Ask Me Better Questions: Active Learning Queries Based on Rule Induction

Parisa Rashidi
Washington State University
Pullman, Washington
prashidi@eecs.wsu.edu

Diane J. Cook
Washington State University
Pullman, Washington
cook@eecs.wsu.edu

ABSTRACT

Active learning methods are used to improve the classification accuracy when little labeled data is available. Most traditional active learning methods pose a very specific query to the oracle, i.e. they ask for the label of an unlabeled example. This paper proposes a novel active learning method called RIQY (**R**ule **I**nduced active learning **Q**uer**Y**). It can construct generic active learning queries based on rule induction from multiple unlabeled instances. These queries are shorter and more readable for the oracle and encompass many similar cases. Also the learning algorithm can achieve higher accuracy rates by asking fewer queries. We evaluate our algorithm on 12 different real datasets. Our results show that we can achieve higher accuracy rates using fewer queries compared to the traditional active learning methods.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Learning*; I.5.1 [**Computing Methodologies**]: Pattern Recognition—*Models*; H.2.8 [**Information Systems**]: DATABASE MANAGEMENT—*Data mining*

General Terms

Algorithms, Design, Performance, Human Factors

Keywords

active learning, machine learning

1. INTRODUCTION

In recent years, a variety of active learning methods have been proposed [26, 29] and it has been used in various application domains such as drug discovery [31], text classification [19], media retrieval [3] and medical image classification [15]. Active learning is primarily used when little labeled data is available, but unlabeled data is abundant. Its goal is to minimize the human annotation efforts via posing targeted queries to an oracle, instead of labeling the whole dataset. Active learning methods usually select an informative unlabeled instance and ask the oracle for the label of the instance. The oracle, mainly a human oracle, provides the correct label of the instance and then the newly labeled instance is added to the training dataset.

Despite enormous progress in the active learning field in recent years, there are still some shortcomings that need to be addressed. Traditional active learning methods usually ask for the label of a specific unlabeled instance. Though this might result in some accuracy improvement, however it might not be very easy for an oracle to label a very specific case. This can be especially true if the query contains many features, and if those features represent high precision numeric data. Also some features might be irrelevant for a certain query and eliminating those features can result in a shorter and more readable query. This can also prevent the oracle's confusion.

For example in the real world, a medical domain expert prefers to be presented with a generic diabetes query which (1) embodies similar patient cases together, (2) only includes relevant symptoms and (3) involves range values instead of very specific exact values for the lab test results. Such a query will be shorter, less confusing and more intuitive. Besides the domain expert will be able to answer more queries in less time and the learning algorithm can achieve higher accuracy rates by posing fewer queries.

We present a novel method for constructing generic active learning queries based on rule induction. We call our method RIQY, standing for **R**ule **I**nduced active learning **Q**uer**Y** method. Our method exploits the underlying density distribution to find an informative instance as well as its most similar cases. Then by using a rule induction classifier to infer rules for separating those similar cases from the rest of data, we construct a generic query. We evaluate our algorithm on two different sets of data. The first set of data consists of various real world datasets from the UCI repository [11]. The second set of data includes several human activity recognition datasets from the WSU repository [6]. Our results on both sets of data show that our method is able to achieve higher accuracy rates using fewer queries compared to the traditional active learning methods. Our method is also relatively easy to implement as it employs the well known machine learning components.

The remainder of this paper is organized as follows. First we will provide a brief overview of the active learning and rule induction literature in Section 2. Next, in Section 3 we will present a motivational example. In Section 4, we

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

explain our approach and its details. We then present the results of our experiments in Section 5. Finally we bring up our conclusions and discussion of future work in section 6.

2. RELATED WORK

A variety of active learning approaches have been proposed during the past decade [26, 29]. The simplest form of active learning is the uncertainty sampling method which was introduced by Lewis and Gale [18]. Uncertainty sampling first labels all the unlabeled instances using a classifier. Then it chooses the most uncertain instance and asks the oracle for its label. Other researchers have extended this method to multi-class classification problems by using uncertainty measures such as the entropy measure [16]. It should be noted that the uncertainty sampling method can select the outliers, as it ignores the underlying distribution and the outliers as isolated points can be highly uncertain.

Another type of active learning method is the committee based active learning method. It constructs a committee of classifiers and selects an unlabeled example that causes the maximum disagreement among the classifiers [28, 12, 19]. Similar to the committee based method is the co-testing method which trains two classifiers on two uncorrelated views of data [22]. Version space reduction methods also discard areas of the hypothesis space that have no direct effect on the error rate [30].

More recently, density weighted methods have been proposed [33, 23, 32, 27]. Density weighted methods balance the uncertainty of a sample with its representativeness according to the underlying data distribution. These methods ask for the label of an uncertain instance which is also similar to the other unlabeled instances. As density weighted methods sample from the maximal density regions, they perform well with minimally labeled data. They are also better able to deal with the outlier problem [8]. Settles and Craven [27] have shown that if the proximities are pre-computed and cached, then the computational complexity of those methods is no different than the traditional active learning methods.

We use a weighted density method for selecting the initial informative instance. As we are adding many instances at a time, selecting the non-outlier instances can be quite vital to avoid the accumulating errors. Using a density weighted method allows us to avoid selecting such outliers.

There are a number of active learning frameworks that resemble our work. Batch mode active learning [1, 14] generates queries in groups instead of a single instance at a time. However batch mode active learning still requires the oracle to label "all" of the instances in a batch and does not reduce the set of similar queries to a generic query.

The multiple instance active learning method also groups several instances together [7, 27]. The group is labeled negative if the oracle labels all the instances in the group as negative, but it is labeled positive if at least one of the instances is marked positive. Again the oracle has to label all the instances instead of a generic query.

Druck et al [9] proposed a feature labeling method where a single feature is queried for its label. For example, in a baseball vs. hockey text classification problem, the presence of the word "puck" is a strong indicator of hockey. As this method is selecting one feature at a time, it can be said that it is a specific case of our method. Note that we exploit both instances and features.

Du et al. [10] also recently proposed a method for grouping several instances together. Their method does not take into account the actual data distribution as they are grouping synthetic data points together. They randomly generate a fixed number of synthetic data points around an informative instance selected by an uncertainty sampling method. As pointed out by others [17], synthetic data points might not exactly reflect the actual data distribution and taking this approach might in fact result in adding the outliers.

In contrast to the above methods, our RIQY method aggregates multiple actual instances into a single generic query and it takes into account the actual data distribution. It avoids selecting the outliers by employing a density weighted method. Our method is also easy to implement as it uses the well known machine learning components.

We employ rule induction for forming generic active learning queries. Rule based classifiers have a long history and many rule based classifiers have proposed in the literature. A number of such methods include CN2 [4], Ripper [5], AQ [20] and C4.5 decision tree rules [25]. Rule induction classifiers try to identify the sets of highly predictive features as a rule. They usually prune redundant or noisy rules in order to achieve better prediction accuracies. Rules offer the advantage of having a natural expression that is quite easy for humans to understand. We use a C4.5 classifier [25] to induce generic active learning queries based on a number of similar instances.

3. MOTIVATIONAL EXAMPLE

We show by an example how our RIQY method can pose shorter and more meaningful queries to the oracle and how such queries can aggregate multiple similar cases together. Figure 1 better highlights the differences between a traditional active learning method versus our RIQY method.

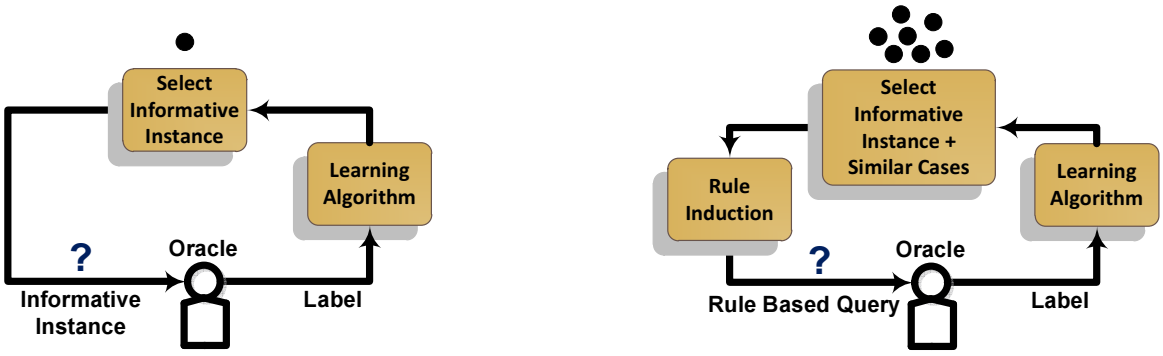
For example consider a heart disease dataset. We assume that it has 20+ features and the classification task is to predict whether a patient has heart disease or not. An example query will be as following.

"*What is the class label if (sex= female) and (age =39) and (chest pain type = 3) and (serum cholesterol = 150.2 mg/dL) and (fasting blood sugar = 150 mg/dL) ... and (electrocardiographic result = 1) and (maximum heart rate achieved = 126) and (exercise induced angina = 90) and (heart old peak = 2.3) and (number of major vessels colored by fluoroscopy = 3)?*"

Note that for the sake of brevity here we only show a number of features. But in reality all the features are included in the query to be presented to the oracle. From this example query, one can clearly see how such a long and overly specific query with so many features can be difficult to be answered by a domain expert. Besides, a similar case would require another query to be posed to the oracle, without taking advantage of the previous cases. Also many features might be indeed irrelevant to the query at hand.

Although one can apply a feature selection method as a preprocessing step here, but it should be noted that it would discard the features in a global manner. In contrast, we are looking at the problem of discarding a "locally" irrelevant feature for a number of similar instances. For example, "exercise induced angina" feature might be relevant for the whole dataset, but it might not very discriminating for a group of patients whose "age > 65".

In the real world, a domain expert usually expects queries in a shorter and more intuitive form, with range values in-



(a) Traditional active learning.

(b) Our RIQY active learning method.

Figure 1: Traditional active learning methods versus our RIQY active learning method.

stead of exact values and with similar cases aggregated together as a generic case. One such example is:

"*What is the class label if (age > 65) and (chest pain type = 3) and (serum cholesterol > 240 mg/dL) ?*"

In the following sections, we show how we can construct such generic queries as a form of rule induction.

4. MODEL

Our input data is an n-dimensional feature vector which is denoted by $x = \langle x_1, x_2, \cdots, x_n \rangle$. Each labeled instance x is assigned a class $y_j \in y$. The function that measures the informativeness of an instance x is denoted by $\Phi(x)$ and the most informative instance is denoted by x^*. We assume that each dataset is composed of a small labeled dataset $\mathcal{L}$, a large unlabeled dataset $\mathcal{U}$ and a test dataset $\mathcal{T}$ which is set aside for the evaluation purposes only.

We also assume that the oracle is able to answer our queries and to provide us with a label l and a confidence value c. There is a proximity matrix M that shows the similarity between each two data points in our dataset. The proximity matrix is used to identify the nearest neighbors of x^*. All the proximities are cached and pre-computed as mixed Euclidean distances. For a more efficient nearest neighbor search, one can employ methods such as kd-tree [24] or locality sensitive hashing (LSH) [13]. If the original dataset contains a large number of features, one can perform conventional feature selection methods to obtain a more representative dataset. But as we mentioned before, a preprocessing feature selection step is quite different than the local feature selection process during rule induction, as the locally relevant features can vary from query to query.

In summary our method works as following. First we train a classifier $\mathcal{C}$ on a small labeled dataset $\mathcal{L}$ to identify the potential informative instances. This step is similar to most of the traditional active learning methods. Our informativeness measure is a variation of density weighted method that also takes into account the dissimilarity to the previously labeled data in order to achieve a more efficient method.

After identifying the most informative instance, we select its nearest neighbors as well as its enemies. The enemies set is obtained through random sampling of the rest of the unlabeled data. The nearest neighbors and the enemies are combined together into a single dataset. The nearest neighbors are assigned a label of 0, while the enemies are assigned a label of 1, regardless of their original labels. By separating the nearest neighbors from the enemies using a rule induction classifier, we can obtain the essential discriminating features. Rules naturally have an expressive format which is quite suitable for human oracle. Next, we present the induced rules to the oracle and update the labeled dataset. This is repeated until a maximum number of queries is posed or until a specified accuracy improvement is achieved.

Figure 2 shows the main components of our method. In the next subsection, we will explain our method in more detail.

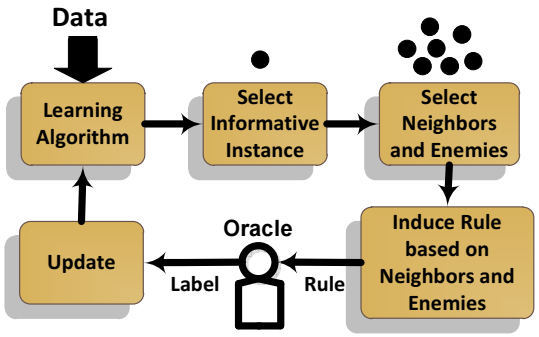


Figure 2: The main components of our method.

4.1 Model Details

The first step in our method is to select the most informative instance x^* among the pool of unlabeled instances. As mentioned before, we use a variation of density weighted method for measuring the informativeness of an instance x. Our measure considers the similarity of x to the other unlabeled instances, as well as its dissimilarity to the previously labeled instances. Considering similarity to the other unlabeled instances allows us to select an instance that is representative of many other similar cases. Considering its dissimilarity to the previously labeled instances allows us to avoid querying similar instances repetitively, thus leading to a more effective method. Equation 1 shows our method for selecting the most informative instance x^*.

$$x^* = \arg\max_x \left[(1-\alpha) * \Phi(x) + \alpha * \frac{|\mathcal{L}| * \sum_{u=1}^{|\mathcal{U}|} M[x, x_u]}{|\mathcal{U}| * \sum_{l=1}^{|\mathcal{L}|} M[x, x_l]} \right] \quad (1)$$

In Equation 1, parameter α balances the contribution of density versus the base informativeness measure $\Phi(x)$. Here the second term shows the contribution of density with respect to both labeled and unlabeled data. The first term, $\Phi(x)$, refers to a base informativeness measure. The base informativeness measure here is an entropy measure as in Equation 2. But it can be replaced by any other base informativeness measure. Here y_i ranges over all possible label values.

$$\Phi(x) = -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x) \quad (2)$$

After an informative instance has been selected, a generic query is constructed based on the informative instance and its similar instances. Algorithm 1 outlines further details of our RIQY method.

Algorithm 1 RIQY Active Learning method

```
1: procedure FindRuleBasedQuery(L, U, M)
2:     while stopping criteria is not met do
3:         Train a classifier C using L
4:         Use C to compute p(y|x) ∀x ∈ U
5:         Find x* using Equation 1
6:         Find N_x*
7:         Sample {U − N_x*} to get E_x*
8:         Δ = N+_x* ⋃ E−_x*
9:         Form rule set R from Δ
10:        Select Ŕ from R based on γ and a
11:        c, l ← askOracle(Ŕ)
12:        for all x_j ∈ Δ do
13:            if x_j is covered by some r_k ∈ Ŕ then
14:                Update w_j as in Equation 3
15:                if w_j > θ then
16:                    Add x_j to L with label l_k
17:                end if
18:            end if
19:        end for
20:    end while
21: end procedure
```

Lines 3−4 trains a classifier $\mathcal{C}$ based on the available training data $\mathcal{L}$, and computes the class probabilities for each unlabeled instance $x \in \mathcal{U}$. In line 5, based on the obtained class probabilities and the underlying distribution, we select the most informative instance x^* according to Equation 1.

Next, we find x^* similar instances and form a generic query based on those instances. We first select the nearest neighbors of x^* in line 6. The size of neighborhood (or the enemy vicinity) is a fixed fraction ϵ of the unlabeled dataset $\mathcal{U}$. In other words, $|\mathcal{N}_{x^*}| = |\mathcal{U}| * \epsilon$. To get the enemies set $\mathcal{E}_{x^*}$, we sample from $\mathcal{U} - \mathcal{N}_{x^*}$. The size of the enemies set is the same as the neighbors set, i.e. $|\mathcal{E}_{x^*}| = |\mathcal{U}| * \epsilon$. We chose them to be of the same size to avoid any class imbalance problems. As over time the unlabeled dataset gets smaller and as more labeled data becomes available, the neighborhood size as well as the enemy vicinity becomes smaller. This allows us to pose more focused queries over time.

Next in line 8, we combine the nearest neighbors $\mathcal{N}_{x^*}$ and the enemies $\mathcal{E}_{x^*}$ into a single set Δ. Each instance of the nearest neighbors is assigned a label of 0 in Δ, while each instance belonging to the enemies is assigned a label of 1 in Δ, regardless of their original labels. We denote the relabeled sets by $\mathcal{N}^+_{x^*}$ and $\mathcal{E}^-_{x^*}$.

Then a rule induction classifier such as C4.5 [25] is used to generate a number of rules $\mathcal{R}$ from Δ in line 9. Through separation of the nearest neighbors of x^* from its enemies by using a rule induction classifier, we can obtain the essential features for discriminating x^* and its similar cases from the rest of data in form of rules. Note that each rule $r \in \mathcal{R}$ is accompanied by its coverage value γ and its accuracy a. The coverage value shows how many instances are covered by a certain rule and the accuracy shows the discriminative power of a rule for distinguishing between positive and negative instances. We select the rules with a minimum accuracy $a_{\min}$ in order to avoid adding the incorrect instances. The rules are then sorted according to their coverage values and the top N rules with the highest coverage value are selected as $\acute{\mathcal{R}}$ in line 10. Note that normally we set N to 1 to present one rule a time to the oracle. But it is also possible to present several rules at a time in a batch mode.

Line 11 poses the rule set $\acute{\mathcal{R}}$ to the oracle and receives the corresponding labels l_i and confidences c_i for each rule $r_i \in \acute{\mathcal{R}}$. Lines 12-19 update each unlabeled instance $x_j \in \Delta$ that is covered by some rule $r_k \in \acute{\mathcal{R}}$, according to Equation 3.

$$w_j = c_k * \frac{1}{|X_{r_k}|} \sum_k^{|X_{r_k}|} M[j,k] \quad (3)$$

Here w_j shows the weight of instance x_j in our dataset. The set of all instances that are covered by a certain rule r_k is denoted by X_{r_k}. The instance's weight increases proportionately with its similarity to the query neighborhood and the oracle's confidence about query's label.

At the end, those instances covered by some rule and with a weight above the weight threshold θ are added to the labeled dataset $\mathcal{L}$. This process continues until we reach the stopping criteria. The stopping criteria can be either reaching a maximum number of queries or reaching a certain minimum classification accuracy improvement.

5. EXPERIMENTS

We perform a number of experiments on two sets of real world data in order to evaluate our RIQY active learning algorithm. The data in our experiments consists of 6 real world datasets from the UCI repository [11] and 6 human activity recognition datasets from the WSU repository [6].

5.1 Experiment Setup

The first set of data includes 6 real world datasets from the UCI repository [11]. These datasets include the Germany credit approval dataset, the Wisconsin breast cancer dataset, the heart disease dataset, the wine dataset, the ionosphere dataset and the chess dataset. All the datasets are binary classification problems, except for the wine dataset. Further details of those datasets is shown in Table 1.

Dataset	Number of Features	Attributes Type	Number of Examples
Credit approval	20	Numeric-Nominal	1000
Ionosphere	34	Numeric	351
Wine	13	Numeric	178
Heart Disease	13	Numeric-Nominal	270
Breast Cancer	9	Nominal	699
Chess	36	Nominal	3196

Table 1: The UCI datasets.

The activity recognition datasets consists of data collected from 6 different smart apartments [6]. We will refer to those datasets as B3, C, K4, M, T1 and T2. Each smart apartment is equipped with different types of sensors such as infrared motion sensors installed on ceilings and walls, water sensors installed on faucets, contact switch sensors installed on doors and cabinets, and item sensors on the key items. All the datasets are collected during normal day-to-day life of residents without any experimental settings. The annotated activities in those datasets are shown in Table 2.

	B3	C	K4	M	T1	T2
Hygiene	✓		✓	✓		
Leave Home	✓	✓	✓	✓	✓	✓
Cook	✓		✓	✓	✓	✓
Relax	✓		✓	✓	✓	✓
Take Med	✓	✓		✓		
Eat	✓	✓	✓	✓	✓	✓
Sleep	✓	✓	✓	✓		
Bathing	✓		✓			
Bed to toilet	✓	✓	✓	✓		
Work	✓	✓	✓	✓		

Table 2: The annotated activities in each dataset.

All the activity datasets represent multi-class classification problems. Each activity instance is represented by features composed of sensor locations and activity duration. Further information about these datasets is shown in Table 3.

We split each dataset into three disjoint parts: a small labeled dataset $\mathcal{L}$ (about 1%-2% of data), a test dataset $\mathcal{T}$ (about 25% of data) and an unlabeled dataset $\mathcal{U}$ (the rest of data). All the results are averaged over 3 runs in order to reduce the experiment variation. The initial classification step is performed using a support vector machine from the LibSVM library [2]. Using a cross validation method, we found the following values for our parameters: $\alpha = 0.5$, $\theta = 0.5$, $a_{\min} = 0.85$ and $\epsilon = 0.2$ for the UCI dataset and $\epsilon = 0.01$ for the WSU activity recognition datasets. A C4.5 decision tree from the RapidMiner tool was used for constructing rules [21]. We set the minimal information gain to 0.1. We confined the depth of the tree to a maximum of 10 attributes to prevent very long queries, though in most cases the generated query is shorter.

To simulate the effect of a human oracle in determining the label of a specific rule, we consider all the instances that are covered by a specific rule. The label of the rule is then reported as the majority label and the confidence is reported as the majority fraction. It should be noted that our datasets as real world datasets contain noisy examples, which can lead to lower confidences. Similarly, the noisy examples can also confuse a domain expert in the real world. Though by infusing many similar cases together into a generic query, the effect of noisy examples is reduced, still in some cases this might lead to a lower confidence.

5.2 Experiment Results

After performing the initial preprocessing steps such as proximity computation, we ran our algorithm on both sets of UCI and WSU datasets.

First we show two example queries from the wine dataset and the B3 activity dataset. In the B3 query, the actual sensor numbers have been replaced by their location.

Wine query "*What is the alcohol type if alcohol percentage is* 12–14 *and Proanthocyanidins is* 0.4–1.2 *and color intensity is* 1.2–7.1 *?*"

B3 query "*What is the activity label if duration is* 5–101 *minutes and* 78%–100% *of the sensors are work area sensors and* 1%–2% *of the activated sensors are bedroom sensors ?*"

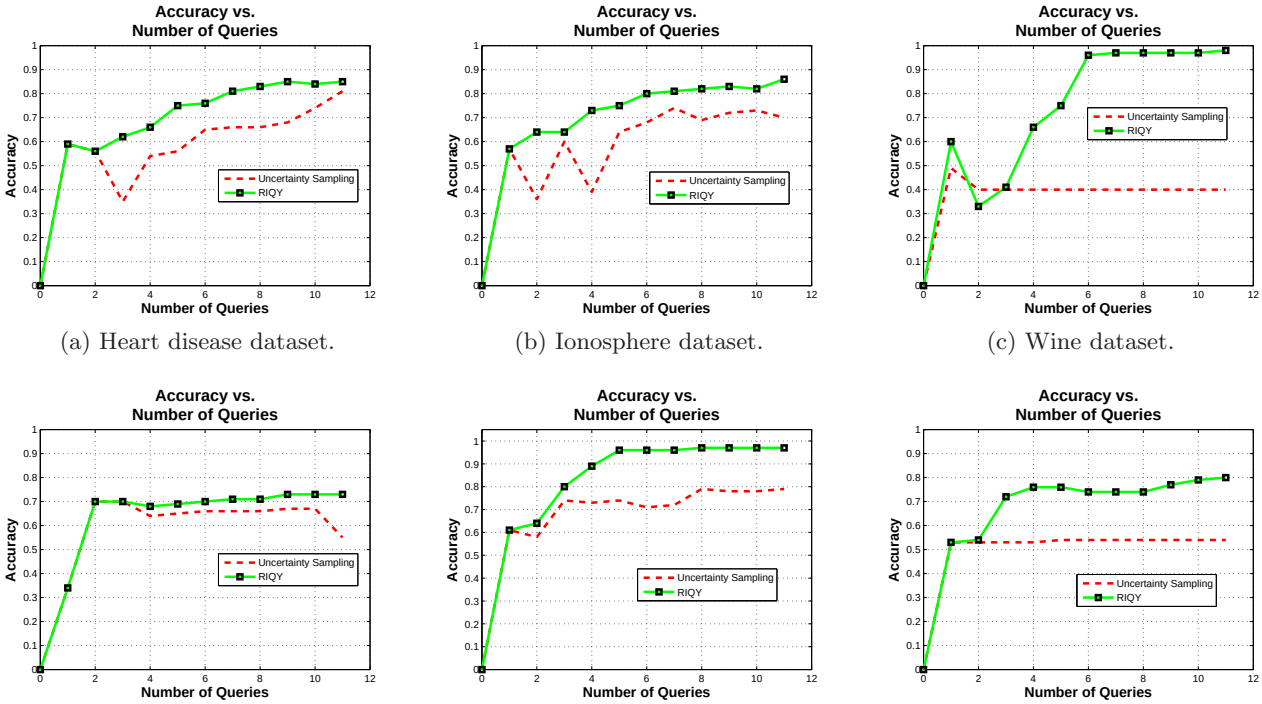
The first query reduces the number of features by 76.9% and it aggregates a total of 13 instances. The second query reduces the number of features by 80% and it aggregates a total of 113 instances. One can clearly see how our method results in shorter and more intuitive queries.

Next, we computed the classification accuracy of our method. We set the number of selected rules N to 1. In order to compare our method to a traditional active learning method, we also performed the same experiments using an uncertainty sampling method. The results of our experiments on UCI datasets can be seen in Figure 3. Similarly, the results on the WSU datasets are shown in Figure 4. We can see that our method outperforms the uncertainty sampling method in most cases by reaching higher accuracy rates with fewer queries. This can be attributed to utilizing the actual underlying distribution of data and avoiding querying the outliers.

It should be noted that the results of applying our RIQY method might vary from dataset to dataset, depending on the type of data and whether it can be generalized easily or not. Also the noisy instances and the regularity of the dataset can play a role in determining the classification accuracy. For example, dataset T2 represents a smart apartment

Dataset	Number of Features	Number of Activities	Number of Examples	Data Type
B3	15	10	3361	Numeric
C	10	6	511	Numeric
K4	14	9	746	Numeric
M	13	9	2270	Numeric
T1	10	4	1431	Numeric
T2	11	4	163	Numeric

Table 3: The WSU datasets.



(a) Heart disease dataset. (b) Ionosphere dataset. (c) Wine dataset.

(d) German credit approval dataset. (e) Wisconsin breast cancer dataset. (f) Chess dataset.

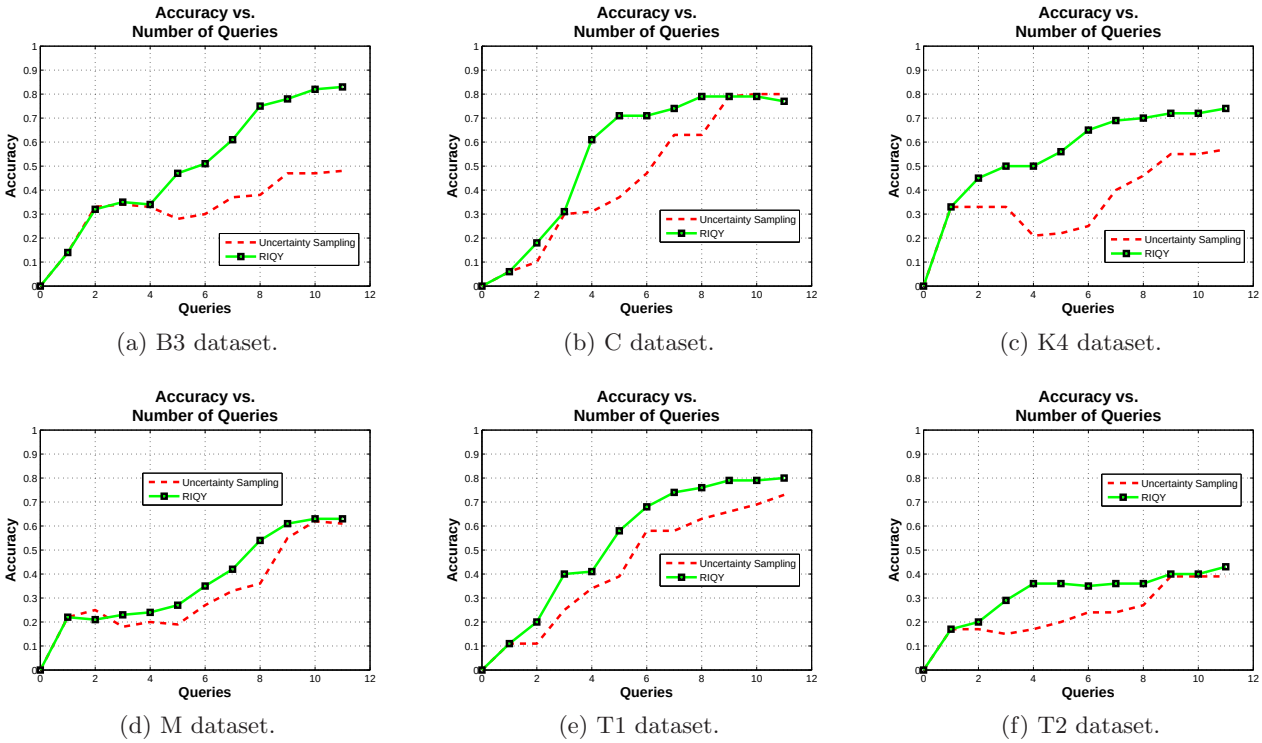
Figure 3: Accuracy for different UCI datasets vs. the number of queries.

where the residents did not have a regular schedule. Therefore, the algorithm is not able to take much advantage of the similarities between various instances of the same activity. Despite all these, we still can see that in most cases our method offers a higher accuracy rate by using fewer and shorter queries.

Table 4 and Table 5 show the average feature reduction, the average oracle confidence and the average number of added instances for the UCI and WSU datasets. Note that the average number of added instances is 1 for the uncertainty sampling method. From these tables we can see how in average the number of features is reduced by our RIQY method. This makes it possible to pose shorter queries.

In another experiment, we explored the effect of increasing the number of posed rules (N) to the oracle. Figure 5 shows the results of our experiment on two example datasets: the Wisconsin breast cancer dataset and the T2 dataset. We can see that increasing the number of posed rules to the oracle can lead to higher accuracy rates. Note that as we add the rules in the order of descending coverage, posing the last rule might not have the same effect as posing the first rule. Also though increasing the number of posed rules to the oracle increases the total number of queries, however many irrelevant features are not included in each rule. Therefore despite posing more queries, the total number of features to be handled by the oracle is still small.

In summary, the above results show how we can construct more generic active learning queries based on rule induction. Our results confirm that such a method can lead to higher classification accuracy rates with fewer and shorter queries.



(a) B3 dataset. (b) C dataset. (c) K4 dataset.

(d) M dataset. (e) T1 dataset. (f) T2 dataset.

Figure 4: Accuracy for different WSU datasets vs. the number of queries.

Dataset	Avg. Features Reduction	Avg. Oracle Confidence	Avg. Added Instances
Credit approval	84.5%	74.4%	51
Ionosphere	90.8%	76.4%	23
Wine	77.6%	75.0%	10
Heart Disease	76.9%	77.6%	14
Breast Cancer	68.8%%	94.5%	34
Chess	88.6%	88.0%	200

Table 4: Some statistics based on our RIQY active learning method for UCI datasets.

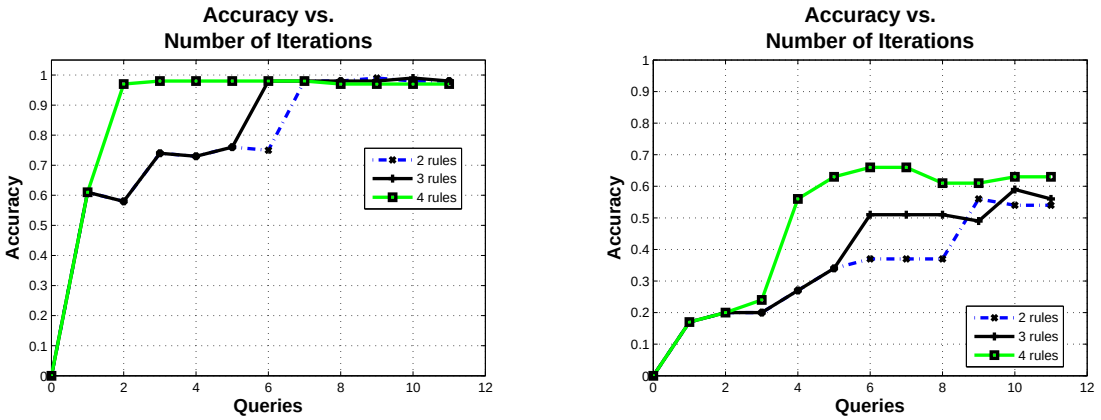
Dataset	Avg. Features Reduction	Avg. Oracle Confidence	Avg. Added Instances
B3	88.0%	64.6%	66
C	85.0%	74.9%	12
K4	86.0%	71.1%	16
M	81.0%	64.9%	44
T1	73.6%%	88.9%	40
T2	90.3%	46.4%	2

Table 5: Some statistics based on our RIQY active learning method for WSU datasets.

6. CONCLUSION AND FUTURE WORK

We showed a method called RIQY for constructing generic active learning queries based on rule induction. RIQY is able to construct shorter and more intuitive queries that are easier for a human oracle to answer, allowing us to better utilize our human resources. Our method also allows the learning algorithm to achieve a higher accuracy rate using fewer queries. RIQY is easy to implement as it employs the well known machine learning components.

In the future, we intend to employ our method on more sophisticated data types, such as graphs and sequences. We also intend to explore the effects of a noisy oracle and to pro-



(a) Wisconsin breast cancer dataset.

(b) T2 dataset.

Figure 5: Accuracy vs. number of posed rules.

pose methods for preventing the resulting accuracy degradation. It is also part of our future plan to perform actual user studies with human oracles to further explore the advantages of using our method over similar traditional active learning methods.

7. REFERENCES

[1] K. Brinker. Incorporating diversity in active learning with support vector machines. In *International Conference on Machine Learning*, pages 59–66, 2003.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] M.-y. Chen, M. Christel, A. Hauptmann, and H. Wactlar. Putting active learning into multimedia applications: dynamic definition and refinement of concept classifiers. In *13th annual ACM international conference on Multimedia*, MultiMedia 2005, pages 902–911, New York, NY, USA, 2005. ACM.

[4] P. Clark and T. Niblett. The cn2 induction algorithm. *Mach. Learn.*, 3:261–283, March 1989.

[5] W. W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning*, ICML'95, pages 115–123, 1995.

[6] D. Cook, L. Holder, B. Shirazi, and M. Schmitter-Edgecombe. *WSU CASAS Smart Home Project*, 2010. At `http://ailab.eecs.wsu.edu/casas/datasets.html`.

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, January 1997.

[8] P. Donmez, J. G. Carbonell, and P. N. Bennett. Dual strategy active learning. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 116–127, Berlin, Heidelberg, 2007. Springer-Verlag.

[9] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 595–602, New York, NY, USA, 2008. ACM.

[10] J. Du and C. X. Ling. Asking generalized queries to domain experts to improve learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:812–825, 2010.

[11] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[12] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28:133–168, September 1997.

[13] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[14] Y. Guo and D. Schuurmans. Discriminative Batch Mode Active Learning. In *Advances in Neural Information Processing Systems*, volume 20, pages 593–600, 2008.

[15] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *23rd international conference on Machine learning*, ICML 2006, pages 417–424, New York, NY, USA, 2006. ACM.

[16] R. Hwa. Sample selection for statistical parsing. *Computational Linguistics*, 30:253–276, September 2004.

[17] K. Lang. NewsWeeder: learning to filter netnews. In *International Conference on Machine Learning*, pages 331–339, 1995.

[18] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Information Retrieval*, pages 3–12, 1994.

[19] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *International Conference on Machine Learning*, ICML 1998, pages 350–358, 1998.

[20] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system aq15

and its testing application to three medical domains. In *AAAI'86*, pages 1041–1047, 1986.

[21] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In L. Ungar, M. Craven, D. Gunopulos, and T. Eliassi-Rad, editors, *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, August 2006. ACM.

[22] I. Muslea, S. Minton, and C. A. Knoblock. Selective sampling with redundant views. In *17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, AAAI 2000, pages 621–626. AAAI Press, 2000.

[23] H. T. Nguyen and A. Smeulders. Active learning using preclustering. In *International Conference on Machine Learning*, pages 79–89, 2004.

[24] R. Panigrahy. An improved algorithm finding nearest neighbor using kd-trees. In *Proceedings of the 8th Latin American conference on Theoretical informatics*, LATIN'08, pages 387–398, Berlin, Heidelberg, 2008. Springer-Verlag.

[25] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[26] B. Settles. Active Learning Literature Survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

[27] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Empirical Methods in Natural Language Processing*, pages 1070–1079, 2008.

[28] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 287–294, New York, NY, USA, 1992. ACM.

[29] K. Tomanek and F. Olsson. A web survey on the use of active learning to support annotation of text data. In *Workshop on Active Learning for Natural Language Processing*, NAACL HLT 2009, pages 45–48, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[30] S. Tong and D. Koller. Support vector machine active learning with application sto text classification. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 999–1006, 2000.

[31] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen. Active learning with support vector machines in the drug discovery process. *Journal of Chemical Information and Computer Sciences*, 43, 2003.

[32] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *European conference on IR research*, ECIR'07, pages 246–257, 2007.

[33] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *European Conference on IR research*, ECIR'03, pages 393–407, 2003.