Hierarchical Active Learning with Proportion Feedback on Regions

Zhipeng Luo(✉) and Milos Hauskrecht

Department of Computer Science, University of Pittsburgh,
Pittsburgh, PA 15260, USA
{ZHL78,milos}@pitt.edu

Abstract. Learning of classification models in practice often relies on human annotation effort in which humans assign class labels to data *instances*. As this process can be very time-consuming and costly, finding effective ways to reduce the annotation cost becomes critical for building such models. To solve this problem, instead of soliciting instance-based annotation we explore *region*-based annotation as the feedback. A region is defined as a hyper-cubic subspace of the input feature space and it covers a subpopulation of data instances that fall into this region. Each region is labeled with a number in [0, 1] (in binary classification setting), representing a human estimate of the positive (or negative) class proportion in the subpopulation. To learn a classifier from region-based feedback we develop an active learning framework that hierarchically divides the input space into smaller and smaller regions. In each iteration we split the region with the highest potential to improve the classification models. This iterative process allows us to gradually learn more refined classification models from more specific regions with more accurate proportions. Through experiments on numerous datasets we demonstrate that our approach offers a new and promising active learning direction that can outperform existing active learning approaches especially in situations when labeling budget is limited and small. Code related to this paper is available at: https://github.com/patrick-luo/hierarchical-active-learning.git.

Keywords: Active learning · Proportion label · Classification

1 Introduction

Learning of classification models from real-world data often requires non-trivial human annotation effort on labeling data instances. As this annotation process is often time-consuming and costly, the key challenge then is to find effective ways to reduce the annotation effort while guaranteeing that models built from the limited feedback are accurate enough to be applied in practice. One popular machine learning solution to address the annotation problem is active learning. It aims to sequentially select examples to be labeled next by evaluating the possible

M. Berlingerio et al. (Eds.): ECML PKDD 2018, LNAI 11052, pp. 464–480, 2019.
https://doi.org/10.1007/978-3-030-10928-8_28

impact of the examples on the solution. Active learning has been successfully applied in domains as diverse as computer vision, natural language processing and bio-medical data mining [8,14,15].

Despite enormous progress in active learning research in recent years, the majority of current active learning solutions focus on *instance-based* methods that query and label individual data instances. Unfortunately, this may limit its applicability when targeting complex real-world classification tasks. There are two reasons for this. First, when the labeling budget is severely restricted, a small number of labeled data may not properly cover or represent the entire input space. In other words, the data selected by active learning are likely to suffer from *sampling bias* problem. To mitigate this issue Dasgupta [2] has developed a hierarchical active learning approach to sample instances in a more robust way which is driven by not only the current sampled data, but also the underlying structure in the data.

Second, instance-based learning framework often assumes instances are easy to label for humans. But it is not always true. Consider two realms of applications: (1) in political elections where the privacy is a concern, collecting one's feedback is hard or infeasible [9,10,16]; (2) in medical domain patient records can be very complex as each record has numerous entries which require careful reviewing [3,11]. For example, when a physician diagnoses a patient (e.g. for possible heart condition) he/she must review the patient record that consists of complex collections of results, symptoms and findings (such as *age, BMI, glucose levels, HbA1c blood test, blood pressure, etc.*). The review and the assessment of these records w.r.t. a specific condition may become extremely time-consuming as it often requires physicians to peruse through a large quantity of data [4,5].

In light of this, novel active learning methods based on *group* queries have been proposed: AGQ+ [3], RIQY [11] and HALG [7]. The basic idea here is to (1) embody similar instances together as a *group*, (2) induce the most compact *region* which are conjunctive patterns of the input feature space to represent the group and (3) solicit a *generic label* on the region instead of on any specific instance. The region label is a number in [0, 1] (known as *proportion label* [6,9,10,16]) which represents a human estimate of the proportion of instances in *positive* or *negative* class in the subpopulation of instances in that region. This line of work has shown empirically that active learning with proportion feedback on generic regions works more efficiently than instance-based active learning.

Our Contribution. In this work, we develop and explore a new region-based active learning framework called HALR (**H**ierarchical **A**ctive **L**earning with proportion feedback on **R**egions) that learns instance-level classifiers from region queries and region-proportion feedback. In particular, our framework *actively* builds a hierarchical tree of regions with the aim to refine the leaf regions to be as *pure* as possible after very *few* splits and queries made. Briefly, our method starts from an unbounded region that covers the entire input feature space and this region initializes as the root of the tree. Then we grow this tree incrementally by splitting the most *uncertain* leaf region into two sub-regions. Whenever the new regions are generated, their proportion labels are either directly assigned

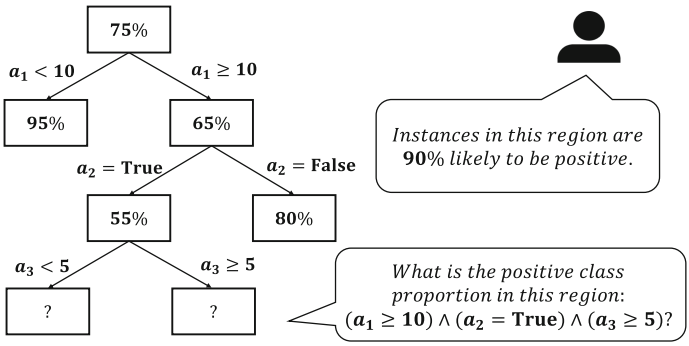

Fig. 1. An example of building a hierarchical tree of regions which is conceptually equivalent to a decision tree. The left shows a snapshot of the tree structure after $t = 3$ splits, generated from the root region on the top level. Each rectangle represents a certain region and the percentage number means its proportion label. Each link is a value constraint on some dimension a_i and it is inherited to all the descendant regions. To query the proportion label of a new region (say the right one on the lowest level), we describe it by using conjunctive patterns shown on the bottom right and a human annotator will assign a label to it according to its description. The label of the complementary region (the one on the left) will be inferred according to the constraint between its parent's label and sibling's label.

by a human annotator, or inferred by the proportion constraint. The general picture is illustrated in Fig. 1. At the end our algorithm outputs a hierarchical tree of labeled regions that can be either (1) directly used as a decision tree classifier, or alternatively, (2) be used to learn many different parametric binary classification models from proportion labels as proposed by [6,9,10,16], or by simply sampling instance labels [7] according to the known class proportion in each region and feeding them to standard instance-level learning algorithms.

The crucial part of our algorithm is to develop a strategy to split the leaf regions *without* knowing any labeled instances. To meet this challenge we design a competition procedure which dynamically tests and chooses one of two heuristic strategies to split the regions. The first one is *unsupervised* which is based on clustering. The second one is *supervised* and it relies on classification model that assigns class probabilities to every data instance. We will show that these heuristics can actively compete and also assist each other to drive our splits.

The remainder of the paper is organized as follows. First, we will review past work closely related to our framework. Second, we will explain the details of our proposed framework from Sects. 3 to 7. After that we will test our approach on a number datasets and compare its performance to multiple other active learning approaches. Finally, we will discuss the experiment results.

2 Related Work

2.1 Hierarchical Active Learning

Our hierarchical learning framework is motivated by Dasgupta *et al.*'s work [2, 13] that leverages a pre-compiled hierarchical clustering to drive the instance selection procedure. They start learning from a few coarse-grained clusters and gradually split clusters that are impure (in terms of class labels) to smaller ones such that the label entropy is reduced. In terms of training models, not only the labeled instances but also the ones with predicted labels in the *sufficiently pure* clusters are used for learning. While their approach is able to reduce the sampling bias, learning with predicted labeled data can be risky especially when the class distribution is severely unbalanced, as the instances from the minor class are hardly sampled. In our work we overcome this limitation by directly querying and learning from regions of which the proportion labels are friendlier to the minor class. Another difference worth noting is that we do not pre-compile a hierarchy of regions which can be done totally unsupervisedly (e.g. build a K-Dimension tree beforehand). Instead, we build the tree dynamically where each of the splits is determined by not only the unsupervised heuristic but also a supervised heuristic which reflects the current belief of the base model.

2.2 Learning from Group Proportion Feedback

Multiple works [6,9,10,16] study the problem of learning instance-level classifiers from apriori given groups/regions and their class proportion labels. The motivation scenarios can be political election, online purchasing or spam filtering. For example, we can easily obtain the percentage of voting results on election in each county and use these group proportions to predict individual's voting preference. These real life examples have greatly encouraged the development of learning algorithms that can eat proportion feedback. There are two main categories of the algorithms. The first one uses the proportion label as a proxy that approximates to the sufficient statistics required by the final likelihood function [9,10]. The second category develops models that generate consistent instance labels with the group proportions [6,16]. What beyond the scope of the above works is that they assume the groups are formed and labeled *apriori*, and thus they do not study the problem of how to form the groups and how to obtain the proportion labels for these groups.

2.3 Active Learning from Group Proportion Feedback

AGQ+ [3] and RIQY [11] are the early works that explore active learning strategies with group/region proportion feedback instead of instance-based feedback. The motivation for the group queries is that in many practical domains, annotators may prefer to work with region-based queries which are shorter (in terms of feature space), less confusing and more intuitive. As an example consider the heart disease classification task presented in [11]:

An Instance Query Example. An instance query for the heart disease problem covers all features of the patient case: *"Consider a patient with (sex = female) ∧ (age = 39) ∧ (chest pain type = 3) ∧ (fasting blood sugar = 150 mg/dL) ... (20 more features omitted). Does the patient has a heart disease?"* The label is a binary (true, false) response.

A Group Query Example. In contrast, a group query using conjunctive patterns which represent a region of the input feature space may be only associated with a subset of the features: *"Consider a population of patients with (sex = female) ∧ (40 < age < 50) ∧ (chest pain type = 3) ∧ (fasting blood sugar within [130,150] mg/dL) ... (not necessarily using all the features). What is the chance that a patient from this population has a heart disease?"*. The label is an empirical estimate of the proportion of cases in the population who suffer from the heart disease, say *"75% patients within this region suffer from the disease"*.

In terms of group formation, both AGQ+ and RIQY build groups by (1) choosing the most uncertain instance $\boldsymbol{x}_u$ from the unlabeled data pool according to the current classification model, and (2) aggregating a number of instances as a group G_u in a close neighborhood of $\boldsymbol{x}_u$. The region description of the group G_u is then automatically learned using decision tree algorithm. After the proportion label of the group G_u is annotated, all the instances inside the group G_u are either assigned hard labels (RIQY) or weighted labels (AGQ+). Finally the classification model is re-trained using all the labeled data. The major limitation of the methods is that their group selection approach is ad-hoc, driven by instance-based selection and enriched by nearby data instances. As a consequence, this approach may fail to discover meaningful regions.

A more recent approach that addresses some limitations of the early group active-learning methods is HALG [7]. HALG uses a hierarchical clustering, similarly to Dasgupta *et al*'s work, to generate clusters of instances which are then approximated by regions. As this hierarchy of regions is pre-clustered, their active learning algorithm, which selects groups/regions to be split and labeled next, can only make decisions within this fixed hierarchy. While this novel group formation approach is able to capture the structure of the unlabeled data (unsupervised heuristic), the fixed hierarchy can significantly limit the behavior of seeking the class information which is important to the model (supervised heuristic). That is, the unsupervised heuristic used in HALG overly dominates its supervised heuristic. To overcome this issue, our proposed HALR method *dynamically* refines regions by *directly* dividing the input feature space into sub-spaces (still in a hierarchical fashion) and further, our active region refinement is explicitly controlled and balanced between the supervised and unsupervised heuristics.

3 Our Framework

Our HALR framework is summarized in Algorithm 1. It aims to actively build a hierarchical tree of regions with proportion labels and then uses this tree to learn an instance-level binary classification model. We assume the classification model is a probabilistic one (e.g. Logistic Regression or an Support Vector Machine

Algorithm 1. Hierarchical Active Learning Framework (HALR)

Input: An unlabeled data pool $\mathcal{U}$; A labeling budget
Output: A binary classification model $P(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}})$

1: $T \leftarrow$ Build a 1-node tree whose root region is the entire feature space of $\mathcal{U}$;
2: Query the proportion label of T's root;
3: Leaf nodes $L^{(1)} \leftarrow \{T$'s root$\}$;
4: Active learning time $t \leftarrow 1$;
5: **repeat**
6: Train the base model $P(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}^{(t)})$ with current leaf nodes $L^{(t)}$;
7: Choose a most *uncertain* region R_* in $L^{(t)}$ to be split;
8: Divide R_* into two sub-regions (it is co-decided by probabilistic clustering and probabilistic classification (based on $P(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}^{(t)})$);
9: Query or infer the proportion labels of the sub-regions derived from R_*;
10: $L^{(t+1)} \leftarrow \{L^{(t)} - R_*\} \cup \{R'_* s$ sub-regions$\}$;
11: $t \leftarrow t + 1$
12: **until** the labeling budget runs out
13: **return** $P(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}^{(t)})$

with Platt's transformation). Such a model is treated as our base model which will be used to provide supervised heuristic and decisions to guide the tree-building process. Our algorithm works as follows. The tree is initialized with a root region covering the entire input space and as well as all the unlabeled data $\mathcal{U}$ (line 1). The root region is assigned a proportion label which can be interpreted as the *prior* probability of classes (line 2). The tree is gradually refined through active learning cycles (Line 5–12) which iteratively replace leaf regions with more refined sub-regions. In each cycle, we (1) select the most *uncertain* leaf region R_* to split; (2) divide it two sub-regions using a condition that placed on one the input dimension; (3) query or infer the proportion labels of the new sub-regions and (4) replace R_* with the new sub-regions in the tree. Every time the new regions are generated and labeled, the base classification model will be re-learned with all the labeled leaf regions. The whole process resembles decision tree learning algorithm, but in our case we do not have any labeled instances to drive the splits. In the following we will define region concept (Sect. 4) and uncertainty of regions (Sect. 5) and then explain how we split the most uncertain region (Sect. 6).

4 The Concept of Regions

Our base learning task is to learn a binary classification model and our active learning scenario is a pool-based one [12] which assumes the unlabeled data are abundant. That is, a pool of n unlabeled training instances $\mathcal{U}$ are randomly drawn from a fixed marginal distribution $p(\boldsymbol{x})$ of an unknown joint distribution of $p(\boldsymbol{x}, y)$. Each instance $\boldsymbol{x}$ is a vector of d features, each of which can be symbolic or numeric. So the input feature space is a d-dimensional one where each

dimension is either discrete or continuous and the domain depends on the natural definition of that feature. $\boldsymbol{x}$ also has a binary class label $y \in \{0, 1\}$ which is never queried individually. In our framework, however, the class information is given only on aggregated instances which are described as regions. Initially, there is only region that is defined as the entire feature space of $\mathcal{U}$. Because there is no value constraint on any of the dimensions, this first region is unbounded and it conceptually contains all the instances from $\mathcal{U}$. When a binary split is made on some value v from some dimension a, there will be two sub-regions generated with one value constraint on the dimension a either $<v$ or $\geq v$. This type of binary splits will recursively divide the sub-regions and in the end a hierarchical tree of regions will be generated where the leaf regions do not overlap with each other but co-partition the whole feature space and data in $\mathcal{U}$. Each region is thus a hyper-cubic subspace defined by conjunctive patterns. For example a region of patients may be described as: *(gender = male)* $\wedge$ *(heart rate 80–100)* $\wedge$ *(temperature 100–110 F)...(other dimensions unbounded).*

In terms of the region feedback, the human assessment is made via a proportion label which is an estimate of the proportion of the *positive* or *negative* class in the population of instances that fall into the definition of that region. For example, given the region of patients described above, physicians could say *"70% of patients in the population defined by a region suffer from a heart disease"*. Or alternatively, we can interpret the proportion label as an instance-level likelihood: *"Each patient in the population is 70% likely to have a heart disease"*. Initially, the root region is assigned a proportion label which corresponds to the *prior* probability of classes. So in this sense, the proportion label of each subregion can be understood as a *conditional* probability of classes given the value constraints on some of the input dimensions.

5 The Uncertainty of Regions

Given the definition of regions we now want to define a score that would help us to decide which region should be split next in each active learning cycle. One sensible way is to use the uncertainty (or impurity) of regions. This idea has been successfully used in decision tree learning process. Here, the impurity is measured in terms of the entropy (C4.5) or the Gini-Index (CART) scores. With the help of the impurity measure one can build a decision tree recursively where in each step one leaf region is split along one of the input dimensions. By comparing all possible splits for all eligible leaf regions, the best region and the best split that leads to the maximum reduction in uncertainty, or the maximum information gain, can be identified. Unfortunately, this process applied in the decision tree learning to assess uncertainty and gain requires instance labels and hence, it cannot be replicated in our framework where instance labels are unknown.

Another issue to consider in the development of the region splitting criteria is that the information gain ignores the region size. Here the region size is defined as the empirical number of instances contained in a region. Intuitively, the largest benefit from the split should be realized when not only the impure regions but

also large regions are split. In light of this, we propose a new *uncertainty* score that takes into account both the size and the proportion label in deciding which region should be split next.

Suppose that at time t there are $N^{(t)}$ leaf regions $L^{(t)} = \{(R_i, \mu_i)\}_{i=1}^{N^{(t)}}$ where each region $R_i = \{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}$ has n_i instances and has been assigned a label $\mu_i \in [0, 1]$ representing the *positive* class proportion, our goal is to choose the most uncertain region R_* to split. The uncertainty of each region R_i is defined as the expected number of wrong labels (denoted by w_i) if we randomly guess the class labels of all instances in R_i based on its proportion label μ_i. In particular, the procedure to calculate uncertainty is explained as follows:

i. For each instance in R_i, sample its label as an independent Bernoulli process with the parameter $= \mu_i$. This creates n_i sampled labels;
ii. Calculate the distribution of w_i, i.e. the number of mismatches between the sampled labels and the true labels. Although the true labels are unknown, each true label can be assumed to follow an independent Bernoulli distribution with the parameter $= \mu_i$. Therefore, the probability of mismatch for each instance also follows in independent Bernoulli distribution with parameter $= P(mismatch) = P[false\ positive] + P[false\ negative] = 2\mu_i(1-\mu_i)$. Then apparently w_i follows a Binomial distribution $Bin(n_i, 2\mu_i(1-\mu_i))$;
iii. And use the expectation $\mathbb{E}(w_i) = 2\mu_i(1-\mu_i)n_i$ as the uncertainty of R_i.

This uncertainty defined above clearly shows that larger n_i or more uncertain μ_i (closer to 0.5) leads to more uncertainty of region R_i. Please note here $2\mu_i(1-\mu_i)$ matches exactly the definition of Gini-Index, so throughout this paper we will choose Gini-Index as the gain measurement for later use. Finally we select $R_* = \arg\max_{R_i \in L^{(t)}} \mathbb{E}(w_i)$ to be the most uncertain region to split at current active learning cycle t.

6 The Split of Regions

Now given the region R_*, we need to determine what input dimension to split and what value should be used to define the split. Since there are no labeled instances in our framework, we resort to two heuristics to drive the split.

6.1 Unsupervised Heuristic

The first heuristic is *unsupervised*. It is based on probabilistic clustering. Clustering is a simple yet often effective guidance. The assumption behind it is that similar data instances tend to carry similar class labels and it has been used frequently in semi-supervised learning [17]. In other words, dissimilar data are likely to fall into different classes and so the region splits should be driven by the underlying structure of data. To implement this idea, we perform a 2-means probabilistic clustering on the instances $\{\boldsymbol{x}_{*j}\}_{j=1}^{n_*}$ in R_*, assuming there is mix of two cluster centers in $\{\boldsymbol{x}_{*j}\}$ and the probabilities of cluster membership are given by Expectation and Maximization (EM) algorithm. Thus each instance

$\boldsymbol{x}_{*j}$ will have an **U**nsupervised probabilistic label p_j^U indicating the chance of belonging to one of the two clusters. Given these instance-level labels, standard decision tree splitting procedure based on information gain can be now directly applied to split R_*. Here we use Gini-Index and say this procedure gives us the empirically optimal split of R_* from value v^U on dimension a^U based on the set of probabilistic unsupervised labels $\{p_j^U\}$.

6.2 Supervised Heuristic

Our second heuristic is *supervised* and it relies on the base classification model. In various active learning algorithms the base model plays an important role in determining which data should be queried next. An example is the classic Uncertainty Sampling approach [12]. The base model reflects the current belief of the class distribution on instances and thus its guidance on the region splitting cannot be ignored. Formally at learning time t, the base model is learned as $P(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}^{(t)})$, so each instance $\boldsymbol{x}_{*j}$ will also have a **S**upervised probabilistic label p_j^S reflecting the likelihood of belonging to one of the two classes. Here $p_j^S = P(y = 1|\boldsymbol{x}_{*j}; \hat{\boldsymbol{\theta}}^{(t)})$. Similarly, given these instance-level labels Gini-Index-based gain can again be applied to split R_* and say it gives the best split from value v^S on dimension a^S.

6.3 Combination of the Two Heuristics

Table 1 summarizes the pros and cons of the two heuristics. Initially when the supervision is scarce, the base model trained can be very likely to make biased decisions. This problem was formally stated as *sampling bias* by Dasgupta *et al.* [2] and they leverage hierarchical clustering to assist the base model. In our framework we use clustering too as an unsupervised heuristic to alleviate the bias issue. However, the unsupervised heuristic may not always work well in the long run. Hence the best option appears to be the combination of the two heuristics.

Table 1. Comparison of the two heuristics

	Unsupervised heuristic	Supervised heuristic
Pros	Relies on the semi-supervised assumption which is often effective	Gives instance-level estimates which directly reflect the class distribution
Cons	But this assumption may not hold all the time	But initially these estimates are poor simply because the supervision is little

To combine and also to evaluate the two heuristics, we introduce a competition procedure described in Algorithm 2. The general idea is to perform a test

split on each of the proposed splits separately and compare their actual gains. Larger gain is better and so the final split will take whatever the corresponding heuristic suggests. We also maintain a list H that records the winning history of the heuristics in the past splits and this H will be used to test whether the supervised heuristic is doing significant better than the unsupervised one in the long run. If the test result is significant, it marks that our base model is good enough to make splitting decisions alone and from then on, every region split will only be determined by the supervised heuristic. That is, Algorithm 2 will *not* be called any more once we believe the supervised heuristic is performing significantly better and the final split will directly take the supervised proposal.

Algorithm 2. The competition procedure of choosing heuristic

Input: Unsupervised split (a_U, v_U); Supervised split (a_S, v_S); Winning history of heuristics H
Output: The final split (a_F, v_F); updated history H; Binomial test result of supervised heuristic

1: Binomial test result $r \leftarrow$ *Not significant*
2: **if** $a_U = a_S$ and $v_U = v_S$ **then**
3: $a_F \leftarrow a_S$; $v_F \leftarrow v_S$;
4: **else**
5: Do a test split on (a_U, v_U) and get its gain G_U;
6: Do a test split on (a_S, v_S) and get its gain G_S;
7: **if** $G_U > G_S$ **then**
8: Append *"Unsupervised heuristic wins"* to H;
9: $a_F \leftarrow a_U$; $v_F \leftarrow v_U$;
10: **else**
11: Append *"Supervised heuristic wins"* to H;
12: $a_F \leftarrow a_S$; $v_F \leftarrow v_S$;
13: Test result $r \leftarrow$ Binomial test (Algorithm 3) on H;
14: **end if**
15: **end if**
16: **return** (a_F, v_F), H and r

Test Split. The test split and the calculation of the gain procedure called in Line 5 or 6 in Algorithm 2 is identical to the evaluation of a standard decision tree splitting. Here we show how to calculate the gain G_S of the test split on R_* proposed by the supervised heuristic. The gain of G_U can be calculated similarly.

i. Split R_* from value v_S on dimension a_S into two sub-regions R^L and R^R;
ii. Route each instance in R_* to R^L or R^R by testing the feature value of the instance on dimension a_S either $< v_S$ or $\geq v_S$;
iii. Query the proportion label of one sub-region. Say R^L is given a label μ^L;
iv. Infer the label μ^R of R^R. This does not require a human assessment because of the proportion label constraint: $n^L\mu^L + n^R\mu^R = n_*\mu_*$ with $n^L + n^R = n_*$,

where n^L, n^R and n_* are the number of instances contained in R^L, R^R and R_*, and μ_* is the label of R_*. Simply $\mu_R = (n_*\mu_* - n^L\mu^L)/n^R$;

v. Apply Gini-Index to calculate the gain (or uncertainty reduction):

$$G_S = GI(\mu_*) - \frac{n^L}{n_*}GI(\mu^L) - \frac{n^R}{n_*}GI(\mu^R)$$

where $GI(\mu) = 2\mu(1-\mu)$.

Algorithm 3. Binomial test of the supervised heuristic

Input: Winning history of heuristics H; Window size W; Significance level α
Output: *Significant* or *Not significant*

```
if length(H) < W then
    return Not significant
end if
H_0: winning chance of supervised heuristic p_S ≤ 0.5 in the last W trials in H;
H_A: p_S > 0.5
Test statistic B* ← number of supervised wins in the last W outcomes;
p_value ← do binomial test on B*;
return Significant if p_value < α else Not significant
```

Binomial Test. Algorithm 3 provides the details of the Binomial test that decides whether the supervised heuristic is doing significantly better than the unsupervised one. The null hypothesis H_0 means the supervised heuristic is doing equally well or worse than the unsupervised heuristic in the latest W trials. In other words, the winning chance of the supervised heuristic p_S is $\leq$0.5. Under H_0 the number of supervised wins B^* follows a Binomial distribution $Bin(W, 0.5)$ and we do a right-tailed test of B^* to carry out the p-value. We reject H_0 if the p-value is less than a given confidence level α and choose the alternative.

To make the test stronger, or to be more conservative, multiple such tests with different window sizes can be done simultaneously. To ensure the same family wise error rate α, Bonferroni correction can be applied. In our implementation, we combine a short term window $W_S = 5$ and a long term window $W_L = 10$ with the same family wise $\alpha = 0.05$. The purpose of performing two tests together is to ensure the supervised heuristic is indeed doing stably well both in the most recent time and in the long run.

7 Learning a Model from Labeled Regions

Now the last remaining question is how to learn a general instance-level model from labeled regions. As introduced in Related Work section, various algorithms can be applied to learn instance-level classification models from proportion

labels [6,9,10,16]. Hence, at any time t the base classification model $P(y|\boldsymbol{x};\boldsymbol{\theta})$ can be learned from the set of leaf regions $L^{(t)} = \{(R_i, \mu_i)\}$ where each region R_i has been labeled as μ_i and contains a certain number of training instances.

Apart from the complex learning methods, we adopt another simple but effective method based on *instance sampling* such that instance-based learning algorithms can be used (introduced by HALR [7]). The idea is to create a sample of labeled instances $S = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{K}$ from $L^{(t)}$. The $\{\boldsymbol{x}_k\}_{k=1}^{K}$ part in S is sort of fixed while each of the label y_k is sampled from Bernoulli distribution with the parameter equal to μ_i, which is the proportion label of region R_i that contains $\boldsymbol{x}_k$. Now given S, the parameter vector of the base model can be learned through maximum likelihood estimation (MLE), denoted by $\hat{\boldsymbol{\theta}}$. $\hat{\boldsymbol{\theta}}$ may vary because of the randomness in S, however under some moderate MLE assumptions required by Central Limit Theorem, $\hat{\boldsymbol{\theta}}$ asymptotically follows a normal distribution $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ conditioned on $\{\boldsymbol{x}_k\}$, where $\boldsymbol{\theta}$ is the converged parameter when $K \to \infty$ and the variance $\boldsymbol{\Sigma}$ is the inverse of Fisher information matrix $\mathcal{I}_K(\boldsymbol{\theta})$ depending on the actual finite sample size K. In practice, the asymptotic property can be satisfied by sampling multiple times the label of each $\boldsymbol{x}_k$ and aggregating them up into S. In our experiments each instance label is sampled from 5 to 10 times depending on datasets and then S is large enough to give a small Σ (estimated as $\hat{\Sigma}$ by $\hat{\boldsymbol{\theta}}$).

8 Experiments

We conduct an empirical study to evaluate our proposed approach on 8 general binary classification data sets collected from UCI machine learning repository [1]. The purpose of this study is to research how efficiently (in terms of number of queries) our framework can learn classification models in cost-sensitive tasks.

8.1 Data Sets

The 8 data sets come from a variety of real life applications:

i. **Seismic:** Predict if seismic bumps are in hazardous state.
ii. **Ozone:** Detect ozone level for some days.
iii. **Messidor:** Predict if Messidor images contain signs of diabetic retinopathy.
iv. **Spam:** Detect spam emails in commercial emails.
v. **Music:** Classify the geographical origin of music.
vi. **Wine:** Predict wine quality based on its properties.
vii. **SUSY:** Distinguish a physical signal from background process.

Table 2 suggests various properties of the datasets. Some have been used in previous work (*Wine*) [11,15]; some are high-dimensional (*Ozone, Spam, Music*); and some are unbalanced in class distribution (*Seismic, Ozone, Wine unbalance*).

Table 2. 8 UCI data sets

Dataset	# of data	# of features	Major class %	Feature type
Seismic	2584	18	93%	Numeric, Symbolic
Ozone	1847	72	93%	Numeric
Messidor	1151	19	53%	Numeric, Symbolic
Spam	4601	57	60%	Numeric, Symbolic
Music	1059	68	53%	Numeric
Wine	4898	11	67%	Numeric
Wine_{ub}	1895	11	95%	Numeric
SUSY	5000	18	55%	Numeric

8.2 Methods Tested

We compare our method (HALR) to 3 different methods:

i. **DWUS:** Density-Weighted Uncertainty Sampling is an instance-based method that combines both the uncertainty score and the structure of data [12].
ii. **RIQY:** The state-of-the-art method with proportion feedback on regions [11].
iii. **HS:** Hierarchical Sampling by Dasgupta [2].

8.3 Experimental Settings

Data Split. We split each data set into three disjoint parts: the initial labeled dataset (about 1%–2% of all available data), a test dataset (about 25% of data) and an unlabeled dataset $\mathcal{U}$ (the rest) used as training data. DWUS and RIQY require the initial labeled data to start training, but not our method nor HS.

Region Proportion Label Feedback. To simulate the effect of a human oracle in determining the label of a region, RIQY has originally introduced the way of region queries, which is to simply count the class proportion from labels of the empirical instances that fall into the region.

Evaluation Metrics. We adopt Area Under the Receiver Operating Characteristic curve (AUC) to evaluate the generalized classification quality of Logistic Regression on the test data. Our graphs will plot the AUC scores iteratively after each $t \leq 200$ queries are posed, which is large enough for all methods to converge. Also we assume all kinds of queries consume the same unit cost, although in practice sometimes a instance query is cheaper or oppositely in our cases a region query is more feasible and efficient. To reduce the experiment variations all results are averaged over 20 runs in different random splits.

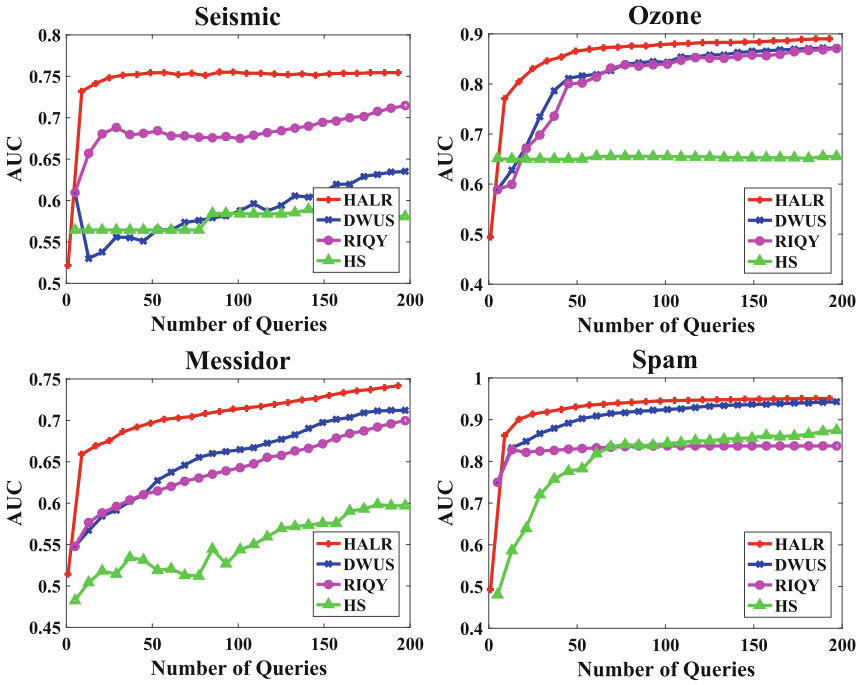

Fig. 2. Performances of different methods on the first 4 datasets (Color figure online)

8.4 Experiment Results

The main results are shown in Figs. 2 and 3. Overall, our HALR (in red line) is able to outperform other methods on majority of the datasets and is close to the best performing method on the remaining sets. There are two primary strengths: first, initially when the labeling budget is severely limited, learning with region-based feedback is superior to learning with the same number of labeled instances, simply because generic region-based queries can carry richer class information than specific instance queries. Second, the initial steep slopes and early convergence in our learning curves lend great credence to our active learning strategy that it is capable of splitting the most uncertain region in the right way and consequently it can accelerate the base model convergence rate.

Unbalanced Class. For data sets *Seismic, Ozone* and *Wine unbalance* (simulated from *Wine*) with unbalanced class distribution, our method performs even better as it could capture the minor class information via proportion labels. In contrast, instance-based methods (e.g. DWUS) may find them slowly; hierarchical sampling (HS) completely failed due to the reason that it always determines the labels of unlabeled instances by majority vote in those *pure enough* (but not entirely pure) clusters, which may totally lose the minor class information.

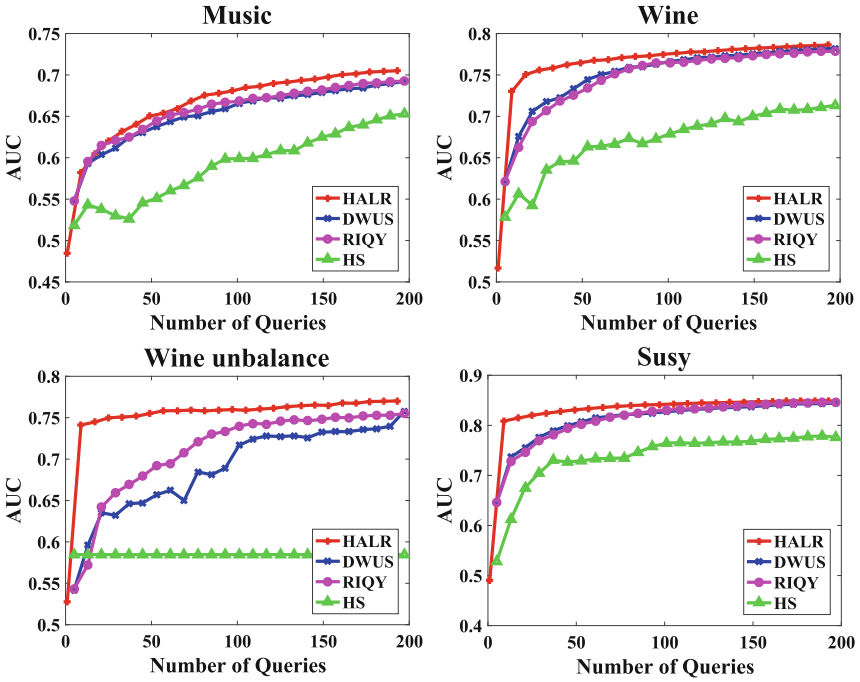

Fig. 3. Performances of different methods on the last 4 datasets (Color figure online)

Complexity of Region Description. Here we show how complex on average our region description could be, in terms of number of features used in the conjunctive patterns. In particular, we calculate *feature reduction rate* for each region R, which is defined as $1 - \frac{\#features\ to\ describe\ R}{\#(All\ features)}$. The results in Table 3 show the average reduction rate among 20 repetitions. This table suggests that region-based queries only use less than half or even 10% of the full dimensional information for human to annotate. This property considerably simplifies the interaction with human annotators when objects are high-dimensional, as region-based queries will present only the relevant features for querying.

Table 3. The averaged feature reduction rate (FRR) of region queries

Dataset	FRR	Dataset	FRR
Wine	59%	Spam	76%
Ozone	90%	Music	90%
Messidor	66%	SUSY	74%
Seismic	77%	Wine_{ub}	58%

9 Conclusions

We develop a new learning framework HALR that can actively learn instance-based classification models from proportion feedback on regions. The regions used in our framework are formed by hierarchical division of the input feature space. In each of the splits, we choose the most uncertain region to divide which considers both the size and the label purity of the region. Then the actual splits are co-decided by both unsupervised and supervised heuristics. Our empirical experiment results show that the regions can be refined to be pure in very few splits and thus they are able to improve the base model quality rapidly. In terms of application, our framework is best suited when providing region-based feedback is more feasible or easier than instance-based queries, as we only present the relevant and partial feature information for querying.

Acknowledgements. The work presented in this paper was supported by NIH grants R01GM088224 and R01LM010019. The content of the paper is solely the responsibility of the authors and does not necessarily represent the official views of NIH.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
2. Dasgupta, S., Hsu, D.: Hierarchical sampling for active learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 208–215. ACM (2008)
3. Du, J., Ling, C.X.: Asking generalized queries to domain experts to improve learning. IEEE Trans. Knowl. Data Eng. **22**(6), 812–825 (2010)
4. Hauskrecht, M., et al.: Outlier-based detection of unusual patient-management actions: an ICU study. J. Biomed. Inform. **64**, 211–221 (2016)
5. Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G.F., Clermont, G.: Outlier detection for patient monitoring and alerting. J. Biomed. Inform. **46**(1), 47–55 (2013)
6. Kück, H., de Freitas, N.: Learning about individuals from group statistics. CoRR abs/1207.1393 (2012). http://arxiv.org/abs/1207.1393
7. Luo, Z., Hauskrecht, M.: Hierarchical active learning with group proportion feedback. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, pp. 2532–2538 (2018)
8. Nguyen, Q., Valizadegan, H., Hauskrecht, M.: Learning classification models with soft-label information. J. Am. Med. Inform. Assoc. **21**(3), 501–508 (2014)
9. Patrini, G., Nock, R., Rivera, P., Caetano, T.: (Almost) no label no cry. In: Advances in Neural Information Processing Systems, pp. 190–198 (2014)
10. Quadrianto, N., Smola, A.J., Caetano, T.S., Le, Q.V.: Estimating labels from label proportions. J. Mach. Learn. Res. **10**, 2349–2374 (2009)
11. Rashidi, P., Cook, D.J.: Ask me better questions: active learning queries based on rule induction. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 904–912. ACM (2011)
12. Settles, B.: Active learning. Synth. Lect. Artif. Intell. Mach. Learn. **6**(1), 1–114 (2012)
13. Urner, R., Wulff, S., Ben-David, S.: PLAL: cluster-based active learning. In: Conference on Learning Theory, pp. 376–397 (2013)

14. Valizadegan, H., Nguyen, Q., Hauskrecht, M.: Learning classification models from multiple experts. J. Biomed. Inform. **46**(6), 1125–1135 (2013)
15. Xue, Y., Hauskrecht, M.: Active learning of classification models with likert-scale feedback. In: SIAM Data Mining Conference. SIAM (2017)
16. Yu, F., Liu, D., Kumar, S., Tony, J., Chang, S.F.: \proptoSVM for learning with label proportions. In: ICML, pp. 504–512 (2013)
17. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In: ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, vol. 3 (2003)