Data Min Knowl Disc (2017) 31:287–313
DOI 10.1007/s10618-016-0469-7

Active learning: an empirical study of common baselines

Maria E. Ramirez-Loaiza[1] · Manali Sharma[1] · Geet Kumar[1] · Mustafa Bilgic[1]

Received: 16 January 2015 / Accepted: 25 May 2016 / Published online: 6 June 2016

Abstract Most of the empirical evaluations of active learning approaches in the literature have focused on a single classifier and a single performance measure. We present an extensive empirical evaluation of common active learning baselines using two probabilistic classifiers and several performance measures on a number of large datasets. In addition to providing important practical advice, our findings highlight the importance of overlooked choices in active learning experiments in the literature. For example, one of our findings shows that model selection is as important as devising an active learning approach, and choosing one classifier and one performance measure can often lead to unexpected and unwarranted conclusions. Active learning should generally improve the model's capability to distinguish between instances of different classes, but our findings show that the improvements provided by active learning for one performance measure often came at the expense of another measure. We present several such results, raise questions, guide users and researchers to better alternatives, caution against unforeseen side effects of active learning, and suggest future research directions.

Responsible editor: Johannes Fuernkranz.

✉ Mustafa Bilgic
mbilgic@iit.edu

Maria E. Ramirez-Loaiza
mramire8@hawk.iit.edu

Manali Sharma
msharm11@hawk.iit.edu

Geet Kumar
gkumar7@hawk.iit.edu

[1] Illinois Institute of Technology, 10 W 31st Street, Chicago, IL 60616, USA

Keywords Active learning · Query by committee · Uncertainty sampling · Empirical evaluation

1 Introduction

Active learning (Cohn et al. 1994) is the subfield of machine learning that makes algorithms active participants in data annotation with the objective to learn the target function more economically (Settles 2012). By carefully choosing which instances should be annotated, active learning algorithms can reduce the time, effort, and resources needed to train an accurate predictive model.

Many successful active learning methods have been developed in the past two decades; they are often shown to outperform random sampling and common baselines. However, the majority of active learning experiments in the literature focused on a single classifier and a single performance measure. To gain some insight on how common this is, we inspected all the active learning papers in the references section of (Settles 2012) and analyzed which classifiers and performance measures were used in each of the papers. We analyzed 54 papers and found that 45 of them (83 %) used a single classifier, often without justifying their choice of the classifier (Table 1). The most common classifier was log-linear models (e.g., logistic regression). Forty-nine out of the 54 papers (91 %) focused on a single performance measure and the most common measure was accuracy. As we present later in the article, unless the choices of the classifier and performance measure are justified or advocated by the domain, the conclusions drawn in empirical studies might be unwarranted or even misleading.

We present an extensive empirical evaluation of common active learning baselines, comparing them across classifiers and performance measures. We evaluate most-frequently utilized baselines (e.g., random sampling, uncertainty sampling, and query-by-committee) using two commonly utilized probabilistic classifiers, naïve Bayes and logistic regression, and evaluate them through common measures, such as accuracy, precision, recall, area under the receiver operating characteristic curve (AUC), and F_1 score. These performance measures are described in Table 4.

The empirical results in this article shed light on how various active learning strategies behave when compared across classifiers and performance measures. In addition to providing important practical advice, our results highlight the potential negative side effects of active learning and warn against possibly unwarranted conclusions in the current active learning literature. We briefly mention one of the results, deferring the other results and details associated with them to Sect. 4.

In our empirical evaluations on 10 datasets, we found that uncertainty sampling and query-by-committee outperformed random sampling on most datasets when the performance measure was accuracy. However, when the performance measure was AUC, random sampling was fairly competitive, winning over active learning strategies on at least half of the datasets. This is an interesting result because 33 out of the 54 papers that we inspected (61 %) used accuracy as their performance criteria, while only 6 papers (11 %) used AUC (Table 1). This finding, in fact, raises serious concerns about the overall effectiveness of active learning strategies. Note that the accuracy measure requires a threshold for assigning a class whereas the AUC measure does not.

An important question raised by this result is: how much of the observed improvement in accuracy is due to effective learning and how much of it is due to simply a shift in the decision threshold caused by biased sampling? In this article, we present several such results, raise similar questions, guide users and researchers to better alternatives, and caution against unforeseen side effects of active learning.

There have been a few extensive empirical evaluations of active learning (e.g., Schein and Ungar 2007; Settles and Craven 2008). However, these studies concentrated on a single classifier and a single measure. Our main contribution in this article is the empirical evaluation of active learning baselines across classifiers and performance measures. Our most important findings stem from this difference. In this article, we are able to draw valuable conclusions and provide guidelines regarding a number of issues including model selection, domain knowledge integration, and trade-offs across performance measures.

The rest of the article is organized as follows: we provide background information on active learning in Sect. 2. We then discuss our experimental methodology in Sect. 3 and present results in Sect. 4. We discuss related work in Sect. 5, present limitations and future work in Sect. 6 and then conclude in Sect. 7. Finally, we introduce an open source active learning Python library, `PyAL`, in Sect. 1 of the Appendix.

2 Background

In pool-based active learning, it is assumed that we are given a set of labeled instances $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in Y$ is its class label, and a large set of unlabeled instances $\mathcal{U} = \{\mathbf{x}_i\}_{i=l+1}^{m}$ whose labels are unknown. Assuming a pre-specified annotation budget B and an annotation cost function $Cost(\mathbf{x})$, the goal of the active learning algorithm (*learner*) is to select $\mathcal{U}^* \subseteq \mathcal{U}$ to be labeled by a human annotator (*oracle*) to expand $\mathcal{L}$ and maximize the classifier's generalization performance subject to the budget constraints:

$$\mathcal{U}^* \leftarrow \underset{\mathcal{U}_i \subseteq \mathcal{U}}{\operatorname{argmax}} \, Performance\left(P_{\mathcal{L} \cup \mathcal{U}_i}(y|\mathbf{x})\right) \text{ s.t. } \sum_{\mathbf{x}_j \in \mathcal{U}_i} Cost(\mathbf{x}_j) \leq B \qquad (1)$$

where $Performance(\cdot)$ is a pre-specified measure of classifier performance (such as accuracy) and $P_{\mathcal{L}}(y|\mathbf{x})$ represents the conditional probability distribution learned by the underlying classifier using the labeled set $\mathcal{L}$. Equation 1 is typically optimized by greedy algorithms, selecting one or more examples at a time according to some heuristic criterion that estimates the *utility* of each labeled example.

Various definitions of *utility* are used in the literature, such as classifier uncertainty (Lewis and Gale 1994), committee disagreement (Seung et al. 1992), expected reduction in error (Roy and McCallum 2001), etc. Uncertainty sampling and query-by-committee are perhaps the two most-frequently utilized baselines in the literature. Lewis and Gale (1994) received 1, 513 citations and Seung et al. (1992) received 974 citations according to Google Scholar, as checked on April 23, 2016. In this article, we investigate how these two approaches behave when they are compared across classifiers and performance measures. Next, we explain these two approaches in detail.

2.1 Uncertainty sampling (UNC)

Uncertainty sampling (UNC) queries the label of the instance for which the current model is most uncertain (Lewis and Gale 1994). For margin-based classifiers, such as support vector machines, uncertainty is defined in terms of the distance to the margin (Tong and Koller 2001). For probabilistic classifiers, entropy and conditional error are common choices. Equation 2 defines the objective of uncertainty sampling based on conditional error:

$$\mathbf{x}^*_{\text{UNC}} = \underset{\mathbf{x}\in\mathcal{U}}{\text{argmax}} \left(1 - \max_{y\in Y} P_{\mathcal{L}}(y|\mathbf{x})\right) \tag{2}$$

Uncertainty sampling is easy to implement, fairly intuitive, and it often improves over random sampling. Even though UNC is known to be susceptible to noise and outliers (e.g., Roy and McCallum 2001; Settles and Craven 2008), it works surprisingly well and has been successfully used in several papers and domains. Examples include (Bilgic et al. 2010; Xu et al. 2003; Hoi et al. 2006a; Thompson et al. 1999; Sculley 2007; Segal et al. 2006; Tong and Chang 2001; Hoi et al. 2006b; Chao et al. 2010; Zhang and Chen 2002; Sindhwani et al. 2009; Settles and Craven 2008; Sharma et al. 2015; Ramirez-Loaiza et al. 2014; Sharma and Bilgic 2013), among many others.

2.2 Query by committee (QBC)

Query by committee (QBC) queries the label of the instance for which a committee of classifiers disagree the most (Seung et al. 1992). The committee is formed by sampling hypotheses from the version space. Under certain conditions, QBC is shown to reduce the prediction error exponentially fast in the number of queries to the oracle (Seung et al. 1992; Freund et al. 1997).

Sampling from the version space, however, is not tractable for most classifiers. Thus, Abe and Mamitsuka (1998) proposed an approximate version of QBC that can generalize to any classifier. In this approximate approach, the committee members are formed through bagging on $\mathcal{L}$ (Breiman 1996). Two common approaches to measuring the disagreement between committee members are margin of disagreement, i.e. the difference between number of votes for the most popular label and number of votes for the next most popular label (Melville and Mooney 2004), and vote entropy (Dagan and Engelson 1995). Vote entropy is defined as follows:

$$\mathbf{x}^*_{\text{QBC}} = \underset{\mathbf{x}\in\mathcal{U}}{\text{argmax}} - \sum_{y\in Y} \frac{V(y)}{C} \log \frac{V(y)}{C} \tag{3}$$

where y ranges over all possible labels in Y, $V(y)$ is the number of votes that a label receives from the committee members and C is the committee size. When the target class y is binary, both margin and vote entropy approaches rank instances in the same order.

2.3 Random sampling (RND)

Random sampling is the most common baseline that is used for comparing against active learning strategies. The RND strategy selects instances randomly from the unlabeled pool, without paying any attention to whether that instance provides any additional information to the classifier. Though it is simple and does not take any classifier-specific utility into account, it implicitly selects representative examples that are *i.i.d.*, and hence it often serves as a very strong baseline that is hard to beat.

In this article, we compare random sampling, uncertainty sampling, and query-by-committee on several synthetic and real-world datasets, using naïve Bayes and logistic regression as the underlying classifiers, and using accuracy, AUC, precision, recall, and F_1 as the performance measures for comparison.

3 Experimental methodology

Our empirical evaluation focused on the following questions and we investigate the answers to each of these questions in Sect. 4.

1. How does active learning (AL) perform in comparison to RND?
2. How does UNC perform in comparison to QBC?
3. How does the choice of underlying classifier affect the performances of UNC and QBC?
4. How does the choice of performance measure affect the performances of UNC and QBC?
5. If the data is continuously collected through AL, how does AL behave in the long term?
6. How does the size of initially labeled set affect the performance of AL?

We performed a literature study, analyzing all the papers (a total of 158) in the reference section of Settles (2012). We identified 54 active learning papers that had experimental results. For these 54 papers, we looked at how many papers used a given classifier and performance measure. We also checked if and how many classifiers and performance measures these 54 papers used in their experiments. Table 1 shows the results of this analysis.

As these results show, a majority (83 %) of the papers used a single classifier and a majority (91 %) of the papers used a single performance measure. We observed that a number of papers indeed focused on text evaluation. We note, however, that this study is not necessarily a representative of all the papers published in the data mining / machine learning / artificial intelligence literature. Therefore, Table 1 does not claim to be a definitive representation of the literature; rather, its purpose is to give some idea as to which classifiers and measures were common. As mentioned in Sect. 2, we chose to study two of the most common active learning approaches that are typically used as baselines and from which many specialized methods have been derived.

Next, we describe the datasets, classifiers, and performance measures used to evaluate RND, UNC, and QBC. In the remainder of this article, when we use the term active learning (AL), we mean both UNC and QBC.

Table 1 Literature study of 54 papers, showing how many (#) and what percentage of the papers (%) used which classifiers and performance measures

The number of classifiers and the number of measures add up to more than 54 because some papers used more than one classifier and some papers used more than one performance measure

Classifier	#	%	Measure	#	%
Log-linear models	28	52	Accuracy	33	61
SVM	17	31	F_1	14	26
Decision Trees	6	11	AUC	6	11
Naïve Bayes	4	7	Precision	3	6
Neural Networks	2	4	Recall	3	6
Others	7	13	Others	3	6
Single classifier	45	83	Single measure	49	91
Two classifiers	8	15	Two measures	2	4
Three classifiers	1	2	Three measures	3	6

Table 2 Description of the real-world datasets: the domain, number of instances, number of features, types of features (numeric/binary/categorical), and percentage of the minority class

Dataset	Domain	# of instances	# of features	Types of features	Min. %
Calif. Housing	Social	20,640	8	Numeric	29
Hiva	Chemo-inform.	42,678	1,617	Binary	3.5
Ibn Sina	Handwr. recog.	20,722	92	Numeric	37.8
KDD99	Network	494,020	41	Numeric + Categorical	16
LetterO	Letter recog.	20,000	16	Numeric	4
LetterAM	Letter recog.	20,000	16	Numeric	8
Nova	Text processing	19,466	16,969	Binary	28.5
Orange	Marketing	50,000	230	Numeric	1.6
Sylva	Ecology	145,252	216	Numeric + Binary	6.2
Zebra	Embryology	61,488	154	Numeric	4.6

3.1 Datasets

We experimented with both synthetic and real-world datasets. We generated several binary synthetic datasets with positive class distributions of 50, 25, 10, and 1 %. For each class distribution case, we generated five datasets (a pair of train and test split, each having 10K and 1K instances respectively), resulting in 20 synthetic datasets. The synthetic datasets were generated using a naïve Bayes model with 1000 binary features.

We used 10 large real-world binary classification datasets. The smallest dataset had 19 K instances and the largest had more than 490K instances. We used all six active learning challenge datasets (Guyon et al. 2011) and four additional large datasets to complement the study (Frey and Slate 1991; Pace and Barry 1997; Bay et al. 2000). The domains and class distributions of these datasets are diverse (Table 2).

3.2 Evaluation methodology

For synthetic datasets, because we were able to generate as much data as we wanted, we generated train-test splits, and hence performed a train-test evaluation. For real-world datasets, we performed five-fold cross validation. For each experiment, the train split was treated as the unlabeled pool, $\mathcal{U}$; randomly chosen 10 instances (five from each class) were used as the initially labeled set (i.e., the bootstrap), $\mathcal{L}$. At the beginning of each trial for an experiment, if the unlabeled pool, $\mathcal{U}$, consisted of more than 10K instances, we randomly subsampled 10K instances from $\mathcal{U}$, which is a common practice in active learning. At each iteration, we picked the top 10 utility instances, as determined by the `AL` strategy. We repeated each experiment 10 times for each train-test split for synthetic datasets and five times per fold, and hence $5 \times 5 = 25$ times per dataset, for the real-world datasets.

To seek an answer to question 3, we experimented with a generative model (naïve Bayes) and a discriminative model (logistic regression). A naïve Bayes classifier uses the Bayes rule to compute $P(y|\mathbf{x})$ and assumes that features, f_i, are conditionally independent given class, y:

$$P(y|\mathbf{x}) = P(y|f_1, f_2, \cdots, f_d) = \frac{P(y)\prod_{f_i} P(f_i|y)}{P(f_1, f_2, \cdots, f_d)} \tag{4}$$

where, $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional feature vector, $\mathbf{x} = \langle f_1, f_2, \cdots, f_d \rangle$.

For naïve Bayes, $P(y)$ is modeled as a Bernoulli or multinomial distribution depending on whether it is a binary or multi-class classification problem. How $P(fi|y)$ is modeled depends on the type of the feature. Typically, continuous variables are modeled as Gaussian distributions, binary features are modeled as Bernoulli distributions, and categorical features are modeled as multivariate multinomial distributions.

The specific implementation of logistic regression we used in this article optimizes the following objective function:

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) \tag{5}$$

where, $\mathbf{w}$ is the weight vector corresponding to each feature and C is the regularization parameter that balances between model fit and complexity.

We used Weka's (Hall et al. 2009) implementation of naïve Bayes and Weka's interface to L_2-regularized logistic regression implementation by LibLinear (Fan et al. 2008) using the default regularization parameters.

Note that `UNC` uses a single model whereas `QBC` uses a committee of classifiers, trained through bagging (Breiman 1996), to select the next set of instances to be labeled. For bagging, we used the default setting in Weka, which resulted in 10 bags, corresponding to 10 committee members. For a fair comparison between `UNC` and `QBC`, we used the `QBC` committee only for selecting the next set of instances and not for evaluation; for evaluation purposes, `QBC` trained an additional single classifier on

Table 3 Confusion matrix for a binary classification problem

		Predicted class	
		Positive	Negative
True class	Positive	True positive (tp)	False negative (fn)
	Negative	False positive (fp)	True negative (tn)

the entire labeled data. Otherwise, we would be comparing a single classifier of UNC to a bagged classifier of QBC.

To seek an answer to question 4, we compared RND and AL using accuracy, AUC, precision, recall, and F_1 on datasets of varying class imbalance. Table 3 presents the confusion matrix for binary classification and the notation used for the metrics. A confusion matrix compares the test results of a classifier with the gold standard, counting the number of correctly classified instances (true positive and true negative) and the number of incorrectly classified instances (false positive and false negative). Precision, recall and F_1 metrics are calculated based on the positive class. Which class is treated as the positive class depends on the domain. In this article, we treat the minority class as the positive class because predicting the minority class is often harder. Based on the confusion matrix, the five performance measures used in this study are defined in Table 4. For all thresholded metrics, we use the default classification threshold of 0.5.

4 Results and discussion

In this section, we present the results for investigating the answers to the questions posed in Sect. 3.

4.1 AL versus RND

Tables 5 and 6 show how many times UNC and QBC statistically significantly won (W), tied (T), or lost (L) to RND for naïve Bayes (NB) and logistic regression (LR) classifiers. Statistical significance between an AL strategy and RND was measured through a paired t test where the pairs are the learning curves of AL and RND. A p value of 0.05 was used to measure significance (Win or Loss). If AL is statistically significantly better than RND, it is a Win (W), if AL is significantly worse than RND, it is a Loss (L), and if the differences are not significant, it is a Tie (T). W/T/L in a *cell* should sum to 20 for synthetic datasets and to 10 for real datasets.

– *Result 1:* AL *performed well in comparison to* RND *on both synthetic and real datasets, when compared using accuracy, precision, and* F_1. For example, both UNC and QBC statistically significantly outperformed RND on *all* ten real-world datasets for NB on these measures. For LR, AL performed significantly better than RND on at least seven out of ten datasets for these performance measures.

Table 4 Performance measures used in this article

Measure	Formula	Description
AUC	Area under an ROC curve (Hanley and McNeil 1982)	Probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance
Accuracy	$\frac{tp+tn}{tp+fn+fp+tn}$	Percentage of instances that are predicted correctly
Precision	$\frac{tp}{tp+fp}$	Percentage of instances that are true positive out of all the instances that are predicted as positive
Recall	$\frac{tp}{tp+fn}$	Percentage of instances that are true positive out of all the positive instances
F_1	$\frac{2\times Precision\times Recall}{Precision+Recall}$	Harmonic mean of Precision and Recall

The formulas of performance measures are based on the confusion matrix presented in Table 3

Table 5 UNC vs. RND

DATA-CLF	AUC W/T/L	ACCU W/T/L	F_1 W/T/L	PREC W/T/L	REC W/T/L
SYN-NB	**15**/0/5	**14**/1/5	**15**/1/4	**11**/2/7	**16**/1/3
SYN-LR	**20**/0/0	**17**/1/2	**18**/2/0	**18**/2/0	**18**/2/0
REAL-NB	**5**/0/**5**	**10**/0/0	**10**/0/0	**10**/0/0	2/1/**7**
REAL-LR	4/1/**5**	**7**/2/1	**8**/0/2	**7**/1/2	**8**/1/1

Results compare the learning curves of UNC against RND for NB and LR classifiers. Win (W), tie (T), and loss (L) counts of statistical significance test (t test) per measure: AUC, accuracy (ACCU), F_1, precision (PREC), and recall (REC). Results are grouped by synthetic data (SYN) and real-world data (REAL)

Table 6 QBC versus RND

DATA-CLF	AUC W/T/L	ACCU W/T/L	F_1 W/T/L	PREC W/T/L	REC W/T/L
SYN-NB	**12**/0/8	**13**/3/4	**12**/3/5	8/3/**9**	**13**/2/5
SYN-LR	**15**/0/5	**15**/5/0	**15**/5/0	**13**/6/1	**15**/5/0
REAL-NB	**5**/0/**5**	**10**/0/0	**10**/0/0	**10**/0/0	**5**/0/**5**
REAL-LR	3/1/**6**	**9**/1/0	**8**/0/2	**7**/0/3	**8**/0/2

Results compare the learning curves of QBC against RND for NB and LR classifiers. Win (W), tie (T), and loss (L) counts of statistical significance test (t test) per measure: AUC, accuracy (ACCU), F_1, precision (PREC), and recall (REC). Results are grouped by synthetic data (SYN) and real-world data (REAL)

– *Result 2:* RND *was fairly competitive for AUC on real datasets.* RND won significantly over AL on at least five out of ten datasets for AUC for both classifiers.

Table 7 QBC versus UNC

DATA-CLF	AUC W/T/L	ACC. W/T/L	F_1 W/T/L	PREC. W/T/L	REC. W/T/L
SYN-NB	**9**/5/6	**10**/5/5	4/4/**12**	**10**/2/8	2/1/**17**
SYN-LR	3/2/**15**	4/1/**15**	2/3/**15**	5/2/**13**	2/3/**15**
REAL-NB	**5**/0/**5**	2/0/**8**	**5**/0/**5**	0/2/**8**	**8**/0/2
REAL-LR	**5**/1/4	**6**/2/2	3/2/**5**	4/1/**5**	4/0/**6**

Results compare the learning curves of QBC against UNC for NB and LR classifiers. Win (W), tie (T), and loss (L) counts of statistical significance test (t test) per measure: AUC, accuracy (ACCU), F_1, precision (PREC), and recall (REC). Results are grouped by synthetic data (SYN) and real-world data (REAL)

This result is especially interesting, because a mere 6 out of 54 papers used AUC to evaluate their active learning approaches (Table 1). This result shows that UNC did not improve AUC performance over what RND was able to achieve.

These results provide clear empirical evidence that both UNC and QBC are not suited well for improving AUC. There are alternatives that one can consider if the target performance metric is AUC. For example, Saar-Tsechansky and Provost (2004) modified UNC to perform better for ranking tasks. Instead of picking the top uncertain instances, they used the uncertainties as weights and they performed weighted sampling to pick instances for labeling.

4.2 QBC versus UNC

Table 7 shows how many times QBC statistically significantly won (W), tied (T), or lost (L) to UNC.

- *Result 3: On synthetic datasets, there is no clear winner for* NB*. For* LR*, however,* UNC *is a clear winner over* QBC *on all measures.*
- *Result 4: On real-world datasets, there is no clear winner between* QBC *and* UNC *for both classifiers.* For NB, however, even though UNC and QBC are comparable on F_1, UNC has the lead on precision and QBC has the lead on recall.

Looking at result 4 in depth shows that for NB, QBC and UNC are making trade-offs across measures. For example, on real datasets, QBC tilts the balance in favor of recall, whereas UNC tilts the balance in favor of precision. Figure 1 presents the precision and recall results on LetterO dataset, illustrating this behavior.

In general, we can expect a better performance of AL when the objective is aligned with the performance metric. However, formulating an AL approach that targets a specific performance measure is not trivial. A possible approach is to calculate the value of information of acquiring the label of an instance, where the "value" is the expected increase in desired performance measure. There are three possible solutions to computing the expected increase in the desired performance measure.

The first solution is to use a validation dataset. This is the ideal case, only if there is enough labeled data where some of it can be left for validation. However, separating

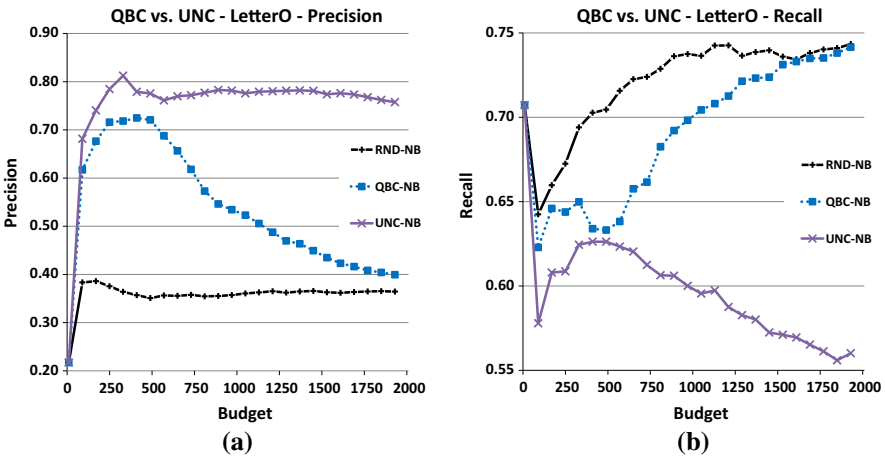

Fig. 1 For NB, very often, UNC outperforms QBC on precision and QBC outperforms UNC for recall on the same dataset. **a** Precision. QBC statistically significantly loses to UNC on 97 % of the learning curve. **b** Recall. QBC statistically significantly wins over UNC on 90 % of the learning curve

a part of the data for validation is especially challenging for active learning scenarios where the labeled data is scarce.

The second solution is to perform cross-validation on the labeled data. This approach, however, has a major drawback; cross-validation on a small labeled data, which is collected using a biased labeling strategy, cannot yield unbiased estimates of the performance measure.

The third solution is to formulate an unsupervised proxy for the desired performance measure and compute it on the unlabeled data, which is abundant and unbiased. This is indeed the approach taken by Roy and McCallum (2001) for reducing the expected error, where the confidence of the classifier is used as a proxy for classification error. Similarly, Culver et al. (2006) proposed an AL method that maximizes AUC of the hypothesis, using a semi-supervised ranking approach. Long et al. (2010) and Bilgic and Bennett (2012) maximized discounted cumulative gain (DCG) performance measure to select the most informative instances.

The main challenge, however, is to formulate an appropriate unsupervised proxy for the desired performance measure, which is not all that trivial. For example, a more confident classifier is not necessarily a more accurate classifier.

W/T/L tables can provide only so much information. There are a number of important results that are not apparent from these tables. We next provide detailed results discussing the choice of the classifier, trade-offs across performance metrics, long-term effect of active learning, and the effect of bootstrap size.

4.3 Choice of the classifier

Result 5: Model selection, which is not trivial for AL*, provides improvements beyond what* AL *can provide.* It is well-known in the data mining community that a single

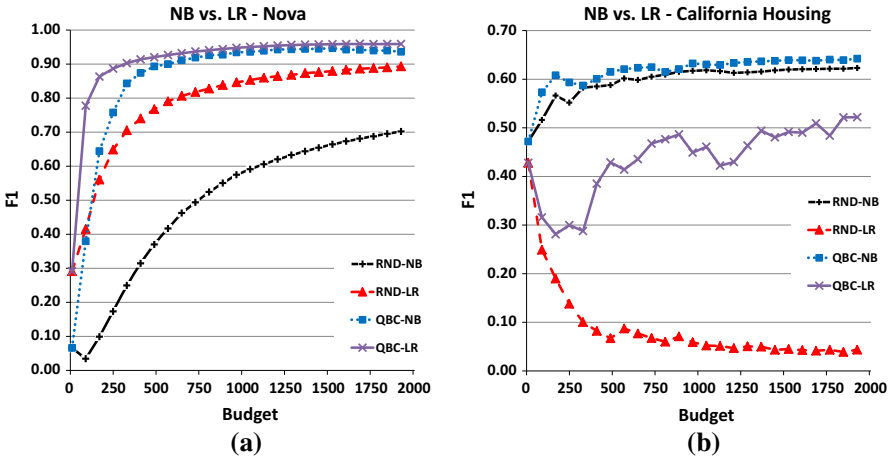

Fig. 2 Model selection (NB vs. LR) provides improvements beyond what AL can provide. F_1 is shown as an example. **a** A case where RND-LR outperforms RND-NB and AL-LR outperforms AL-NB on all measures. **b** A case where RND-NB outperforms even AL-LR

classifier is never the best performing classifier for all datasets and domains. On the ten real-world datasets (Table 2), we observed that RND-LR outperformed RND-NB for *all* measures on only two datasets (Ibn Sina and Nova). Similarly, RND-NB outperformed RND-LR for *all* measures on only two datasets (California Housing and Sylva). In each of the remaining six datasets, one classifier was better at one measure while the other was better at another measure.

We observed that, not surprisingly, when RND of one classifier was better than RND of the other classifier, AL of the superior classifier was also better than the AL of the inferior classifier (e.g., Fig. 2a). More importantly, RND of a superior classifier was better than the AL of the inferior classifier in a considerable number of cases. For example, RND-NB outperformed AL-LR on five out of 10 real datasets on recall. Figure 2b shows an example for F_1.

These results provide empirical evidence that model selection provides improvements beyond those that AL can provide; that is, random sampling with an appropriate model for a given domain can easily surpass active learning with an inappropriate model for that domain. Thus, it is fair to say that in practice, one should pay no less attention to selecting an appropriate model than to devising an active learning approach.

However, model selection, which is often overlooked in the current active learning literature, is not trivial in active learning settings. Two potential approaches for model selection are having a validation set or performing cross validation. The drawbacks of these two approaches in the context of active learning have been discussed in the context of result 4. There are studies that address the model selection for active learning. Recently, Ali et al. (2014) studied model selection during active learning, where the proposed algorithm actively selects instances for learning based on a set of candidate models, and simultaneously selects unbiased instances for model selection.

Result 6: Additional bias can help random sampling outperform active learning. Our synthetic datasets were generated using a naïve Bayes model (Sect. 3.1). When we compared learning a NB versus learning an LR on the synthetic datasets, we observed that RND-NB outperformed RND-LR and AL-NB outperformed AL-LR for *all* measures on all synthetic datasets except the most extremely-imbalanced ones.[1] RND-NB sometimes outperformed even AL-LR (e.g., Fig. 3a).

Note that both NB and LR are linear classifiers; NB has the additional bias that the features of the data are conditionally independent given class. In the synthetic datasets, this bias happens to be true, giving NB a clear advantage over LR. Even though NB and LR are expected to perform similarly given unlimited training data, the additional correct bias plays an important role in AL settings where training data is severely limited. However, even when the bias of NB is incorrect, it can still outperform LR when the training data is limited, as shown by Ng and Jordan (2002).

We designed an experiment to compare NB and LR on a dataset where NB does not have the correct bias. We modified the same synthetic dataset presented in Fig. 3a to add an incorrect bias for NB by randomly selecting 10 % of the features and duplicating them t times. Note that the duplicate features in a dataset are not conditionally independent given class, and thus, NB will have an incorrect bias. We experimented with various values for t and present results with $t = 3$ in Fig. 3b and with $t = 6$ in Fig. 3c. The incorrect bias for NB is greater for larger values of t. The results show that when the incorrect bias for NB is small, e.g. when $t = 3$, RND-NB performs worse than RND-LR, however AL-NB is still able to outperform AL-LR. With even more incorrect bias, e.g. when $t = 6$, RND-NB performs worse than RND-LR, and AL-NB performs worse than AL-LR. Note that LR for both RND and AL, is affected little with the replication of the features because LR does not assume features are conditionally independent given class, whereas NB, especially NB-RND, is significantly hurt by the incorrect bias.

Result 7: A seeming advantage of AL *over* RND *can be misleading.* Based on the W/T/L results (Table 5), we see that for recall, UNC lost to RND on seven out of ten datasets for NB, whereas UNC won over RND on eight out of ten real-world datasets for LR. A paper that used only NB in its empirical study could conclude that UNC hurts recall, whereas a paper that used only LR could conclude that UNC improves recall.

A closer examination, however, revealed surprising results. Even though AL-NB was worse than RND-NB, and AL-LR outperformed RND-LR for recall, AL-NB still significantly outperformed AL-LR on six of the ten real datasets (e.g., Fig. 4a and b). In three of these datasets, the recall of RND-LR eventually converged to zero and none of the AL strategies could help it (e.g., Fig. 4b), whereas this never happened for RND-NB and AL-NB. Therefore, the results in papers that use only a single classifier in their experiments (i.e., 83 % of the papers that we analyzed) should be qualified to emphasize that those results apply only for the classifier evaluated in the paper; general conclusions, such as UNC helps/hurts recall over RND, should be avoided.

[1] In the most extremely-imbalanced synthetic datasets (1 % positive class distribution), the results were mixed across measures.

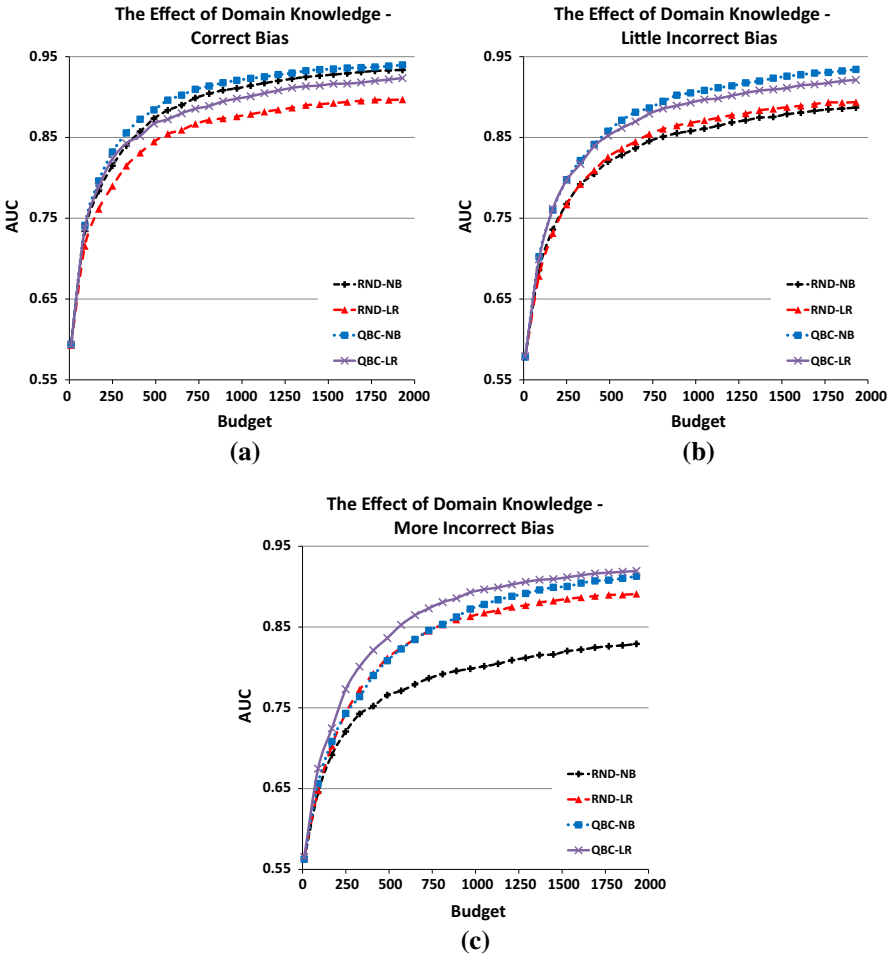

Fig. 3 Additional bias can help random sampling. NB had the additional correct bias on synthetic data. RND-NB and AL-NB outperformed RND-LR and AL-LR respectively on all synthetic datasets except the most extremely-imbalanced ones. **a** The result on a synthetic dataset with 25 % minority class distribution, where NB had the additional correct bias. RND-NB outperformed RND-LR and AL-LR. **b** The result on the same dataset with incorrect bias for NB. RND-LR outperforms RND-NB, however AL-NB still outperforms AL-LR. **c** The result on same dataset with even more incorrect bias for NB. AL-LR outperformed AL-NB, and RND-LR outperformed RND-NB

4.4 Performance measure trade-offs

Result 8: Improvements across the board were rare. Improvement in one measure often came at the expense of another. Note that it is trivial to increase one performance measure at the expense of another. For example, one can increase precision and accuracy at the expense of recall by simply changing the decision threshold. In Fig. 6, we show an example where by simply changing the decision boundary, RND improves over AL for accuracy and precision at the expense of recall. We describe the details of the

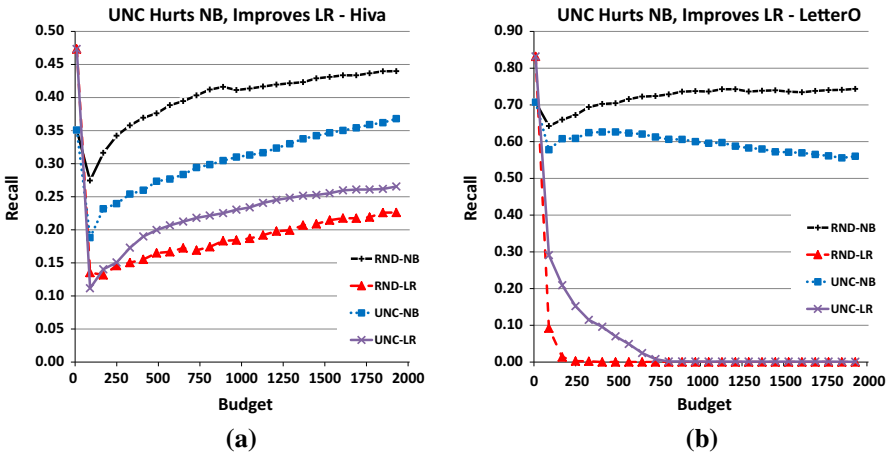

Fig. 4 Studies using a single classifier can have misleading results. **a** UNC loses to RND for NB but it improves for LR. However, still UNC-NB significantly outperforms UNC-LR. **b** RND-LR eventually converges to zero and AL cannot help it. This never happens for NB

experiment in Fig. 6 in the next result. Similarly, it is trivial to increase accuracy in a highly-skewed dataset by simply changing the threshold in the favor of the majority class. Ideally, an AL method should not simply change the decision boundary of the model, but improve learning to better distinguish between the instances of different classes, and thus achieve a better performance across most measures.

Even though there are cases where AL improved over RND across all performance measures, they are rare. For LR, QBC and UNC improved performance across all measures on only three datasets (California Housing, Nova, and Sylva). For NB, QBC improved performance on four datasets, whereas UNC improved performance on only two datasets across all measures. On all other datasets, the improvement in one performance measure came at the expense of another measure.

Result 9: AL *often improved accuracy and precision at the expense of recall*. We show an example in Fig. 5. Figure 5a shows that AL improved over RND on accuracy for California Housing dataset, whereas Fig. 5b shows that AL hurt recall for the same dataset.

In practice, one should adjust the decision threshold based on a targeted performance metric. In this study, we fixed the threshold at 0.5 to observe how active learning affects different performance metrics. For example, we observed that AL often outperformed RND on accuracy and performed poorly on recall. Moreover, AL struggled to beat RND on AUC. This result suggests that AL simply shifted the decision threshold rather than improving learning. We were able to make this observation by keeping the decision threshold constant at 0.5.

This raises the question how much AL is really improving learning when it is essentially improving one measure at the expense of another. Is the improvement due to more effective learning or simply due to a shift in the decision threshold caused by biased sampling? This doubt about AL's effectiveness is magnified, given that RND is very competitive for AUC (Result 2), the only measure in our evaluations that does not

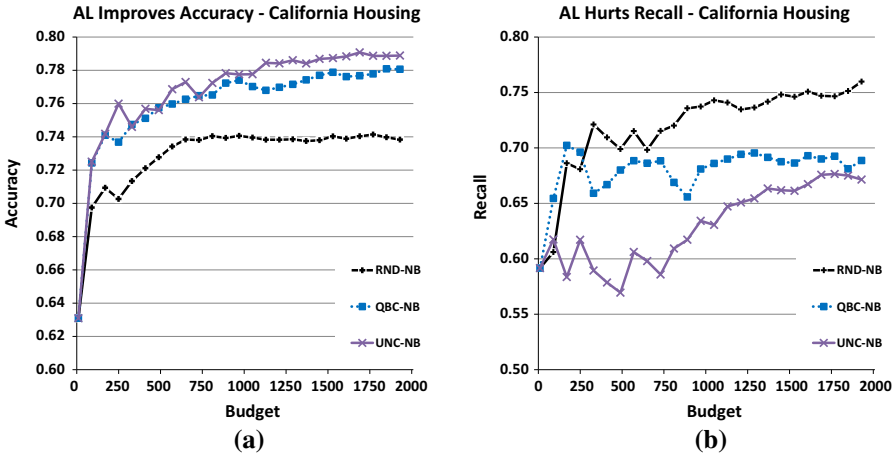

Fig. 5 AL often improved accuracy and precision at the expense of recall. **a** AL improved accuracy for California Housing dataset. **b** AL hurt recall for the same dataset

require a decision threshold. It is also worth noting that only 6 out of 54 papers that we analyzed used AUC as the performance measure for their experiments (Table 1).

Figure 6 shows an example case on Zebra dataset, where UNC improves over RND in precision and accuracy but loses in recall and AUC. We designed an experiment where we chose instances at random but shifted the classification decision boundary away from 0.5 in favor of the majority class; we call this modified method RND+DB. Note that the AUC performance of RND+DB is not affected by the shift in the boundary (Fig. 6a), however it outperforms RND in precision and accuracy (Fig. 6b, c), and performs worse than RND in recall, imitating UNC (Fig. 6d). This result suggests that for the Zebra dataset, UNC was able to outperform RND in accuracy and precision simply by shifting the decision threshold, rather than choosing more informed training instances.

The results of typical active learning empirical evaluations (where one classifier - one measure combination is used) can be misleading. The claimed improvements can be due to a poorly chosen classifier (Results 5, 6, and 7) or due to the specific performance measure discussed in the paper (Results 8 and 9). Given that in our literature review, 83 % of empirical studies used a single classifier and 91 % concentrated on only one performance measure, unless the classifier and performance measure choices are justified by the underlying domain, the results of these studies should be taken with a grain of salt.

Next, we analyze how AL affects the results in the long run and then discuss the effect of the initial training data (bootstrap) size on AL.

4.5 Long-term effect

Result 10: Continuous labeling with AL *can do more harm than good.* Most experiments in the active learning literature consider small budget sizes, perhaps for the right

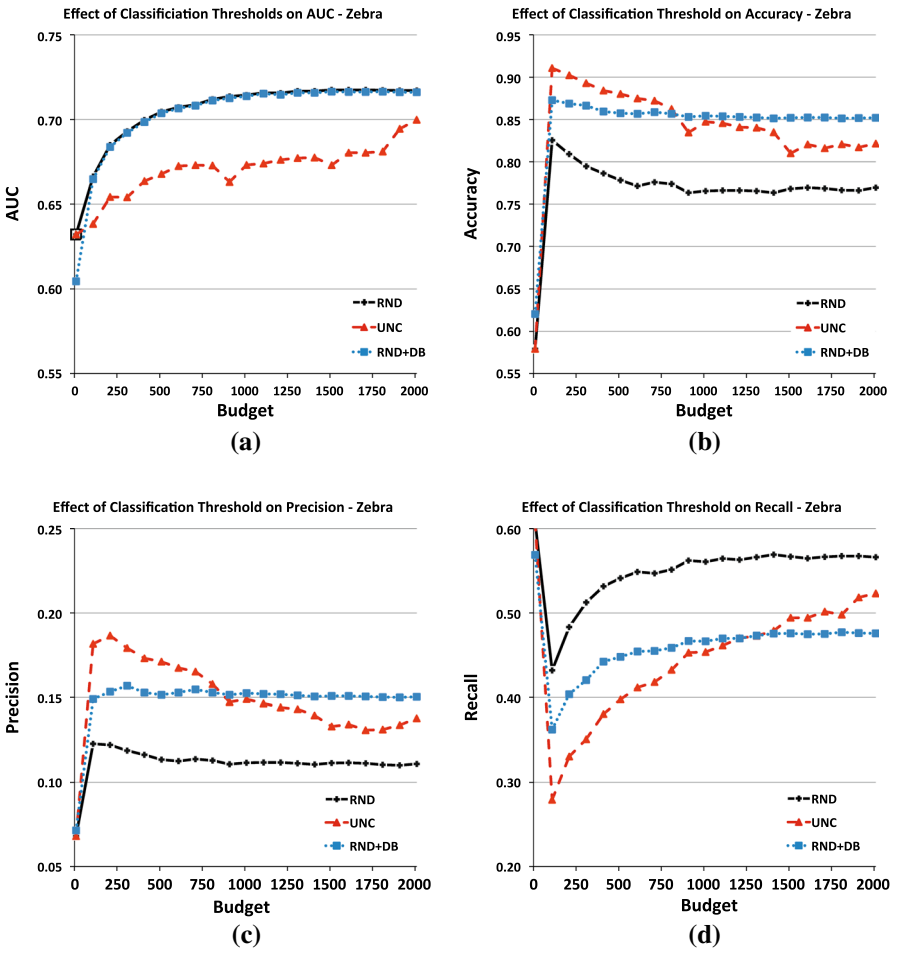

Fig. 6 AL with a classification boundary bias for Zebra dataset. RND and UNC are the unmodified methods and RND+DB modifies the decision boundary. **a** A modified classification threshold does not affect AUC performance of RND. **b** Effect of classification threshold on accuracy. **c** Effect of classification threshold on recall. **d** Effect of classification threshold on precision

reason that labeling is expensive. However, we often do not know the budget a priori, and thus it is important to consider cases where the budget is large (but still limited compared to the size of the data). If the labeled data is continuously collected through AL, the danger is that because the data labeled by AL is not a random sample but is a biased sample, this bias can actually hurt the performance. We empirically tested this conjecture by experimenting with large budget values.

Our main results up to this section were based on a budget of 2,000 labels. For this section, we doubled the budget to 4,000. Even though not all datasets and all measures were negatively affected, we have seen a considerable number of cases where continuous labeling with AL hurts the performance in the long run. We present

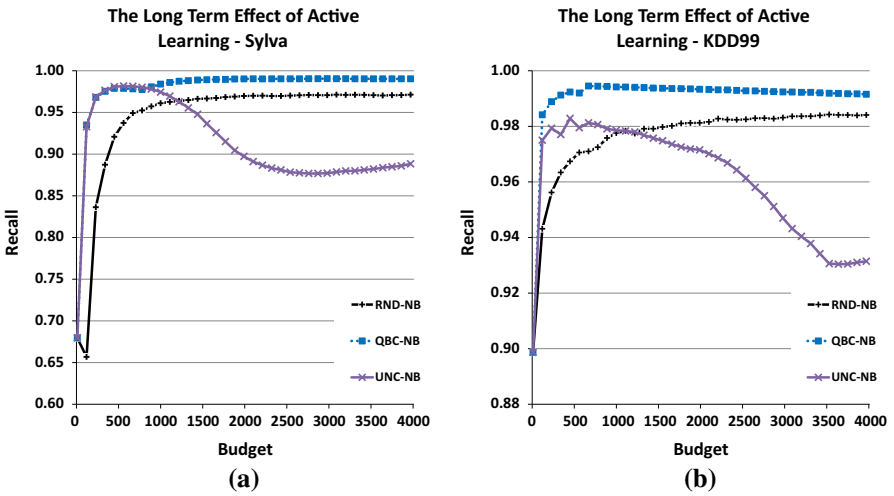

Fig. 7 Continuous labeling with UNC does more harm than good. QBC is more robust. Long-term effect of active learning on **a** Sylva dataset and **b** KDD99 dataset

two examples in Fig. 7 where UNC improved over RND initially and then it started losing.

Result 11: UNC *and* NB *were more prone to the negative effects of continuous labeling with* AL. Our results suggest that NB is more prone to the negative effect of long-term labeling with AL, especially for UNC. For example, we observed negative effects of long-term use of UNC with NB on four out of 10 datasets for recall. The negative long-term effects were observed for UNC with LR as well but with much less frequency compared to UNC with NB. QBC was more robust than UNC for both classifiers. Because it is difficult to determine when the negative results start to take effect, our results suggest utilizing QBC instead of UNC to avoid the long-term negative effect of labeling with AL. A possible approach is to alternate between random sampling and active learning to counter-balance the negative effects of biased sampling.

4.6 Bootstrap size

Result 12: Using a larger-size initially-labeled data never made a losing AL *strategy a winning strategy or vice versa.* AL papers make varying choices for the size of the initially labeled training data. Some papers start with a small random sample (Baldridge and Osborne 2004; Guo and Schuurmans 2008), while others (Schein and Ungar 2007; Long et al. 2010) use a large set. We investigated the effect of the size of initially labeled data on the performance of AL.

Our results so far were based on an initial training data size of 10: five random instances from each class. In this section, we present results for using larger bootstrap sizes: in addition to 10 initially chosen instances, 100, 500, or 1, 000 instances were chosen randomly and added to the initially labeled data. Interestingly, with a larger bootstrap size, a winning active learning method never became a losing method and

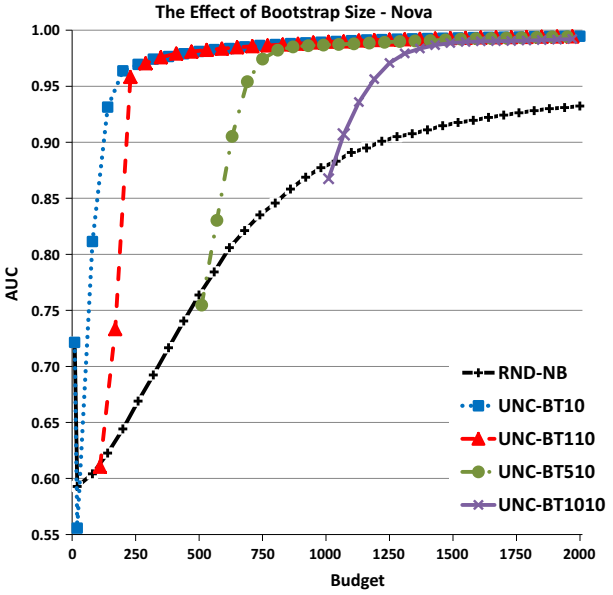

Fig. 8 The size of the initially labeled data does not make a big difference to the performance of AL. AL on Nova dataset using different bootstrap set sizes

vice versa. Eventually, the performance of AL with a larger bootstrap converged to the AL with a smaller bootstrap for all datasets and often the convergence was quite fast, requiring fewer than ten iterations. We show one example in Fig. 8.

This is a surprising result. One would expect that a larger bootstrap would allow for a more representative dataset to start the AL process, and thus, cushion against the negative effects of AL. Schein and Ungar (2007) also observed similar behavior that a larger bootstrap had little effect on AL. The reasons are yet unclear to us. Further analysis is needed to tease out the precise effect the initially labeled training data size has on AL.

5 Related work

The area of active learning has received a lot of attention in the machine learning community. A comprehensive survey is beyond the scope of this article. We refer the interested reader to Settles (2012) and Fu et al. (2012). In this section, we summarize only the general trends and provide only a few example references.

Active learning can be thought of as a specific case of eliciting domain knowledge from experts, which can be traced back to the expert systems (Giarratano and Riley 1998). Active learning refers to the case where the underlying model *actively* constructs its own training data by consulting an expert. The consultation with the expert is often done through posing membership queries, such as asking the expert the class that a specific example belongs to. Several approaches have been developed to tackle the "which example should be labeled" question. Examples include uncertainty sampling

(Lewis and Gale 1994), query-by-committee (Seung et al. 1992; Dagan and Engelson 1995), and expected error reduction (Roy and McCallum 2001).

Earliest examples of active learning made several simplifying assumptions. For example, it was assumed that the queries had a uniform cost, and hence the approaches simply optimized over the number of queries. Cost-sensitive active learning lifted this assumption by allowing each query to have a different cost and optimized over the total cost of the queries (Arora et al. 2009; Haertel et al. 2008; Kapoor et al. 2007; Margineantu 2005; Settles et al. 2008; Tomanek et al. 2007; Vijayanarasimhan and Grauman 2009). Another assumption made by the early work was that the experts were infallible, and hence they were called the oracles. Recent work allowed noisy experts whose answers might be wrong (Donmez and Carbonell 2008; Donmez et al. 2009; Sheng et al. 2008; Wallace et al. 2011). Finally, the earliest work assumed that the learner posed one query at a time and hence they searched for the single best query, though in practice, each one of the examples was given a score and more than one example was chosen to be labeled. Later work looked at batch-mode query selection (Guo and Schuurmans 2008; Hoi et al. 2006b), where a number of training examples, instead of a single example, are selected for querying at each iteration.

Recent active learning work looked at various types of queries and settings. For example, Bilgic et al. (2010) and Jensen et al. (2004) looked at active learning for networked data where the nodes of the network had correlated labels; Melville et al. (2004) and Bilgic and Getoor (2011) looked at acquiring costly missing feature values, such as ordering laboratory tests for medical diagnosis; Melville and Sindhwani (2009), Druck et al. (2009), Small et al. (2011), Raghavan and Allan (2007), and Attenberg et al. (2010) labeled features with class labels; Sharma et al. (2015) and Zaidan et al. (2007) elicited rationales in addition to labels; Ramirez-Loaiza et al. (2013, 2014); Ramirez-Loaiza (2016) extracted snippets from instances to speed-up annotation; Qian et al. (2013) looked at designing easier queries for experts to answer; Bilgic and Bennett (2012), and Long et al. (2010) formulated active learning for ranking; Bilgic (2012) combined active learning with dynamic dimensionality reduction; and so on.

A closely-related area is *active inference*, where the queries are posed to the experts not during training of the model but rather during prediction time. The objective is to pose as few queries as possible while maximizing prediction accuracy during testing. For this to work, the labels of the test instances need to be correlated, such as networked data. Rattigan et al. (2007) and Bilgic and Getoor (2008, 2009, 2010) looked at active inference for classifying nodes of a network; Chen et al. (2009, 2011a, b) formulated active inference for video analysis; Komurlu et al. (2014) formulated active inference for tissue engineering; Komurlu and Bilgic (2016) looked at battery optimization in wireless sensor networks as an active inference problem; and so on.

A big challenge in practical use of active learning is to determine which method to use when. One approach is to perform theoretical analyses to quantify label complexities of various active learning approaches (Balcan et al. 2008; Dasgupta et al. 2007; Hanneke 2007; Wang 2009). An alternative and complementary approach is to perform empirical studies, like this article. There are a number of extensive active learning empirical studies, but to our knowledge, these studies concentrated on a single classifier and a single performance measure. Settles and Craven (2008) com-

pared several variations of uncertainty sampling and query-by-committee for sequence labeling. They concentrated on F_1 as the performance measure and they used conditional random fields (Lafferty et al. 2001) as the underlying model. Schein and Ungar (2007) evaluated a number of uncertainty-based approaches and query-by-committee for logistic regression and they focused on only accuracy. In this article, we performed an empirical study across classifiers and performance measures. Our most important findings stem from this difference.

6 Limitations and future work

Our empirical evaluation included only two of the most common active learning strategies (i.e., uncertainty sampling and query-by-committee) and only two classifiers (naïve Bayes and logistic regression). There are numerous other active learning approaches (e.g., expected error reduction (Roy and McCallum 2001), variance reduction (Cohn et al. 1996), conflicting uncertainty (Sharma and Bilgic 2013; Sharma and Bilgi 2016), etc.) and numerous other classifiers (e.g., support vector machines, decision trees, nearest neighbors, etc.) that we did not include in our study. Nonetheless, our comparison of two active learning approaches across two classifiers and five performance measures revealed interesting results that we hope will raise more questions and perhaps awareness about empirical evaluations of active learning approaches.

We have created a dedicated website at http://www.cs.iit.edu/~ml/projects/empirical-study for empirical comparison of active learning strategies. Currently, the comparison includes random sampling, uncertainty sampling, and query-by-committee. We plan on continuously updating the experimental results on this website to include more classifiers and active learning methods over time.

7 Conclusion

We performed a large number of experiments evaluating common active learning strategies using different classifiers and performance measures on several datasets with various domain characteristics. Our experiments revealed interesting and useful insights that we hope will help the research community for more in-depth evaluations of active learning approaches, and will serve as guiding principles for individuals and companies utilizing active learning in the real-world settings.

Acknowledgements This material is based upon work supported by the National Science Foundation CAREER award no. IIS-1350337.

Appendix: Open-source active learning library: `PyAL`

We performed the experiments in this article using Weka; for naïve Bayes, we used Weka's own implementation and for logistic regression, we used Weka's interface to LibLinear (Fan et al. 2008), version 1.7. We later re-wrote the code in Python,

integrating it with scikit-learn (Pedregosa et al. 2011) and released it as open source under the name PyAL.

We created a dedicated website for this project at http://www.cs.iit.edu/~ml/projects/empirical-study. The website currently has:

- The Java libraries that are necessary to repeat the experiments performed in this paper
- The synthetic datasets that were used in this study
- The link to the GitHub repository for the PyAL library
- A side-by-side comparison of the results obtained using the Java version of the code versus PyAL

Similarities and differences between PyAL and weka results

We repeated the main set of experiments using PyAL and compared them side-by-side with the results we obtained using Weka. To save space, we included the figures on the project website http://www.cs.iit.edu/~ml/projects/empirical-study and here, we include only the *t* test results in Tables 8, 9, and 10.

The actual win/tie/loss counts using Weka and PyAL are not identical, but they vary very little and hence the trends and the main results obtained using Weka also hold for PyAL. We discuss some of the similarities and differences between the Weka and PyAL implementations.

Logistic regression results

The experiments in this paper used Weka's interface to LibLinear version 1.7 for logistic regression. Scikit-learn's logistic regression also uses LibLinear under the

Table 8 UNC vs. RND using PyAL.

DATA-CLF	AUC W/T/L	ACCU W/T/L	F_1 W/T/L	PREC W/T/L	REC W/T/L
REAL-NB	4/0/**6**	**7**/2/1	**6**/1/3	**10**/0/0	1/0/**9**
REAL-LR	3/1/**6**	**6**/3/1	**8**/1/1	**6**/0/4	**8**/2/0

Win (W), tie (T), and loss (L) counts of statistical significance test (*t* test) results comparing the learning curves of UNC against RND. The equivalent win (W), tie (T), and loss (L) results using Weka are in Table 5

Table 9 QBC vs. RND using PyAL

DATA-CLF	AUC W/T/L	ACCU W/T/L	F_1 W/T/L	PREC W/T/L	REC W/T/L
REAL-NB	3/1/**6**	**6**/1/3	**7**/1/2	**7**/1/2	3/2/**5**
REAL-LR	3/3/**4**	**7**/3/0	**7**/1/2	**7**/1/2	**7**/2/1

Win (W), tie (T), and loss (L) counts of statistical significance test (*t* test) results comparing the learning curves of QBC against RND. The equivalent Win (W), tie (T), and loss (L) results using Weka are in Table 6

Table 10 QBC vs. UNC using PyAL

DATA-CLF	AUC W/T/L	ACCU W/T/L	F_1 W/T/L	PREC W/T/L	REC W/T/L
REAL-NB	**4**/2/**4**	2/2/**6**	4/1/**5**	1/3/**6**	**8**/1/1
REAL-LR	4/**5**/1	4/**6**/0	3/**4**/3	3/**6**/1	2/2/**6**

Win (W), tie (T), and loss (L) counts of statistical significance test (*t* test) results comparing the learning curves of QBC against UNC. The equivalent win (W), tie (T), and loss (L) results using Weka are in Table 7

hood and hence logistic regression results are almost identical (modulo the random number sequences of Python vs. Java) for most datasets. The biggest visible difference occurs for the California Housing dataset and we verified that this difference is due to an older version of LibLinear port that Weka used.

Naïve Bayes results

For naïve Bayes, scikit-learn has two generic naïve Bayes implementations: a Bernoulli naïve Bayes for datasets with binary features and a Gaussian naïve Bayes for datasets with continuous features. Weka on the other hand has a generic naïve Bayes implementation that can work with datasets that have mixed feature types: binary, continuous, and categorical.[2]

For datasets that contain only binary features, both scikit-learn's and Weka's naïve Bayes implementations are identical. All synthetic datasets and two real datasets, Hiva and Nova, have only binary features. For these datasets, the naïve Bayes results for both PyAL and Weka are almost identical, except minor differences in random number sequences of Python vs. Java.

For datasets that contain only continuous features, both scikit-learn and Weka's naïve Bayes implementation is Gaussian naïve Bayes. Six out of 10 real datasets have features that are all continuous (Table 2). The remaining two real datasets, KDD99 and Sylva, have mixed feature types. Weka's implementation of naïve Bayes can handle a mix of features whereas scikit-learn's naïve Bayes implementation requires all features to be either continuous or binary, and hence datasets need to be pre-processed to conform to one of these formats. For these datasets, there are visual differences between the learning curves, some of which are significant, between PyAL and Weka results, though as the *t* test tables show, the general conclusions (e.g., RND being competitive in AUC, etc.) still hold.

PyAL library

The PyAL code consists of:

– an active learning algorithm implementation, `learning_curve`, that given the parameters for an active learning session, such as the underlying classifier, an active

[2] Both Weka and scikit-learn have a multinomial naïve Bayes implementation for text classification.

learning strategy, and a budget, runs an active learning session, and evaluates the classifier at each step of the active learning,
- an active learning API, which provides the base classes for choosing a bootstrap, the base classes for choosing the next instance(s) to be labeled at each step of the labeling process, and implementation of a few active learning approaches,
- a command-line interface, which reads the active learning settings from a command line, that loads the dataset(s), runs the `learning_curve` code, plots the results, and saves the results to files,
- and a GUI interface written in Tkinter as a visual alternative to the command-line interface.

Currently implemented bootstrap strategies are (i) random sampling, where the initially labeled instances are chosen completely at random, and (ii) random sampling from each class, where equal number of random instances are chosen from each class. The code can be extended to implement additional bootstrap strategies, by extending the bootstrap class; for example, unsupervised batch-mode active learning strategies can be used to bootstrap the active learning process.

Currently implemented active learning approaches include uncertainty sampling (Lewis and Gale 1994), query-by-committee through bagging (Abe and Mamitsuka 1998), and expected error reduction (Roy and McCallum 2001), with a possibility to implement additional active learning strategies by extending the base strategy class.

A detailed documentation of the code, access to the GitHub repository, and Java executables can be found at http://www.cs.iit.edu/~ml/projects/empirical-study.

References

Abe N, Mamitsuka H (1998) Query learning strategies using boosting and bagging. In: Proceedings of the international conference on machine learning (ICML), pp 1–9

Ali A, Caruana R, Kapoor A (2014) Active learning with model selection. In: Proceedings of the AAAI conference on artificial intelligence, pp 1673–1679

Arora S, Nyberg E, Rose C (2009) Estimating annotation cost for active learning in a multiannotator environment. In: NAACL HLT Workshop on Active Learning for Natural Language Processing, pp 18–26

Attenberg J, Melville P, Provost F (2010) A unified approach to active dual supervision for labeling features and examples. In: Proceedings of the European conference on machine learning (ECML), pp 40–55

Balcan MF, Hanneke S, Wortman J (2008) The true sample complexity of active learning. In: Proceedings of the conference on learning theory (COLT), pp 45–56

Baldridge J, Osborne M (2004) Active learning and the total cost of annotation. In: Proceedings of the conference on empirical methods in natural language processing, pp 9–16

Bay SD, Kibler DF, Pazzani MJ, Smyth P (2000) The UCI KDD archive of large data sets for data mining research and experimentation. SIGKDD Explor 2:81

Bilgic M (2012) Combining active learning and dynamic dimensionality reduction. In: Proceedings of the SIAM international conference on data mining (SDM)

Bilgic M, Bennett PN (2012) Active query selection for learning rankers. In: Proceedings of the ACM SIGIR conference on research and development in information retrieval, pp 1033–1034

Bilgic M, Getoor L (2008) Effective label acquisition for collective classification. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 43–51

Bilgic M, Getoor L (2009) Reflect and correct: a misclassification prediction approach to active inference. ACM Trans Knowl Discov Data 3(4):1–32

Bilgic M, Getoor L (2010) Active inference for collective classification. In: Proceedings of the conference on artificial intelligence (AAAI NECTAR track), pp 1652–1655

Bilgic M, Getoor L (2011) Value of information lattice: Exploiting probabilistic independence for effective feature subset acquisition. J Artif Intell Res (JAIR) 41:69–95
Bilgic M, Mihalkova L, Getoor L (2010) Active learning for networked data. In: Proceedings of the international conference on machine learning (ICML)
Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
Chao C, Cakmak M, Thomaz AL (2010) Transparent active learning for robots. In: Proceedings of the ACM/IEEE international conference on human-robot interaction, pp 317–324
Chen D, Bilgic M, Getoor L, Jacobs D (2009) Efficient resource-constrained retrospective analysis of long video sequences. In: Proceedings of the NIPS workshop on adaptive sensing, active learning and experimental design: theory, methods and applications
Chen D, Bilgic M, Getoor L, Jacobs D (2011a) Dynamic processing allocation in video. IEEE Trans Pattern Anal Mach Intell 33:2174–2187
Chen D, Bilgic M, Getoor L, Jacobs D, Mihalkova L, Yeh T (2011b) Active inference for retrieval in camera networks. In: Proceedings of the workshop on person oriented vision
Cohn D, Atlas L, Ladner R (1994) Improving generalization with active learning. Mach Learn 15(2):201–221
Cohn DA, Ghahramani Z, Jordan MI (1996) Active learning with statistical models. J Artif Intell Res 4:129–145
Culver M, Kun D, Scott S (2006) Active learning to maximize area under the roc curve. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 149–158
Dagan I, Engelson SP (1995) Committee-based sampling for training probabilistic classifiers. In: Proceedings of the international conference on machine learning (ICML), pp 150–157
Dasgupta S, Monteleoni C, Hsu DJ (2007) A general agnostic active learning algorithm. In: Proceedings of the conference on advances in neural information processing systems (NIPS), pp 353–360
Donmez P, Carbonell JG (2008) Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In: Proceeding of the ACM conference on information and knowledge mining
Donmez P, Carbonell JG, Schneider J (2009) Efficiently learning the accuracy of labeling sources for selective sampling. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 259–268
Druck G, Settles B, McCallum A (2009) Active learning by labeling features. In: Proceedings of the conference on empirical methods in natural language processing, pp 81–90
Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: A library for large linear classification. J Mach Learn Res 9:1871–1874
Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. Mach Learn 28(2):133–168
Frey PW, Slate DJ (1991) Letter recognition using holland-style adaptive classifiers. Mach Learn 6:161
Fu Y, Zhu X, Li B (2012) A survey on instance selection for active learning. Knowl Inf Syst 35(2):249–283
Giarratano JC, Riley G (1998) Expert Systems, 3rd edn. PWS Publishing Co., Boston
Guo Y, Schuurmans D (2008) Discriminative batch mode active learning. In: Proceedings of the conference on advances in neural information processing systems (NIPS), pp 593–600
Guyon I, Cawley G, Dror G, Lemaire V (2011) Datasets of the active learning challenge. In: Proceedings of the JMLR workshop on active learning and experimental design, vol 16, pp 19–45
Haertel R, Seppi K, Ringger E, Carroll J (2008) Return on investment for active learning. In: NIPS workshop on cost-sensitive learning
Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18
Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143(1):29–36
Hanneke S (2007) A bound on the label complexity of agnostic active learning. In: Proceedings of the international conference on machine learning (ICML), pp 353–360
Hoi SC, Jin R, Lyu MR (2006a) Large-scale text categorization by batch mode active learning. In: Proceedings of the international conference on world wide web (WWW), pp 633–642
Hoi SCH, Jin R, Zhu J, Lyu MR (2006b) Batch mode active learning and its application to medical image classification. In: Proceedings of the international conference on machine learning (ICML), pp 417–424

Jensen D, Neville J, Gallagher B (2004) Why collective inference improves relational classification. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 93–598

Kapoor A, Horvitz E, Basu S (2007) Selective supervision: Guiding supervised learning with decision-theoretic active learning. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), vol 7, pp 877–882

Komurlu C, Bilgic M (2016) Active inference and dynamic gaussian bayesian networks for battery optimization in wireless sensor networks. In: Proceedings of AAAI workshop on artificial intelligence for smart grids and smart buildings

Komurlu C, Shao J, Bilgic M (2014) Dynamic bayesian network modeling of vascularization in engineered tissues. In: Proceedings of the 11th annual conference on uncertainty in artificial intelligence—workshop on bayesian modeling applications, pp 89–98

Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the international conference on machine learning (ICML), pp 282–289

Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: Proceedings of the ACM SIGIR conference on research and development in information retrieval, pp 3–12

Long B, Chapelle O, Zhang Y, Chang Y, Zheng Z, Tseng B (2010) Active learning for ranking through expected loss optimization. In: Proceedings of the ACM SIGIR conference on research and development in information retrieval, pp 267–274

Margineantu DD (2005) Active cost-sensitive learning. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 1622–1623

Melville P, Mooney RJ (2004) Diverse ensembles for active learning. In: Proceedings of the international conference on machine learning (ICML), pp 584–591

Melville P, Sindhwani V (2009) Active dual supervision: Reducing the cost of annotating examples and features. In: Proceedings of the NAACL HLT workshop on active learning for natural language processing, pp 49–57

Melville P, Saar-Tsechansky M, Provost F, Mooney R (2004) Active feature-value acquisition for classifier induction. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 483–486

Ng AY, Jordan MI (2002) On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In: Proceedings of the conference on advances in neural information processing systems (NIPS), vol 14, pp 841–848

Pace RK, Barry R (1997) Sparse spatial autoregressions. Stat Probab Lett 33(3):291–297

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Qian B, Wang X, Wang F, Li H, Ye J, Davidson I (2013) Active learning from relative queries. In: Proceedings of the international joint conference on artificial intelligence (IJCAI)

Raghavan H, Allan J (2007) An interactive algorithm for asking and incorporating feature feedback into support vector machines. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, pp 79–86

Ramirez-Loaiza ME (2016) Anytime active learning. PhD thesis, Illinois Institute of Technology

Ramirez-Loaiza ME, Culotta A, Bilgic M (2013) Towards anytime active learning: Interrupting experts to reduce annotation costs. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics (IDEA), pp 87–94

Ramirez-Loaiza ME, Culotta A, Bilgic M (2014) Anytime active learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2048–2054

Rattigan M, Maier M, Jensen D (2007) Exploiting network structure for active inference in collective classification. In: Proceedings of the international conference on machine learning workshop on mining graphs and complex structures, pp 429–434

Roy N, McCallum A (2001) Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the international conference on machine learning (ICML), pp 441–448

Saar-Tsechansky M, Provost F (2004) Active sampling for class probability estimation and ranking. Mach Learn 54(2):153–178

Schein AI, Ungar LH (2007) Active learning for logistic regression: an evaluation. Mach Learn 68(3):235–265

Sculley D (2007) Online active learning methods for fast label-efficient spam filtering. In: Proceedings of the conference on email and anti-spam, vol 7, pp 173–180
Segal R, Markowitz T, Arnold W (2006) Fast uncertainty sampling for labeling large e-mail corpora. In: Proceedings of the conference on email and anti-spam
Settles B (2012) Active learning. Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool Publishers, San Rafael
Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing, pp 1070–1079
Settles B, Craven M, Friedland L (2008) Active learning with real annotation costs. In: Proceedings of the NIPS workshop on cost-sensitive learning
Seung HS, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the conference on learning theory (COLT), pp 287–294
Sharma M, Bilgic M (2013) Most-surely vs. least-surely uncertain. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 667–676
Sharma M, Bilgic M (2016) Evidence-based uncertainty sampling for active learning. Data Min Knowl Discov 1–39
Sharma M, Zhuang D, Bilgic M (2015) Active learning with rationales for text classification. In: North American chapter of the association for computational linguistics. Human Language Technologies, pp 441–451
Sheng VS, Provost F, Ipeirotis PG (2008) Get another label? improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 614–622
Sindhwani V, Melville P, Lawrence RD (2009) Uncertainty sampling and transductive experimental design for active dual supervision. In: Proceedings of the international conference on machine learning (ICML), pp 953–960
Small K, Wallace BC, Brodley CE, Trikalinos TA (2011) The constrained weight space svm: learning with ranked features. In: Proceedings of the international conference on machine learning (ICML), pp 865–872
Thompson CA, Califf ME, Mooney RJ (1999) Active learning for natural language parsing and information extraction. In: Proceedings of the international conference on machine learning (ICML), pp 406–414
Tomanek K, Wermter J, Hahn U (2007) An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In: Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp 486–495
Tong S, Chang E (2001) Support vector machine active learning for image retrieval. In: Proceedings of the 9th ACM international conference on multimedia, pp 107–118
Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. J Mach Learn Res 2:45–66
Vijayanarasimhan S, Grauman K (2009) What's it going to cost you? predicting effort vs. informativeness for multi-label image annotations. In: Conference on computer vision and pattern recognition (CVPR), pp 2262–2269
Wallace BC, Small K, Brodley CE, Trikalinos TA (2011) Who should label what? Instance allocation in multiple expert active learning. In: Proceedings of the SIAM international conference on data mining (SDM), pp 176–187
Wang L (2009) Sufficient conditions for agnostic active learnable. In: Proceedings of the conference on advances in neural information processing systems (NIPS), pp 1999–2007
Xu Z, Yu K, Tresp V, Xu X, Wang J (2003) Representative sampling for text classification using support vector machines. Adv Inf Retr 393–407
Zaidan O, Eisner J, Piatko CD (2007) Using "annotator rationales" to improve machine learning for text categorization. In: Proceedings of annual conference of the north american chapter of the association for computational linguistics, Human Language Technologies, pp 260–267
Zhang C, Chen T (2002) An active learning framework for content-based information retrieval. IEEE Trans Multimed 4(2):260–268