mathcal{U} = \{\langle x, ?\rangle\}$.

Active learning algorithm iteratively selects an instance $\langle x, ?\rangle \in \mathcal{U}$ and obtains the resulting target value y by querying an expert for its label and incorporating the new example $\langle x, y\rangle$ into its training set $\mathcal{L}$. This process continues until a stopping criterion is met, usually until a given budget, B, is exhausted. The goal of active learning is to learn the correct classification function $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ using minimal budget. Algorithm 1 describes this process more formally.

Algorithm 1 Budget-Based Active Learning

1: **Input:** $\mathcal{U}$ - unlabeled data, $\mathcal{L}$ - labeled data, θ - underlying classification model, B - budget
2: **repeat**
3: **for all** $\langle x, ?\rangle \in \mathcal{U}$ **do**
4: compute *utility*(x, θ)
5: **end for**
6: pick highest utility x^* and query its label
7: $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle x^*, y^*\rangle\}$
8: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle x^*, y^*\rangle\}$
9: Train θ on $\mathcal{L}$
10: **until** Budget B is exhausted; e.g., $|\mathcal{L}| = B$

A number of successful active learning methods have been developed in the past two decades. Examples include uncertainty sampling [2], query-by-committee [3], bias reduction [4], variance reduction [5], expected error reduction [6, 7], and many more. We refer the reader to [1] for a survey of active learning methods.

Arguably, the most-frequently utilized active learning strategy is uncertainty sampling[1], which is also the topic of this paper. Next, we describe uncertainty sampling in detail.

A. Uncertainty Sampling

Uncertainty sampling selects those instances for which the current model is most uncertain how to label [2]. These instances correspond to the ones that lie on the decision boundary of the current model.

Uncertainty of an underlying model can be measured in several ways. We present the three most common approaches. One approach is to use conditional entropy:

$$x^* = \underset{x\in\mathcal{U}}{\arg\max} - \sum_{y\in\mathcal{Y}} P_\theta(y|x)\log\left(P_\theta(y|x)\right) \quad (1)$$

where θ is the current model trained on $\mathcal{L}$ and $P_\theta(y|x)$ is the probability that instance x has label y. Another approach is to use maximum conditional:

$$x^* = \underset{x\in\mathcal{U}}{\operatorname{argmin}} \left(1 - \max_{y\in\mathcal{Y}} P_\theta(y|x)\right) \quad (2)$$

The last approach we discuss uses margin of confidence:

$$x^* = \underset{x\in\mathcal{U}}{\operatorname{argmin}} \left(P_\theta(y^{(m)}|x) - P_\theta(y^{(n)}|x)\right) \quad (3)$$

where, $y^{(m)}$ is the most likely label and $y^{(n)}$ is the next likely label for x. More formally, $y^{(m)} = \underset{y\in\mathcal{Y}}{\arg\max}\, P_\theta(y|x)$ and $y^{(n)} = \underset{y\in\mathcal{Y}\backslash\{y^{(m)}\}}{\arg\max}\, P_\theta(y|x)$.

When the task is binary classification, that is when $\mathcal{Y} = \{+1, -1\}$, each of these objective functions rank the instances $x \in \mathcal{U}$ in the same order, and the highest utility is achieved when $P_\theta(+1|x) = P_\theta(-1|x) = 0.5$.

In this paper, we distinguish between two types of uncertainties that we define next. On one extreme, the model is uncertain about an instance because the instance's attribute values provide equally strong evidence for each class. We call this kind of uncertainty *most-surely uncertain*. On the other extreme, the model is uncertain about an instance because the instance's attribute values provide very weak evidence for each class. We refer to this type of uncertainty as *least-surely uncertain*.

III. Problem Formulation

In this section, we first formally define *evidence* in the context of binary classification, $\mathcal{Y} = \{+1, -1\}$, using a naive Bayes classifier. Then, we show how the definition can be extended to logistic regression and support vector machines. Finally, we discuss how it can be generalized to multi-class classification domains.

A. Evidence using naive Bayes

A naive Bayes classifier uses the Bayes rule to compute $P(y|x)$ and assumes that the attributes a_i are conditionally independent given y:

$$P(y|x) = P(y|a_1, a_2, \cdots, a_f) = \frac{P(y)\prod_{a_i} P(a_i|y)}{P(a_1, a_2, \cdots, a_f)} \quad (4)$$

An instance can be classified based on the ratio of $\frac{P(+1|x)}{P(-1|x)}$:

$$y = \begin{cases} +1 & \text{if } \left(\frac{P(+1)}{P(-1)}\prod_{a_i}\frac{P(a_i|+1)}{P(a_i|-1)}\right) > 1 \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

[1] 1,130 citations on Google scholar on October 20th, 2013.

From Equation 5, it follows that the attribute value a_i of instance x provides *evidence* for the positive class if $\frac{P(a_i|+1)}{P(a_i|-1)} > 1$, and it provides *evidence* for the negative class otherwise.

Let $\mathcal{P}_x$ and $\mathcal{N}_x$ be two sets, such that $\mathcal{P}_x$ contains the attribute values that provide evidence for the positive class and $\mathcal{N}_x$ contains the attribute values that provide evidence for the negative class:

$$\mathcal{P}_x \triangleq \{a_i \mid \frac{P(a_i|+1)}{P(a_i|-1)} > 1\}$$

$$\mathcal{N}_x \triangleq \{a_j \mid \frac{P(a_j|-1)}{P(a_j|+1)} > 1\}$$

Note that these sets are defined around a particular instance x. That is, whether an attribute provides evidence for the the positive or negative class depends on its value, and thus it is dependent on the current x. For example, for medical diagnosis, whether a lab test provides evidence for one class vs. another depends on the outcome of the lab test, and hence is dependent on the patient under consideration.

The total evidence that instance x provides for the positive class is:

$$E^{+1}(x) = \prod_{a_i \in \mathcal{P}_x} \frac{P(a_i|+1)}{P(a_i|-1)} \tag{6}$$

and, the total evidence that instance x provides for the negative class is:

$$E^{-1}(x) = \prod_{a_j \in \mathcal{N}_x} \frac{P(a_j|-1)}{P(a_j|+1)} \tag{7}$$

Note that the ratio of prior probabilities $P(+1)/P(-1)$ also provides evidence for one or the other class. In the above definitions, we focused on the evidence that the attributes values of specific instance x provide. With these definitions, we can rewrite the classification rule for naive Bayes as:

$$y = \begin{cases} +1 & \text{if } \left(\frac{P(+1)}{P(-1)} \frac{E^{+1}(x)}{E^{-1}(x)}\right) > 1 \\ -1 & \text{otherwise} \end{cases} \tag{8}$$

B. Evidence using Logistic Regression

Logistic regression is a discriminative classifier. The parametric model assumed by logistic regression for binary classification is:

$$P(y=-1|x) = \frac{1}{1+e^{(w_0+\sum_{i=1}^{f} w_i a_i)}} \tag{9}$$

$$P(y=+1|x) = \frac{e^{(w_0+\sum_{i=1}^{f} w_i a_i)}}{1+e^{(w_0+\sum_{i=1}^{f} w_i a_i)}} \tag{10}$$

An instance can then be classified using:

$$y = \text{sgn}\left(w_0 + \sum_{i=1}^{f} w_i a_i\right) \tag{11}$$

From Equation 11, it follows that the attribute value a_i of instance x provides *evidence* for the positive class if $w_i a_i > 0$, and it provides *evidence* for the negative class otherwise.

Let $\mathcal{P}_x$ and $\mathcal{N}_x$ be two sets, such that $\mathcal{P}_x$ contains the attribute values that provide evidence for the positive class and $\mathcal{N}_x$ contains the attribute values that provide evidence for the negative class:

$$\mathcal{P}_x \triangleq \{a_i \mid w_i a_i > 0\}$$

$$\mathcal{N}_x \triangleq \{a_j \mid w_j a_j < 0\}$$

Then, the total evidence that instance x provides for the positive class is:

$$E^{+1}(x) = \sum_{a_i \in \mathcal{P}_x} w_i a_i \tag{12}$$

and, the total evidence that instance x provides for the negative class is:

$$E^{-1}(x) = -\sum_{a_j \in \mathcal{N}_x} w_j a_j \tag{13}$$

With these definitions, we can rewrite the classification rule (Eqn. 11) for logistic regression as:

$$y = \text{sgn}\left(w_0 + E^{+1}(x) - E^{-1}(x)\right) \tag{14}$$

C. Evidence using Support Vector Machines

Support Vector Machines (SVM) maximize the margin of classification:

$$w = \arg\max_{w} \left(y \times (w_0 + \sum_{a_i} w_i a_i)\right) \tag{15}$$

and the classification rule is identical to that of logistic regression (Eqn. 11):

$$y = \text{sgn}\left(w_0 + \sum_{i=1}^{f} w_i a_i\right) \tag{16}$$

Following the reasoning of evidence using logistic regression, the equations for $E^{+1}(x)$, $E^{-1}(x)$, and the classification rule for SVM are identical to those for logistic regression.

Next, we briefly outline how the evidence can be generalized to multi-class classification.

D. Evidence for Multi-class Classification

For binary classification, all three types of uncertainties (Equations 1, 2, and 3) prefer instances closest to the decision boundary as specified by Equations 5, 11, and 16. However, their preferences differ in multi-class classification. The entropy approach (Equation 1), for example, considers overall uncertainty and takes into account all classes, whereas the maximum conditional approach (Equation 2) considers how confident the model is about the most likely class. To keep the discussion simple and brief, and as a proof-of-concept, we show how the evidence for multi-class can be extended for naive Bayes (Equation 4) when used with the margin uncertainty approach (Equation 3).

The margin uncertainty prefers instances for which the difference between the probabilities of most-likely class $y^{(m)}$ and next-likely class $y^{(n)}$ is minimum. Let $\mathcal{M}_x$ and $\mathcal{N}_x$ be two sets, such that $\mathcal{M}_x$ contains the attribute values that provide evidence for the most-likely class and $\mathcal{N}_x$ contains the attribute values that provide evidence for the next likely class:

$$\mathcal{M}_x \triangleq \{a_i \mid \frac{P(a_i|y^{(m)})}{P(a_i|y^{(n)})} > 1\}$$

$$\mathcal{N}_x \triangleq \{a_j \mid \frac{P(a_j|y^{(n)})}{P(a_j|y^{(m)})} > 1\}$$

Then, the total evidence that instance x provides for the most-likely class (in comparison to the next-likely class) is:

$$E^m(x) = \prod_{a_i \in \mathcal{M}_x} \frac{P(a_i|y^{(m)})}{P(a_i|y^{(n)})} \tag{17}$$

and, the total evidence that instance x provides for the next-likely class (in comparison to the most-likely class) is:

$$E^n(x) = \prod_{a_j \in \mathcal{N}_x} \frac{P(a_j|y^{(n)})}{P(a_j|y^{(m)})} \tag{18}$$

E. Most-Surely vs. Least-Surely Uncertain

In this paper, we investigate whether the evidence framework provides useful criteria to distinguish between uncertain instances and whether such an approach leads to more or less effective active learning. We have several objectives to optimize at the same time:

- The model needs to be uncertain on instance x.
- For most-surely uncertain, both $E^{+1}(x)$ and $E^{-1}(x)$ need to be large.
- For least-surely uncertain, both $E^{+1}(x)$ and $E^{-1}(x)$ need to be small.

This is a multi-criteria optimization problem where we have to make trade-offs across objectives.

First, we discuss how we define the overall evidence ($E(x)$) as a function of $E^{+1}(x)$ and $E^{-1}(x)$. There are a number of aggregation choices, which include multiplication, summation, taking the minimum and taking the maximum.

In this paper, we focus on the multiplication aggregation:

$$E(x) = E^{+1}(x) \times E^{-1}(x) \tag{19}$$

This aggregation makes sense because the overall evidence $E(x)$ is largest when both $E^{+1}(x)$ and $E^{-1}(x)$ are large and close to each other. Similarly, $E(x)$ is smallest when both $E^{+1}(x)$ and $E^{-1}(x)$ are small. Hence, we mainly focus on the multiplication aggregation in our experiments.

Picking an unlabeled instance x where $E(x)$ is largest (or smallest) will obviously not guarantee that the underlying model is uncertain on x. To guarantee uncertainty, we take a simple approach. We first rank the instances $x \in \mathcal{U}$ in decreasing order of their uncertainty score (measured through one of the Equations 1, 2, or 3) and work with the top k instances, where k is a hyper-parameter. Let $\mathcal{S}$ be the set of top k uncertain instances. *Most-surely uncertain* picks the instance with maximum overall evidence:

$$x^* = \arg\max_{x \in \mathcal{S}} E(x) \tag{20}$$

and, *least-surely uncertain* picks the instance with minimum overall evidence:

$$x^* = \operatorname{argmin}_{x \in \mathcal{S}} E(x) \tag{21}$$

where, $E(x)$ is defined according to Equation 19.

IV. Experimental Methodology

We designed our experiments to test whether distinguishing between most-surely and least-surely uncertain instances makes a difference to the performance of active learning. We experimented with the following approaches:

1) *Random Sampling* (`RND`): This is a common baseline for active learning, in which instances are picked at random from the set of candidate unlabeled instances.
2) *Uncertainty Sampling* - 1^{st} (`UNC-1`): This method picks the instance for which the underlying model is most uncertain, as defined in Section II-A.
3) *Most-Surely Uncertain* (`UNC-MS`): Among the top 10 uncertain instances, this method picks the instance for which the model is most-surely uncertain (as defined in Equation 20) and uses Equation 19 to calculate the overall evidence.
4) *Least-Surely Uncertain* (`UNC-LS`): Among the top 10 uncertain instances, this method picks the instance for which the model is least-surely uncertain (as defined in Equation 21) and uses Equation 19 to calculate the overall evidence.
5) *Uncertainty Sampling* - 10^{th} (`UNC-10`): Among the top 10 uncertain instances, this method picks the 10^{th} most uncertain instance. The motivation behind this method is that `UNC-1` is expected to be better than `UNC-10`, because the top instance is more uncertain than the 10^{th} instance. `UNC-MS` and `UNC-LS` methods pick `UNC-t`, where t is between 1 and 10 and changes

Table I
DESCRIPTION OF THE DATASETS: THE DOMAIN, NUMBER OF INSTANCES IN THE DATASET AND THE PERCENTAGE OF MINORITY CLASS. THE DATASETS ARE SORTED IN INCREASING ORDER OF CLASS IMBALANCE.

Dataset	Domain	Size	Min. %
Spambase	Email. classif.	4,601	39.4%
Ibn Sina	Handwr. recog.	20,722	37.8%
Calif. Housing	Social	20,640	29%
Nova	Text processing	19466	28.4%
Sick	Medical	3,772	6.1%
Zebra	Embryology	61,488	4.6%
LetterO	Letter recog.	20,000	4%
Hiva	Chemo-inform.	42,678	3.5%

at every iteration. If UNC−MS and/or UNC−LS are better than UNC−1, then this result would suggest that different types of uncertainties matter. Similarly, if UNC−MS and/or UNC−LS are worse than UNC−10, then this result would also suggest that different types of uncertainties matter.

We experimented with eight publicly available datasets. Active learning methods can behave very differently under varying class imbalance. Thus, we chose a mix of datasets with various class imbalances. We chose four datasets with minority class $\% > 10\%$, which we refer to as *medium-imbalanced*, and four datasets with minority class $\% \leq 10\%$, which we refer to as *extreme-imbalanced* datasets. We provide the description of these datasets in Table I. We evaluated these five methods using three performance measures: AUC, accuracy, and F1. We computed AUC for all the datasets. We computed accuracy for only medium-imbalanced datasets (the top four in Table I) and F1 for only extreme-imbalanced datasets (bottom four in Table I).

A. Parameters and Repeatability

We performed five-fold cross validation and repeated the experiments five times per fold. In each experiment, the train split was treated as the unlabeled set, $\mathcal{U}$, and randomly chosen 10 instances (five from each class) were used as the initially labeled set, $\mathcal{L}$. At each iteration, each method picks only one instance to be labeled. We set our budget, B, in Algorithm 1 to 500 instances. UNC−MS and UNC−LS operate within top k uncertain instances, as described in Section III-E. We set $k = 10$. We evaluated each method using a naive Bayes classifier with Laplace smoothing. To speed-up the experiments, at each iteration we computed utility over a set of randomly sub-sampled 250 instances, which is a common practice in active learning. We used entropy as a measure of uncertainty (Equation 1), but using the other two measures of uncertainty would lead to identical results because we experimented with binary classification tasks.

B. Scalability

We discuss the comparison of running times of UNC−1, UNC−MS, and UNC−LS methods for naive Bayes for one iteration of active learning. Given dataset $\mathcal{D} = \left\{x^{(i)}, y^{(i)}\right\}_1^m$ where, $x^{(i)} \in \mathbb{R}^f$, and $y^{(i)} \in \{+1, -1\}$ is discrete valued. UNC−1 calculates uncertainty score (measured through one of the Equations 1, 2, or 3). The time complexity of calculating the conditional probabilities $P_\theta(y|x)$ in each of these equations is proportional to the number of attributes, which is O(f). Since we compute uncertainty on m instances, the time complexity of UNC−1 is $O(m * f)$.

UNC−MS and UNC−LS methods also calculate uncertainty on m instances, which takes time $O(m * f)$. Additionally, UNC−MS and UNC−LS methods calculate evidence for each attribute of an instance, which again takes time O(f). This additional step is done only for the top k (where, k=10 for our experiments) uncertain instances. Hence, the running time of UNC−MS and UNC−LS methods is $O((m + k) * f)$. Given that k is a small constant ($k << m$), the running times of UNC−MS and UNC−LS are comparable to the running time of UNC−1.

V. RESULTS AND DISCUSSION

In this section, we present the results for five methods (UNC−MS, UNC−LS, UNC−1, UNC−10 and RND). We present the AUC results in Figure 2 and Figure 3, accuracy results in Figure 4, and F1 results in Figure 5.

As Figures 2, 3, 4 and 5 show, distinguishing between most-surely and least-surely uncertain instances has a huge impact on active learning for all datasets and measures. This result is quite interesting because UNC−MS, UNC−LS, UNC−1 and UNC−10 all rank the instances according to uncertainty and operate within the same top 10 instances; thus their flexibility in choosing a different instance for labeling is rather limited and yet they result in drastically different performances.

Next, we provide the results of statistical significance tests comparing these five methods. Tables II and III provide summary of pairwise one-tailed t-tests results under significance level of 0.05, where the pairs are the learning curves of the compared methods. 'W/L' means that the method significantly wins/loses to the baseline, and 'T' means that there is no significant difference between the method and the baseline. Note that for each method, the total counts of 'W', 'T' and 'L' should add up to 8 for AUC, 4 for accuracy and 4 for F1.

Table II presents summary of 'Win/Tie/Loss' counts of UNC−MS and UNC−LS compared to UNC−1 baseline. With respect to UNC−1, there is a clear difference between UNC−MS and UNC−LS. Our results show that on AUC, UNC−MS wins over UNC−1 on all 8 datasets, whereas UNC−LS loses to UNC−1 on 7 out of 8 datasets. On accuracy, UNC−MS wins over UNC−1 on 3 out of 4 datasets and ties on one dataset (Nova), whereas UNC−LS loses to UNC−1 on

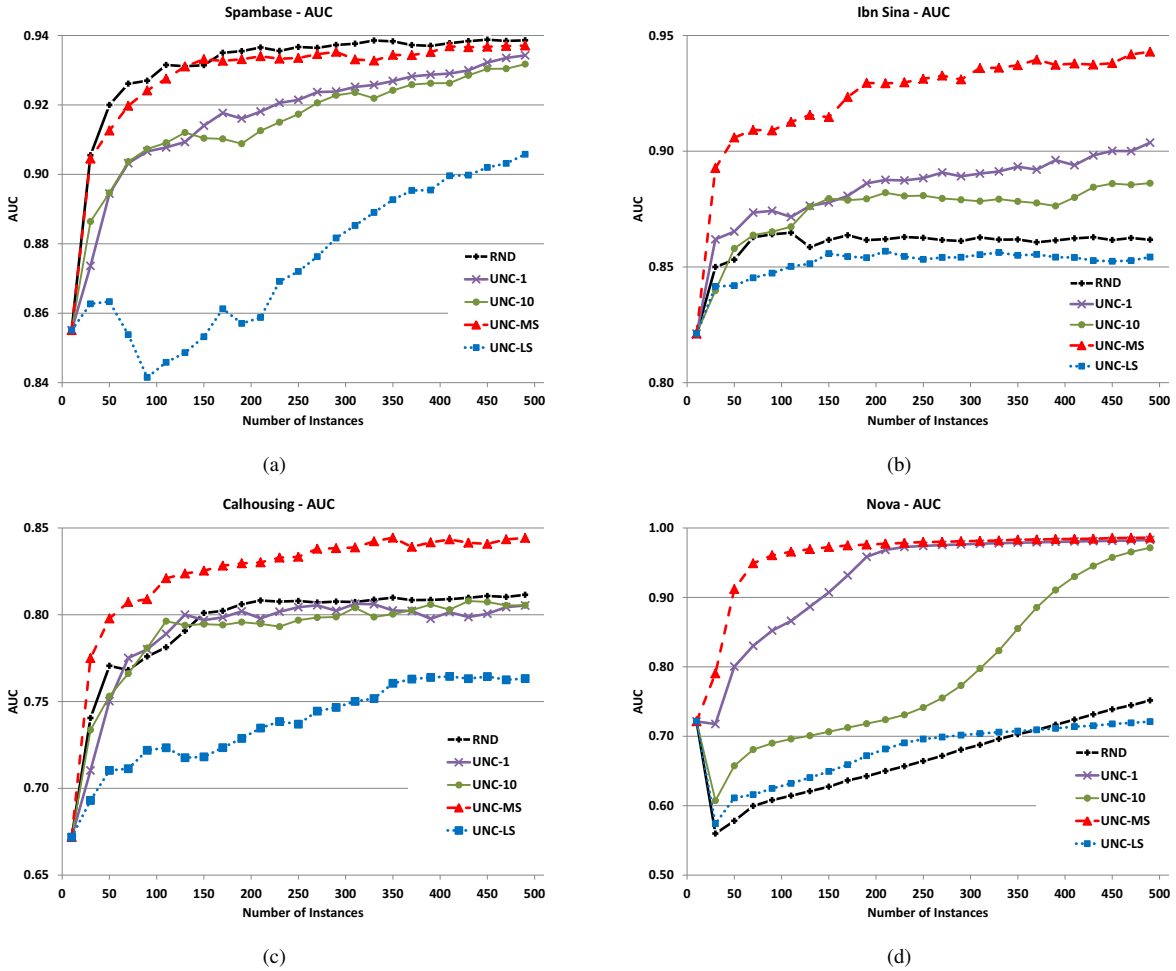

Figure 2. AUC results for four medium-imbalanced datasets (Spambase, Ibn Sina, Calif. Housing and Nova). UNC−MS outperforms UNC−1 on all four datasets. UNC−LS loses to UNC−1 on all four datasets.

all 4 datasets. On F1, UNC−MS wins over UNC−1 on 3 out of 4 datasets and loses on one dataset (LetterO), whereas UNC−LS loses to UNC−1 on all 4 datasets.

UNC−MS clearly stands out as a winner strategy, significantly outperforming UNC−1 on almost all datasets and measures. On the other hand, UNC−LS is clearly the worst performing uncertainty strategy, losing to UNC−1 on almost all datasets and measures. This clear distinction between UNC−MS and UNC−LS holds for both medium-imbalanced and extreme-imbalanced datasets.

Next, we compared UNC−MS and UNC−LS to UNC−10. Table III presents 'Win/Tie/Loss' results using UNC−10 as the baseline. We observe that UNC−MS significantly outperforms UNC−10 on almost all datasets, which is not surprising because even a random strategy from top 10 uncertain instances has the potential to outperform UNC−10. However, it is surprising to observe that UNC−LS is performing statistically significantly worse than UNC−10 for almost all datasets and measures.

Table II
UNC-MS AND UNC-LS VERSUS UNC-1

UNC−1 **baseline**	**AUC**	**ACCU**	**F1**
Method	**W/T/L**	**W/T/L**	**W/T/L**
UNC−MS	**8**/0/0	**3**/1/0	**3**/0/1
UNC−LS	1/0/**7**	0/0/**4**	0/0/**4**

Table III
UNC-MS AND UNC-LS VERSUS UNC-10

UNC−10 **baseline**	**AUC**	**ACCU**	**F1**
Method	**W/T/L**	**W/T/L**	**W/T/L**
UNC−MS	**7**/0/1	**4**/0/0	**3**/0/1
UNC−LS	1/0/**7**	0/0/**4**	0/0/**4**

These results clearly suggest that the two types of uncertainties have an effect on active learning. *Most-surely uncertain* can help to improve active learning, whereas *least-surely uncertain* hurts active learning.

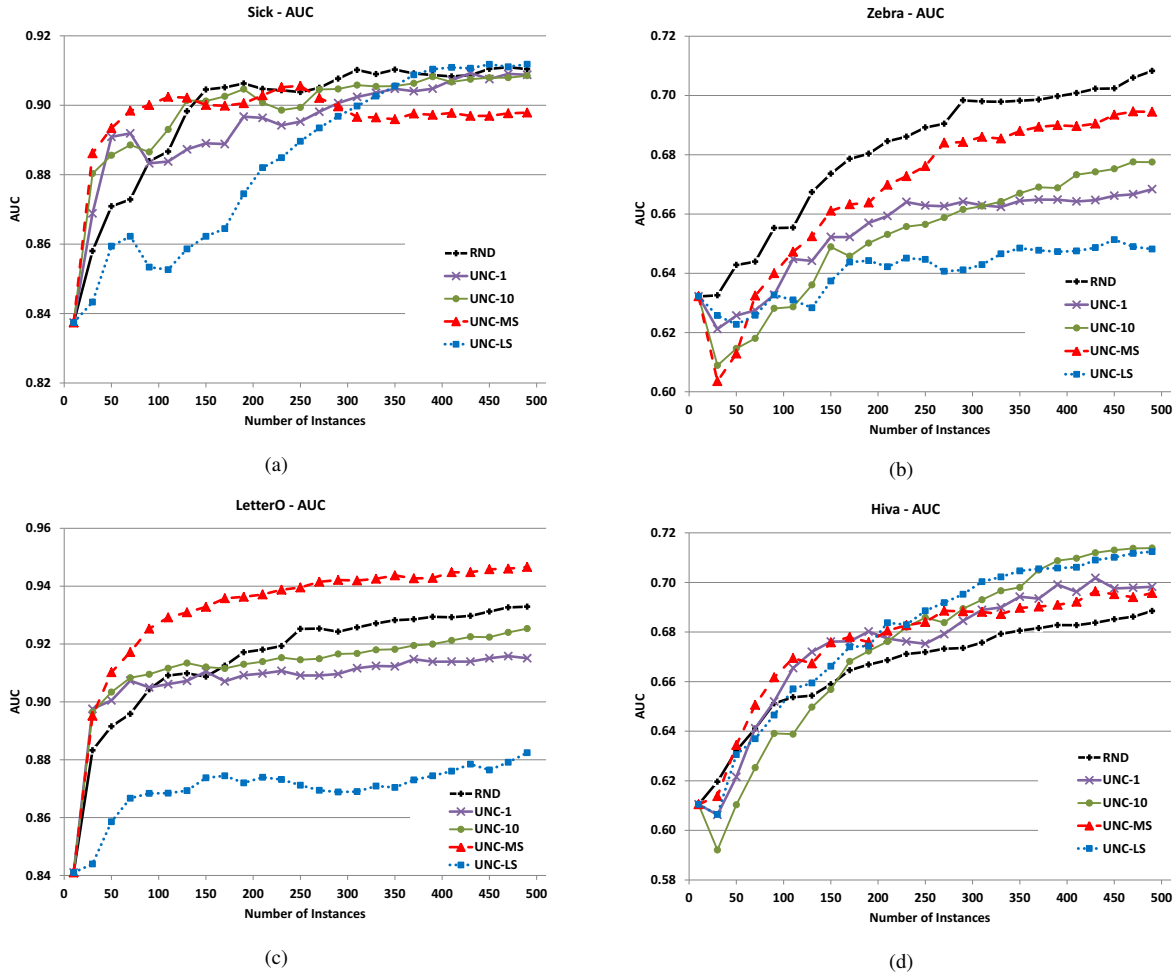

Figure 3. AUC results for four relatively skewed datasets (Sick, Zebra, LetterO and Hiva). UNC−MS outperforms UNC−1 on all four datasets. UNC−LS loses to UNC−1 on all four datasets except Hiva (d).

A. *Effect of Evidence Aggregation*

In this section, we investigate whether other aggregation functions can be used to compute the overall evidence, and how sensitive our experiments are to the choice of aggregation. For *most-surely uncertain*, we also investigate choosing instances where the smaller of the $E^{+1}(x)$ and $E^{-1}(x)$ is large, and hence experiment with choosing the instance where the minimum is largest:

$$x^* = \arg\max_{x \in \mathcal{S}} E(x) = \arg\max_{x \in \mathcal{S}} min(E^{+1}(x), E^{-1}(x)) \tag{22}$$

We refer to this strategy as UNC−MS−MIN. For *least-surely uncertain*, we also investigate choosing instances where the larger of the $E^{+1}(x)$ and $E^{-1}(x)$ is small, and hence experiment with choosing the instance where the maximum is smallest:

$$x^* = \operatorname{argmin}_{x \in \mathcal{S}} E(x) = \operatorname{argmin}_{x \in \mathcal{S}} max(E^{+1}(x), E^{-1}(x)) \tag{23}$$

We refer to this strategy as UNC−LS−MAX.

Tables IV and V provide summary of pairwise one-tailed t-tests results for these methods compared to UNC−1 and UNC−10 baselines. The results in Tables IV and V have same setting as Tables II and III (in Section V).

We observe that UNC−MS−MIN significantly outperforms both UNC−1 and UNC−10 on almost all datasets and all measures, whereas UNC−LS−MAX significantly loses to both UNC−1 and UNC−10 on almost all datasets and all measures. These results are quite similar to the results for UNC−MS and UNC−LS methods. Thus, our claims that *most-surely uncertain* significantly improves regular uncertainty sampling and that *least-surely uncertain* significantly hurts regular uncertainty sampling, hold regardless of the aggregation function used.

VI. Related Work

Many active learning methods have been developed in the past [1]. Uncertainty sampling [2] is arguably one of the

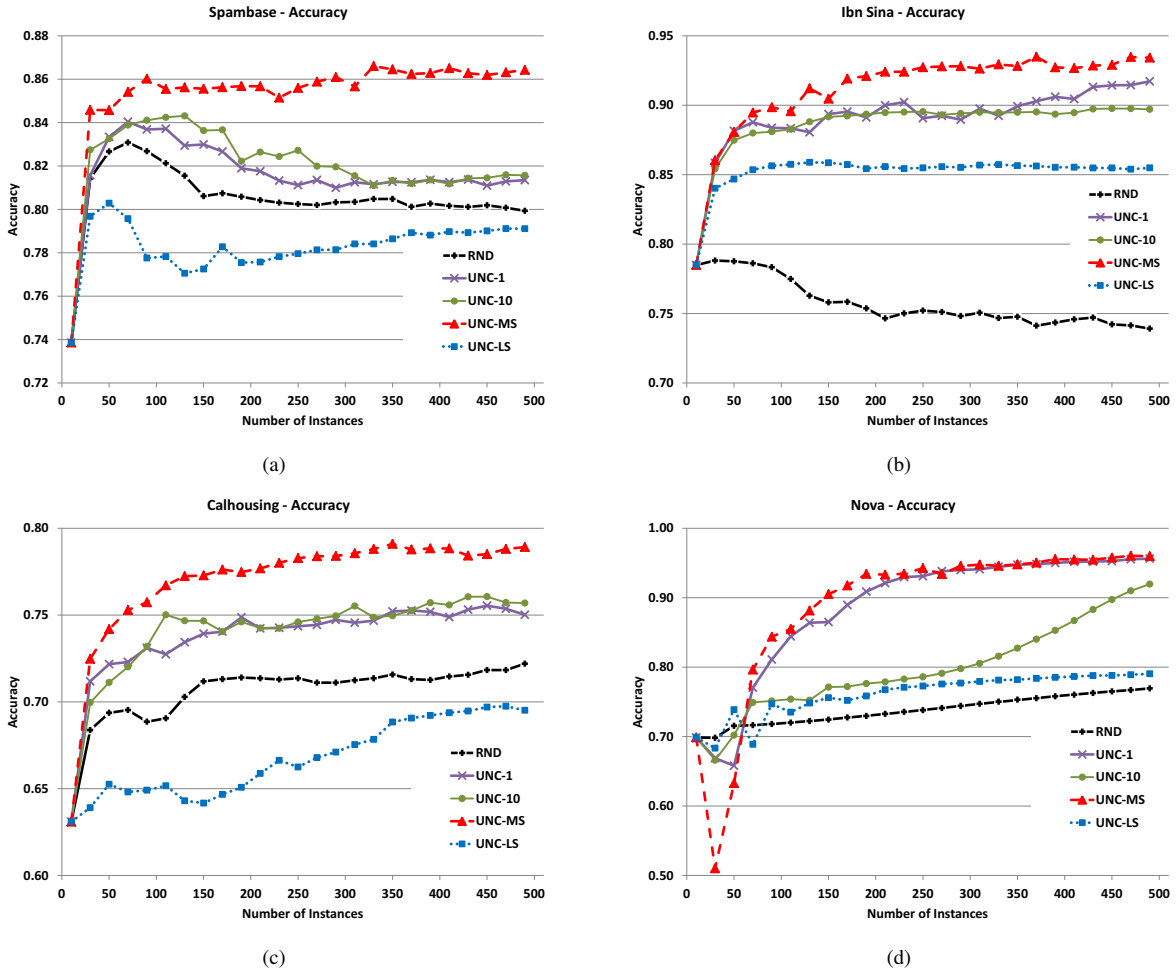

Figure 4. Accuracy results. UNC-MS outperforms UNC-1 significantly on three datasets and ties on Nova (d). UNC-LS loses to UNC-1 on all datasets.

Table IV
UNC-MS-MIN AND UNC-LS-MAX VERSUS UNC-1

UNC-1 **baseline**	**AUC**	**ACCU**	**F1**
Method	**W/T/L**	**W/T/L**	**W/T/L**
UNC-MS-MIN	**7**/0/1	**4**/0/0	**3**/0/1
UNC-LS-MAX	1/0/**7**	0/0/**4**	0/0/**4**

Table V
UNC-MS-MIN AND UNC-LS-MAX VERSUS UNC-10

UNC-10 **baseline**	**AUC**	**ACCU**	**F1**
Method	**W/T/L**	**W/T/L**	**W/T/L**
UNC-MS-MIN	**7**/0/1	**4**/0/0	**3**/0/1
UNC-LS-MAX	1/0/**7**	1/0/**3**	0/0/**4**

most common active learning methods and is frequently used as a baseline for comparing other active learning methods.

Uncertainty sampling has been shown to work successfully in a variety of domains. Example domains include text classification [2, 8, 9, 10], natural language processing [11], email spam filtering [12, 13], image retrieval [14], medical image classification [15], robotics [16], information retrieval [17], dual supervision [18] and sequence labeling [19] among many others.

Even though uncertainty sampling is frequently utilized, it is known to be susceptible to noise and outliers [7]. A number of approaches have been proposed to make it more robust. For example, [19] weights the uncertainty of an instance by its density to avoid outliers, where density of the instance is defined as average similarity to other instances. [20] used a K-Nearest-Neighbor-based density measure to determine whether an unlabeled instance is an outlier. [9] proposed a hybrid approach to combine representative sampling and uncertainty sampling. Other approaches used the cluster structure of the domain to choose more representative examples [21, 8].

Our work is orthogonal to these approaches. We are not providing yet another alternative approach to improve uncertainty sampling, but instead we are highlighting that

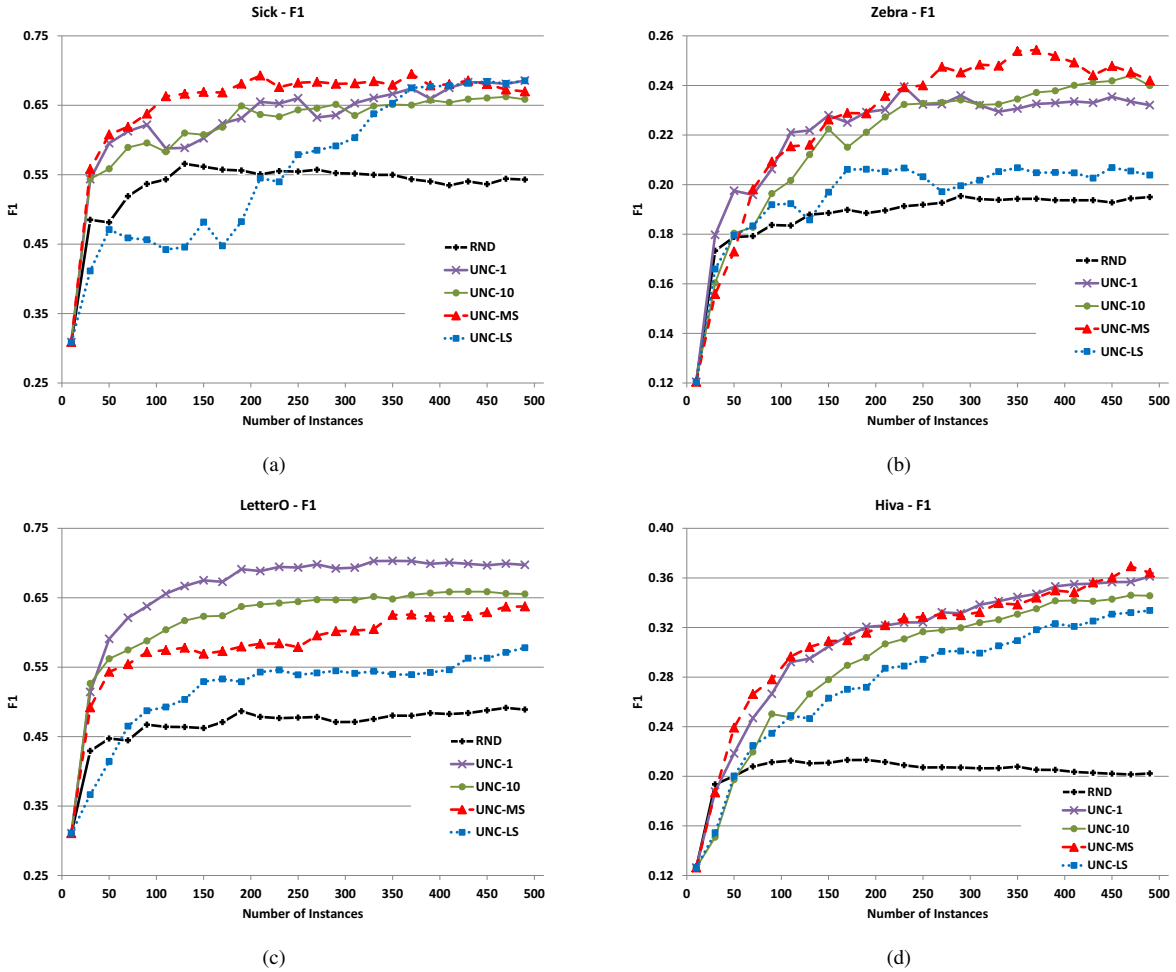

Figure 5. F1 results. UNC−MS outperforms UNC−1 on three datasets ((a), (b) and (d)), and loses on one (c). UNC−LS loses to UNC−1 on all datasets.

distinguishing between the two types of uncertainties (most-sure vs. least-sure) has a big impact on active learning. One can imagine combining uncertainty sampling, density weighting and *most-surely uncertain* methods because they are not mutually exclusive.

VII. Limitations and Future Work

We combined the evidence framework and uncertainty sampling using a simple approach in Section III-E: we first ranked the instances according to uncertainty and then applied the evidence framework to the top k instances. We observed that this simple approach worked very well. In the future, we would like to investigate multi-criteria optimization approaches [22] for combining uncertainty sampling and the evidence framework. A potential approach is to utilize the Knapsack framework. In Knapsack problem, we are given a set of instances in which each instance has a value and a cost, and one needs to pick instances so as to maximize the total value while not exceeding a given budget. For our problem, we can define the value of an instance as its overall evidence and its cost as model's certainty on the instance. The objective is to pick k instances so as to maximize the total evidence, while remaining within a budget of total certainty.

It is well-known in the active learning community that uncertainty sampling is susceptible to noise and outliers [19]. It is not clear at this point whether combining uncertainty sampling with the evidence framework makes it more or less susceptible to noise and outliers. We experimented with real-world datasets, which are expected to be noisy, and showed that most-surely uncertain significantly outperforms uncertainty sampling while least-surely uncertain performed significantly worse on many measures and datasets. The effect of noise and outliers on UNC−MS and UNC−LS needs to be verified through carefully designed experiments with synthetic datasets.

A related question is whether the most-surely uncertain method makes the experts' job easier or more difficult. That is, are the instances chosen by UNC−MS harder or easier to

label than the ones chosen by UNC-LS? Similarly, does it take longer or shorter for experts to label UNC-MS instances versus UNC-LS instances? We note that we define most-sure/least-sure uncertainty with respect to the underlying model, which is trained on a relatively small training set, and not with respect to the expert. Thus, it is hard to estimate the real affect of the evidence framework on annotation time and difficulty. These issues need to be investigated through user and case studies.

VIII. Conclusion

We introduced a framework that distinguishes between two types of uncertainties: a model is uncertain about an instance due to strong and conflicting evidence (most-surely uncertain) vs. a model is uncertain because it does not have conclusive evidence (least-surely uncertain). The regular uncertainty sampling does not distinguish between these types of uncertainties, but our empirical evaluations showed that making this distinction had a big impact on the performance of uncertainty sampling. While least-surely uncertain instances provided the least value to an active learner, actively labeling most-surely uncertain instances performed significantly better than regular uncertainty sampling.

References

[1] B. Settles, *Active Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[2] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 3–12.

[3] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *ACM Annual Workshop on Computational Learning Theory*, 1992, pp. 287–294.

[4] D. A. Cohn, "Minimizing statistical bias with queries," in *Advances in Neural Information Processing Systems*, 1997, pp. 417–423.

[5] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[6] M. Lindenbaum, S. Markovitch, and D. Rusakov, "Selective sampling for nearest neighbor classifiers," *Machine Learning*, vol. 54, no. 2, pp. 125–152, 2004.

[7] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *International Conference on Machine Learning*, 2001, pp. 441–448.

[8] M. Bilgic, L. Mihalkova, and L. Getoor, "Active learning for networked data," in *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[9] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Representative sampling for text classification using support vector machines," *Advances in Information Retrieval*, pp. 11–11, 2003.

[10] S. C. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 633–642.

[11] C. A. Thompson, M. E. Califf, and R. J. Mooney, "Active learning for natural language parsing and information extraction," in *Proceedings of International Conference on Machine Learning*, 1999, pp. 406–414.

[12] D. Sculley, "Online active learning methods for fast label-efficient spam filtering," in *Conference on Email and Anti-Spam (CEAS)*, 2007.

[13] R. Segal, T. Markowitz, and W. Arnold, "Fast uncertainty sampling for labeling large e-mail corpora," in *Conference on Email and Anti-Spam*, 2006.

[14] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001, pp. 107–118.

[15] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proceedings of the International Conference on Machine learning*, 2006, pp. 417–424.

[16] C. Chao, M. Cakmak, and A. L. Thomaz, "Transparent active learning for robots," in *5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010, pp. 317–324.

[17] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 260–268, 2002.

[18] V. Sindhwani, P. Melville, and R. D. Lawrence, "Uncertainty sampling and transductive experimental design for active dual supervision," in *Proceedings of the International Conference on Machine Learning*, 2009, pp. 953–960.

[19] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 1070–1079.

[20] J. Zhu, H. Wang, T. Yao, and B. K. Tsou, "Active learning with sampling by uncertainty and density for word sense disambiguation and text classification," in *Proceedings of the International Conference on Computational Linguistics - Volume 1*, 2008, pp. 1137–1144.

[21] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *International Conference on Machine Learning*, 2004.

[22] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computations, and Application*. Krieger Pub Co, 1989.