Region-Based Active Learning with Hierarchical and Adaptive Region Construction

Zhipeng Luo and **Milos Hauskrecht**
Department of Computer Science, University of Pittsburgh, PA, 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 human feedback. A region is defined as a hyper-cubic subspace of the input space X 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 quickly discover pure regions (in terms of class proportion) in the data, we have developed a novel active learning framework that constructs regions in a *hierarchical* and *adaptive* way. *Hierarchical* means that regions are incrementally built into a hierarchical tree, which is done by repeatedly splitting the input space. *Adaptive* means that our framework can adaptively choose the best heuristic for each of the region splits. Through experiments on numerous datasets we demonstrate that our framework can identify pure regions in very few region queries. Thus our approach is shown to be effective in learning classification models from very limited human feedback.

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 is active learning. It aims to sequentially select examples to be labeled next by evaluating their possible impact on the model. Active learning has been successfully applied in domains as diverse as computer vision, natural language processing and bio-medical data mining [NVH14, XH17, XH19].

As most active learning works focus on reducing the number of labeled instances, in the end it may not be able to save the human annotation cost as desired if instance labeling is expensive. Recently there has been a new direction of research called *learning from label proportion* (LLP) that proposes to learn models from *grouped data* with their *proportion labels* [QSCL09, Rue10]. The best time to apply LLP is when labeling instances is hard while annotating groups of instances is easier. Consider two realms of applications: (1) in political elections or other activities where the privacy is a concern, collecting individual's feedback is infeasible but acquiring a group of people's feedback is easy [QSCL09, Rue10]; (2) in medical domain, patient records can be very complex in that each record has numerous entries with very high precision. The review and the assessment of these records w.r.t. a specific condition may become extremely time-consuming. To speed up such annotation process, [DL10, RC11] propose to learn from *region-based feedback* which is to assess a *group* of patients by reviewing only their *summarized* conditions.

To solve the original annotation cost problem, [DL10, RC11] and [LH18a, LH18b] have combined the above two solutions and made them work together. The most recent work is [LH18b] which develops a *region-based active learning* framework called HALR (**H**ierarchical **A**ctive **L**earning with proportion feedback on **R**egions) that actively constructs a hierarchy of regions and learns instance-level classification models from the region hierarchy. Their basic ideas are: (1) classification models are learned from labeled regions only, no labeled instances needed; (2) a region is a subspace of the input space X and it is described as conjunctive patterns which are joint value ranges over the input features; (3) each region is annotated with a *proportion label* [QSCL09, Rue10] which is a number in [0,1] representing a human estimate of the positive or negative proportion of the instance population in that region; (4) to identify meaningful regions for labeling and learning they implement a hierarchical approach

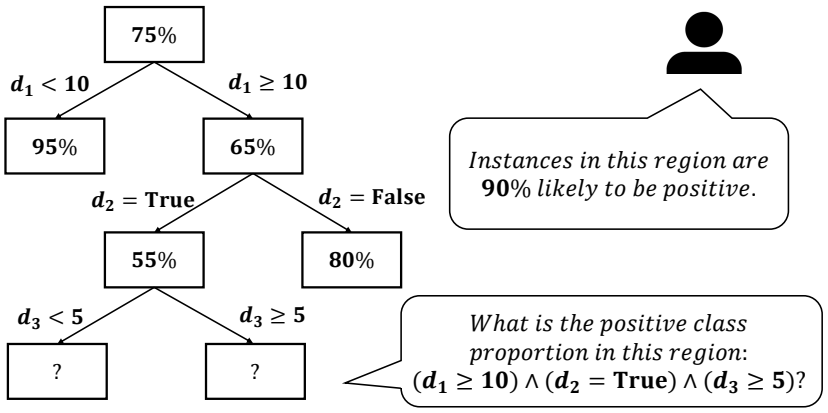

Figure 1: An example of constructing a hierarchical tree of regions. The tree is conceptually equivalent to a decision tree. On the left it shows a snapshot of the tree structure after $t = 3$ splits, generated from the root region on the top level. Each rectangle represents a certain region and the percentage number means its proportion label. Each link is a value constraint on some dimension d_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), the region is described as 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.

that can actively construct a tree of regions, with the goal to find out *pure* leaf regions after very *few* splits are made. The whole framework is illustrated in Figure 1. [LH18a, LH18b] have shown empirically that active learning from region proportion feedback works more efficiently than learning from instance feedback.

Limitation of HALR. The core of HALR is to implement a good region split algorithm such that when splits are made the child regions can be refined as pure as possible. The current implementation in HALR is to have human oracle dynamically evaluate two splitting heuristics separately and then chooses the better one as the final split. The two heuristics are: (1) *unsupervised* split (based on clustering) and (2) *supervised* split (based on classification). We notice that this "dynamic action choosing" puzzle can be formulated as a Multi-Arm Bandit (MAB) problem. With the help of MAB research, we further observe that the current heuristic selection procedure used in HALR is one algorithm belonging to a naive solution set called *uniform exploration* in MAB. Uniform exploration, however, often performs worse than *adaptive* solution set [Sli18].

Contribution of this paper. In this paper, we (1) reformulate HALR as a MAB problem; (2) study the underlying relationship between the two splitting heuristics; and then (3) implement an *adaptive* heuristic selection procedure that should choose a good splitting heuristic to divide regions. Our new approach has been evaluated on a number of datasets and it is shown to outperform many state-of-the-arts methods [RC11, LH18a, LH18b].

2 Background

2.1 Motivation of Region Feedback. The motivation for using region queries is that in many practical domains, annotators may prefer to work with conjunctive pattern-based queries which are shorter, less confusing and more intuitive. As an example consider the heart disease classification task presented in [RC11]:

An instance query example. An instance query for a heart disease diagnosis often covers all feature values of each patient case: *"Consider a patient with (sex=female) ∧ (age=39) ∧ (chest pain type=3) ∧ (fasting blood sugar=150 mg/dL) ... (20 more features). Does the patient have a heart disease?"* The label is a binary {*true*, *false*} response.

A region query example. In contrast, a region query can obtain the class information of a *population* of instances. A region is described by conjunctive patterns which consist of only *relevant* features. For example, *"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 the patient population who suffer from the heart disease. Say *"75% patients within this region suffer from heart disease"*.

2.2 Active Learning from Region Feedback. The central question of learning from region-based feedback is that how can one identify meaningful regions for labeling and learning if there exist no explicit regions in data. The "meaningfulness" is twofold: (1) to human annotators, a region should represent some reasonable population for labeling; (2) to model builders, large and pure regions should be discovered as soon as possible in order to maximize the learning efficiency.

There have been several works that attempt to actively identify regions in data. AGQ+ [DL10] and RIQY [RC11] are early works that form regions by enriching nearby instances centered around a most uncertain instance. However, this ad-hoc approach may fail to identify very meaningful regions [LH18a, LH18b].

To form regions in a more structured way, [LH18a] first proposed to actively construct regions in a hierarchical way. Their framework, named as HALG, relies on hierarchical clustering which outputs a hierarchy of clusters. Within this hierarchy, they actively select the most informative clusters and present them as regions to humans for assessment. One major drawback of this *pre-clustering* based approach is that their active region selection procedure can only select regions within this *fixed* hierarchy. One negative consequence is that the regions formed in this way are overly dominated by clustering, which is an unsupervised heuristic that may be totally irrelevant to classification.

To overcome the issue caused by hierarchical clustering, [LH18b] proposed to *dynamically* construct regions instead. Their new framework is called HALR which identifies regions by directly splitting the input space, working like a decision tree learning process. In each learning iteration, they split a most uncertain leaf region by applying either *unsupervised* heuristic or *supervised* heuristic. More specifically, given a region to split, HALR first (1) performs two test splits separately suggested by both heuristics; then (2) has human annotators evaluate the two splits; and finally (3) chooses the split with more information gain to divide that region. In the view of Multi-Arm Bandit research, this "uniform trying and choosing the best" implementation is one *uniform exploration* strategy which often performs worse than *adaptive* strategies [Sli18]. Therefore, in this work we implement an *adaptive* heuristic selection procedure which can intelligently choose a proper splitting heuristic without uniformly trying each heuristic.

2.3 Multi-Arm Bandit (MAB). MAB is a rich and multi-disciplinary area studied extensively in Statistics, Economics, Operations Research and Computer Science [Sli18]. It is used to model a plethora of *dynamic* optimization problems under uncertainty [BGZ14]. The basic setting is simple: there is a fixed and finite set of actions, aka K arms and each arm can be pulled (chosen) with a unit cost but gives the puller a reward which is independently and randomly generated from a unknown reward distribution. Each arm may have a different reward distribution and the only way we know about the distribution is only through the sample rewards we have pulled. Given a fixed number of times T that one can pull, the goal is to find a good policy (i.e. at each time $t = 1, 2, ..., T$ decides which arm to pull) such that the total sum of rewards can be maximized. In this dynamic learning setting, there exists a tension between the acquisition cost of new information (*exploration*) and the generation of instantaneous rewards based on the existing information (*exploitation*). A simple algorithm called *uniform exploration* is to pull each arm several times and then always choose the arm with the maximum averaged rewards to pull for the rest times. This approach has a major flaw that the exploration schedule does not depend on the history of the observed rewards. It is usually better to *adapt* exploration to the observed rewards. Hence, a lot of good algorithms belonging to *adaptive* solution set have been developed [Sli18].

So if the framework HALR is reformulated as a MAB problem, it is not hard to see: (1) the total number of active learning cycles is equal to T and each cycle t corresponds to each time index; (2) there are $K = 2$ arms that are the two region splitting heuristics *unsupervised heuristic* and *supervised heuristic*; (3) the reward of choosing each heuristic is the information gain brought by that heuristic; (4) at each cycle t, we should develop a good procedure (policy) that chooses one heuristic to split the most uncertain region. In HALR, the policy is one type of uniform exploration. In this work, we implement a near-optimal adaptive policy proposed by [BGZ14].

3 Problem Settings

3.1 Framework Overview. In this paper we implement an adaptive region division algorithm that can be integrated into the framework of HALR, and we name our new approach as A*HALR. We aim to actively build a hierarchical tree of regions annotated with proportion labels and then learn an instance-level binary classification model from the tree leaves. The key steps are summarized in Algorithm 1. The tree is initialized with a root region that covers the entire input space. It also covers all the unlabeled data $\mathcal{U}$ collected beforehand (line 1). The root region is assigned with a proportion label (line 2). Then, the tree gradually grows through active learning cycles where in each cycle a most uncertain leaf region will be split into two sub-regions. More

Algorithm 1 Hierarchical Active Learning with Adaptive Region Construction (A*HALR)

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

1: $\mathcal{T} \leftarrow$ Build a 1-node tree whose root region is the entire feature space X;
2: Query the proportion label μ of $\mathcal{T}$'s root R;
3: Leaf nodes $L^{(1)} \leftarrow \{R\}$;
4: Active learning time $t \leftarrow 1$;
5: Initialize an *adaptive* heuristic-choosing policy π;
6: **repeat**
7: Learn the base model $P(y|\boldsymbol{x};\hat{\boldsymbol{\theta}}^{(t)})$ from $L^{(t)}$;
8: Select the most *uncertain* region R_* in $L^{(t)}$;
9: π chooses *unsupervised* or *supervised* heuristic;
10: Split R_* into two sub-regions guided by the chosen heuristic;
11: Query or infer the proportion labels of the new sub-regions;
12: Update π based on the gain of splitting R_*;
13: $L^{(t+1)} \leftarrow \{L^{(t)} - R_*\} \cup \{R'_* s \text{ sub-regions}\}$;
14: $t \leftarrow t + 1$;
15: **until** the budget T is reached
16: **return** $P(y|\boldsymbol{x};\hat{\boldsymbol{\theta}}^{(T)})$

specifically, from line 7 to 14 we do the following: (1) select the most *uncertain* leaf region R_*; (2) choose one splitting heuristic $a = \pi(t)$ according to the adaptive policy π; (3) divide R_* into two sub-regions suggested by a; (4) query or infer the labels of the new sub-regions; (5) update π's evaluation about heuristic a with the information gain (reward) of splitting R_* by a; (6) retrain the base model with the updated set of leaf regions.

3.2 Base Learning Task. Our goal is to learn an instance-level binary classification model $P(y|\boldsymbol{x};\boldsymbol{\theta})$, which is a discriminate probability distribution function governed by a parameter vector $\boldsymbol{\theta}$. We refer to the model as the *base model*. Each $\boldsymbol{x}$ represents one instance from the input space $X \subset \mathbf{R}^m$ and $y \in \{0, 1\}$ denotes the instance's label.

3.3 Regions. In traditional supervised learning, models are learned through a sample of labeled instance pairs $\{(\boldsymbol{x}_i, y_i)\}$. Our framework, however, does not require labeled instances but we assume that *regions* can be annotated by humans with proportion labels. A region is a hyper-cubic subspace of the input space X, representing a subpopulation of instances that fall into the region. In order to find meaningful regions for labeling and learning, we propose to build a hierarchical tree of regions which are driven by empirical data as well as human feedback on regions. To start the learning process, we first collect a pool of abundant unlabeled training instances $\mathcal{U}$ that are i.i.d. sampled from X. On top of $\mathcal{U}$, a one-node tree is initialized of which the root node (region) is exactly the entire input space X that covers all the instances in $\mathcal{U}$. Then we grow this tree by repeatedly splitting one leaf region into two sub-regions, working like a decision tree building process. This binary split is made on some value v of some dimension d on the parent region, and the two sub-regions are generated with additional value constraint on the dimension d either with $d < v$ or $d \geq v$. Each region is thus defined by conjunctive patterns, which are value ranges over different dimensions joined by *and* operator. In the end a hierarchical tree of regions will be constructed where the leaf regions partition the whole input X space as well as all the data in $\mathcal{U}$.

In terms of providing feedback, human annotators review regions solely based on their conjunctive patterns, without investigating individual instances. Region feedback is given through a proportion label that reflects the positive or negative proportion of the instance population that falls into the region. Equivalently, the proportion label can be also interpreted as the probability that an instance with positive or negative class is drawn from that region population.

4 Active Region Division

This section deals with the problem of how to split the most uncertain leaf region during each active learning cycle (line 7-14 in Algorithm 1). We first define the uncertainty of regions as follows.

4.1 The Uncertainty of Regions. We measure the uncertainty of regions by considering two factors: (1) label impurity and (2) enclosed population density. The former factor is intuitive in that a proportion label would directly reflect the class entropy. The latter one which regards the density of $\boldsymbol{x}$ is also an important factor. To maximize the learning efficiency, we should refine regions which can represent a general and large population of $\boldsymbol{x}$. To measure such density with a single number, a simple way is to use the count of the empirical instances in $\mathcal{U}$ that are enclosed in that region.

To combine the above two factors, we propose to use *Gini-Index* measurement which is a product of label entropy and number of instances[1]. Formally, suppose there are $N^{(t)}$ leaf regions $L^{(t)} = \{(R_i, \mu_i)\}_{i=1}^{N^{(t)}}$ at learning time t. Each region $R_i = \{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}$ contains n_i instances and has been assigned a proportion label $\mu_i \in [0, 1]$ (say the *positive* class proportion). The

[1] Detailed derivation can be checked out in Section 5, [LH18b]

uncertainty score U of each region R_i is defined as:

$$U((R_i, \mu_i)) = 2\mu_i(1-\mu_i)n_i$$

Hence, the most uncertain region is given by:

$$R_* = \arg \max_{(R_i,\mu_i)\in L^{(t)}} U((R_i, \mu_i))$$

4.2 Two Region Splitting Heuristics. Now we need to determine how to perform a "rectangular" split on R_*. It is exactly the situation encountered in decision tree learning. In a standard decision tree learning algorithm, each leaf region is split at some value along one of the input dimensions. By comparing all possible splits, the best one that leads to the maximum information gain can be identified. Unfortunately, such splitting process requires instance labels. Thus it cannot be replicated in our framework where instance labels are unknown. Therefore, we resort to two heuristics that can predict instance labels and then use the predicted labels to determine splits.

4.2.1 Unsupervised Heuristic. The first heuristic is *unsupervised* which is based on clustering. 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. 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 a mixture of two cluster centers in $\{\boldsymbol{x}_{*j}\}$. The probabilities of cluster membership are given by Expectation and Maximization (EM) algorithm. Then 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 (e.g. Gini-Index based) can be now applied to split R_*. Say this procedure gives us the empirical optimal split of R_* from value v^U on dimension d^U.

4.2.2 Supervised Heuristic. Our second heuristic is *supervised* and it relies on the base classification model $P(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}^{(t)})$. The probability that an instance $\boldsymbol{x}_{*j}$ belongs to the positive class can be predicted as $p_j^S = P(y = 1|\boldsymbol{x}_{*j}; \hat{\boldsymbol{\theta}}^{(t)})$, i.e. a **S**upervised probabilistic label. Similarly, given these instance-level labels, Gini-Index based information gain can again be applied to split R_*. Say it suggests a split from value v^S on dimension d^S.

4.2.3 Pros & Cons of the Two Heuristics. Table 1 has summarized the strengths and limitations of the two heuristics. How to choose an appropriate heuristic at each time $t = 1, 2, ...$ in a sequence of region splits becomes a non-trivial challenge.

	Unsupervised	Supervised
Pros	Based on the semi-supervised assumption which is often effective.	Gives instance-level estimate which directly reflects the class distribution.
Cons	But this assumption may not always hold.	But initially the estimate is poor merely because the supervision is scarce.

Table 1: Comparison of the two heuristics.

4.3 Adaptive Heuristic-Choosing Policy. To solve the heuristic-choosing problem we propose an *adaptive* policy based on Multi-Arm Bandits (MAB). First let us reformulate our hierarchical active learning framework (Algorithm 1) as a MAB problem:

1. The total number of active learning cycles T is equal to the total number of pulls in MAB;
2. There are only two heuristics $\{a^U, a^S\}$ (aka $K = 2$ arms) to choose from, where a^U or a^S denotes the unsupervised heuristic or supervised heuristic.
3. In each cycle $t = 1, 2, ..., T$, a policy $\pi(t) : [T] \to \{a^U, a^S\}$ decides which heuristic to use;
4. The reward X_t^π, calculated by Algorithm 2, is the information gain after splitting R_* guided by heuristic $a = \pi(t)$. We assume X_t^π is a random variable drawn from a unknown distribution $P_{\phi^{(t)}}$, where $\phi^{(t)}$ is the mean reward at time t. This $\phi^{(t)}$ can vary for different heuristics, so it is further denoted by $\phi_a^{(t)}$, where $a \in \{a^U, a^S\}$. Another important fact is that the superscript t reflects that the mean rewards for both heuristics *can* change over time. So it is categorized as MAB with *non-stationary* or *stochastic* rewards [BGZ14]. In this sense, $\phi_a^{(t)}$ is also a random variable.
5. In the paper of [BGZ14], $\phi_a^{(t)}$ is assumed to be bounded by a *variation budget* $\mathcal{B} = \{V_t : t = 1, 2, ..., T\}$ which is a non-decreasing sequence of positive real numbers such that $V_1 = 0$, $KV_t \le t$ for all t. Then the possible values of the mean reward sequence for both heuristics, denoted by $\phi = ((\phi_U^{(1)}, ..., \phi_U^{(T)})^\mathbf{T}, \phi_S^{(1)}, ..., \phi_S^{(T)})^\mathbf{T})^\mathbf{T}$, fall into the corresponding *temporal uncertainty set* $\mathcal{V}$:

$$\mathcal{V} = \{\phi \in [0, 1]^{K\times T} : \sum_{t=1}^{T-1} \sup_a |\phi_a^{(t+1)} - \phi_a^{(t)}| \le V_T\}$$

Remarks: the reasoning of using $\mathcal{B}$ and $\mathcal{V}$ is to describe the magnitude that how $\phi_a^{(t)}$ changes over time. V_T bounds the maximum sum of those changes over T. So V_T should in general be designed as a function of T. We will study and compare $\phi_a^{(t)}$ for both our heuristics later.

6. The quality of each policy π is measured by *regret* compared to a *dynamic oracle* as the worst-case difference between the expected performance of choosing at each cycle t the heuristic which has the highest expected reward $\phi_*^{(t)}$ at t (the dynamic oracle performance) and the expected performance under policy π. That is:

$$\mathcal{R}^\pi(\mathcal{V}, T) = \sup_{\phi \in \mathcal{V}} \{\sum_{t=1}^{T} \phi_*^{(t)} - \mathbb{E}^\pi[\sum_{t=1}^{T} \phi_\pi^{(t)}]\}$$

7. Finally, given the prior knowledge of $\mathcal{B}$ and $\mathcal{V}$, our goal is to find a good policy π that minimizes its regret $\mathcal{R}^\pi(\mathcal{V}, T)$. In our problem, it means that we should construct a hierarchical tree of regions where the leaf regions can be refined as *pure* as possible after very *few* splits and queries made. That is, maximum information gain is realized.

Algorithm 2 Evaluation of One Region Split

Input: A labeled region (R_*, μ_*); A splitting heuristic $a \in \{a^U, a^S\}$

Output: the information gain G_a after splitting R_*

1: Split R_* from value v on dimension d suggested by heuristic a into two sub-regions R^L and R^R;
2: Route each instance in R_* to R^L or R^R by testing the feature value of the instance on dimension d either $< v$ or $\geq v$;
3: Query the proportion label of either sub-region. Say R^L is annotated by human with a label μ^L;
4: 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_*, μ_R is calculated as: $(n_*\mu_* - n^L\mu^L)/n^R$;
5: Apply Gini-Index to calculate the information gain:

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

where $GI(\mu) = 2\mu(1-\mu)$.
6: **return** G_a

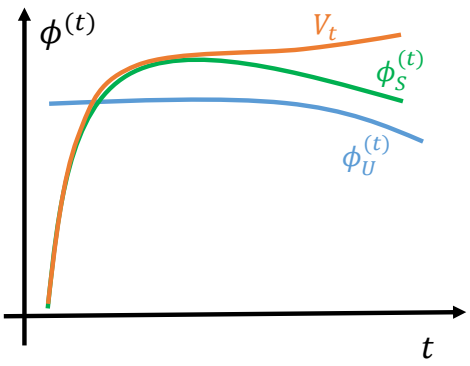

Figure 2: The variation of mean rewards over time.

4.3.1 Mean Rewards of the Two Heuristics. As we have seen, to implement an adaptive policy requires the design of a variation budget $\mathcal{B}$ and a temporal uncertainty set $\mathcal{V}$. So a question is: how do the mean rewards $\phi_a^{(t)}$ of the both heuristics change over time? Recall in Section 4.2.3, we have the prior knowledge that at very beginning (when t is very small), unsupervised heuristic is very likely to perform better ($\phi_U^{(t)} > \phi_S^{(t)}$); however, when more regions are labeled (as t increases), supervised heuristic will catch up and may probably exceed unsupervised heuristic ($\phi_U^{(t)} < \phi_S^{(t)}$). So their changing trends can be roughly plotted in Figure 2. Several remarks:

1. The mean reward of unsupervised heuristic $\phi_U^{(t)}$ should be stable over time as it does not really vary regardless of how many labels are given.

2. The mean reward of supervised heuristic $\phi_S^{(t)}$ will increase when more labels are revealed but its changing trend (slope) may decrease as the base classifier starts to converge. Previous work [LH18b] has shown similar curves of the performance increase of the base models.

3. Both curves drop in the end as the leaf regions become purer and purer (so with less gain). Of course, the tails are of less interest.

4. So how to choose the variation budget function V_T mainly depends on the mean reward curve of supervised heuristic. According to the plot, the shape of $\phi_S^{(t)}$ can be bounded by a log function. Well, to be more conservative, we choose a looser bound of a square-root function: $V_T = O(T^{1/2})$.

4.3.2 A Near-Optimal Adaptive Policy. We implement a near-optimal policy of choosing heuristics

suggested by [BGZ14] with $V_T = O(T^{1/2})$ assumed. The key to dealing with MAB with non-stationary rewards is to break the whole T cycles into batches of size $= \Delta T$ and then explore/exploit heuristics independently in each batch. Each heuristic is chosen with a mixed probability of Boltzmann distribution and uniform distribution, balanced by a constant γ. Within each batch, initially each heuristic is chosen with equal chance and then the probability of the chosen one will be boosted by the reward obtained. Algorithm 3 details this policy. Its near-optimality is proved by Theorem 2 in [BGZ14] when the batch size $\Delta T = \lceil (K \log K)^{1/3} (T/V_T)^{2/3}$ and with $\gamma = \min\{1, (\frac{K \log K}{(e-1)\Delta T})^{1/2}\}$. In our experiments, we set these two hyper-parameters accordingly.

Algorithm 3 A Near-Optimal Policy π

Input: $\gamma \in [0, 1]$; A batch size ΔT.

for all batch of cycles with size $= \Delta T$ **do**
  Set $w_U^{(t)} = w_S^{(t)} = 1$ for both heuristics;
  for all cycle indexed at t in current batch **do**
    $p_U^{(t)} = (1-\gamma)\frac{w_U^{(t)}}{w_U^{(t)} + w_S^{(t)}} + \gamma/K$;
    $p_S^{(t)} = 1 - p_U^{(t)}$;
    Choose a heuristic a randomly according to the distribution $\{p_U^{(t)}, p_U^{(t)}\}$;
    Split R_* suggested by a;
    Receive a reward $X_a^{(t)}$ according to Algorithm 2;
    Boost $w_a^{(t+1)} \leftarrow w_a^{(t)} \exp\{\frac{\gamma X_a^{(t)}}{p_a^{(t)} K}\}$;
    Keep $w_{a'}^{(t+1)} = w_{a'}^{(t)}$ for $a' \neq a$;
  end for
end for

5 Learning a Model from Labeled Regions

Now the last question remained is that how to learn the base classification model from labeled regions. There are existing algorithms that can learn specific models from proportion labels [QSCL09, Rue10]. If a general model needs to be learned, one can adopt a simple yet effective learning algorithm based on *instance sampling*. The idea is to create a large sample of labeled instances $S = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^M$ from $L^{(t)} = \{(R_i, \mu_i)\}$. Each $\boldsymbol{x}_j$ is uniformly sampled from $\mathcal{U} = \bigcup_i R_i$ and the label y_j is sampled from Bernoulli distribution with the parameter equal to μ_i, which is the proportion label of region R_i that contains $\boldsymbol{x}_j$. Based on S, our base model $P(y|\boldsymbol{x}; \boldsymbol{\theta})$ can be learned through maximum likelihood estimation (MLE). Denote by $\hat{\boldsymbol{\theta}}$ the learned parameter vector. $\hat{\boldsymbol{\theta}}$ may vary because of the randomness in S. However, under mild MLE conditions $\hat{\boldsymbol{\theta}}$ asymptotically follows a normal distribution $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ conditioned on $\{\boldsymbol{x}_j\}_{j=1}^M$ [VdV00], where $\boldsymbol{\theta}$ is the converged parameter when $M \to \infty$ and the variance $\boldsymbol{\Sigma}$ is the inverse of Fisher information matrix $\mathcal{I}_M(\boldsymbol{\theta})$ combined with the actual finite sample size M. In practice, the asymptotic property can be satisfied by sampling multiple times the label of each $\boldsymbol{x}_j$ 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 $\hat{\boldsymbol{\theta}}$).

Dataset	# of Data	# of features	Major Class	Feature Type
Seismic	2584	18	93%	N-O-C
Ozone	1847	72	93%	N
Messidor	1151	19	53%	N-C
Spam	4601	57	60%	N-O
Music	1059	68	53%	N
Wine	4898	11	67%	N
Pima	768	8	65%	N-C
Gamma	5000	10	65%	N
SUSY	5000	18	55%	N

Table 2: 9 UCI data sets. 'N', 'O' and 'C' stand for 'Numeric', 'Ordinal' and 'Categorical' respectively.

6 Experiments

We conduct an empirical study to evaluate our proposed approach on 9 general binary classification data sets collected from UCI machine learning repository [AN07]. The purpose of this study is to research how efficiently (in terms of the number of splits and queries) our A*HALR framework can learn classification models in cost-sensitive tasks.

6.1 Data Sets. The 9 data sets come from a variety of real life applications:

- **Seismic**: Predict the states of seismic bumps.
- **Ozone**: Detect ozone level on some days.
- **Messidor**: Predict if Messidor images contain signs of diabetic retinopathy.
- **Spam**: Classify spam commercial emails.
- **Music**: Find the geographical origin of music.
- **Wine**: Predict wine quality.
- **Pima**: Diagnose diabetes disease in Indian women.

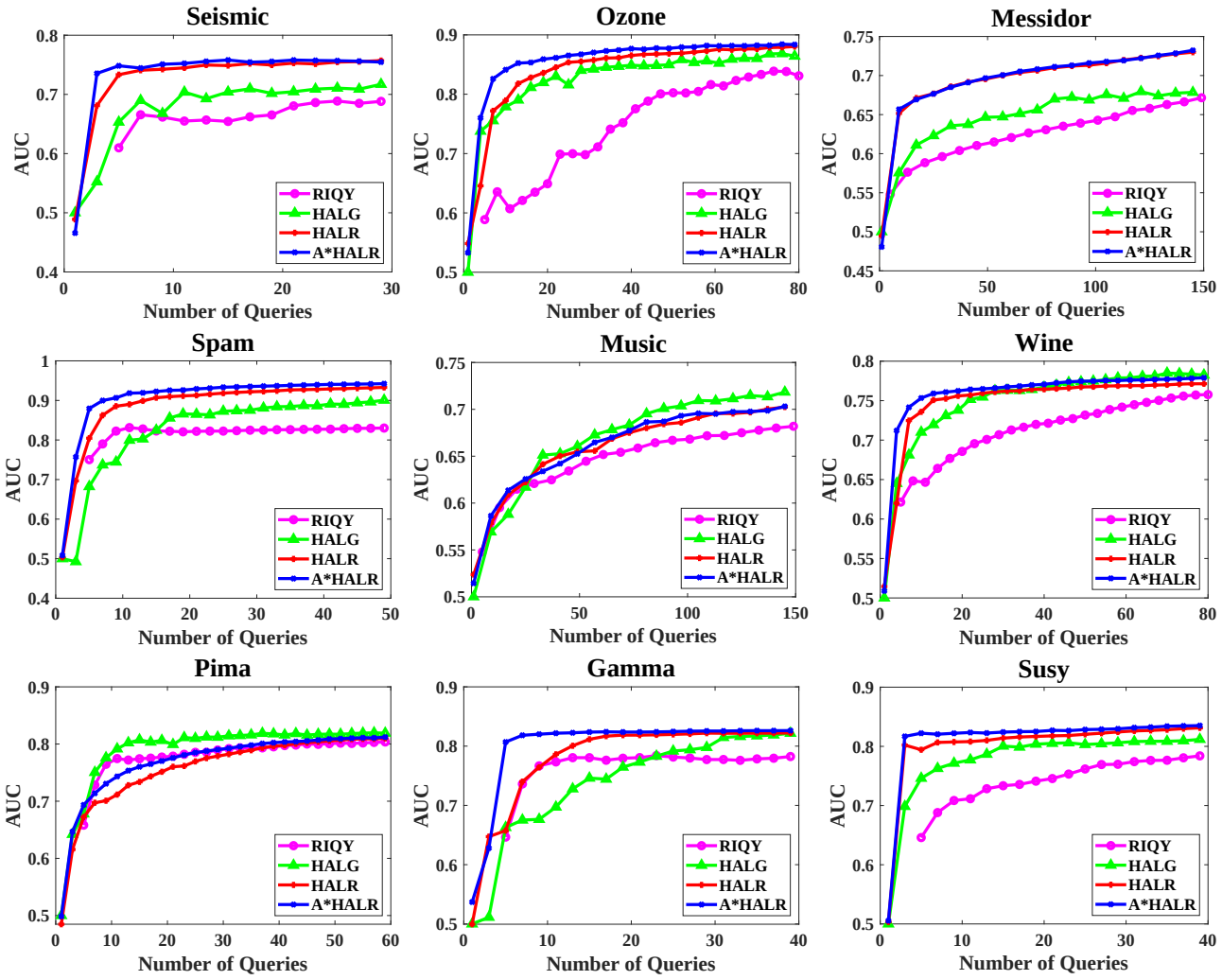

Figure 3: Performance of different methods on 9 UCI data sets.

- **Gamma**: Detect γ-particles in Cherenkov telescope.
- **SUSY**: Distinguish a signal or background process.

Table 2 summarizes the basic statistics of the data sets. Most of them have been widely used in the previous active learning work [RC11, LH18a, LH18b, XH17]. Some have high-dimensional feature space: *Ozone, Spam, Music*; and some carry highly unbalanced class distribution: *Seismic, Ozone*.

6.2 Methods Tested. We compare our new method A*HALR to three previous region-based active learning frameworks. They could be reviewed in Section 2.

- **RIQY**: the first region-based active learning work which constructs regions from instances [RC11];
- **HALG**: actively constructs regions based on hierarchical clustering [LH18a];
- **HALR**: dynamically constructs regions by actively splitting the input space [LH18b].

6.3 Experimental Settings

6.3.1 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. Note that only RIQY requires a small portion of labeled data to start training, while others do not.

6.3.2 Region Proportion Label Feedback. The role of humans in our region-based learning is to review regions' conjunctive patterns and then annotate them

with proportion labels. To mimic such process, in our experiments we use empirical labeled instances to simulate region proportion feedback, as if it were given by humans. That is, to obtain the proportion label for a region R, we count the empirical labeled instances that fall into R to estimate the class proportions. This region label simulation was first introduced to test RIQY.

6.3.3 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 plot the AUC scores after each $t < T = 200$ region queries are posed, which is large enough for all methods to converge (to best visualize each plot, we omit the remaining tails of curves after most methods have converged). To reduce the experiment variations all results are averaged over 20 runs in different random splits.

6.4 Experiment Results. The main results are shown in Figure 3. Overall, our A*HALR (in blue line) is able to outperform other methods on the majority of the data sets and is close to the best method on the remaining sets (*Pima* and *Music*). These results lend great credence to our adaptive region division algorithm which in principle benefits from the near-optimal heuristic-choosing policy developed to solve Multi-Arm Bandit problems with non-stationary rewards.

Compared to HALR, our new method is always performing better simply because that our adaptive approach works more efficiently than uniform exploration approach, which was used to HALR. Compared to HALG, sometimes we lose (on datasets *Pima* and *Music*). One reason could be that for particular datasets if clustering is highly associated with classification, then HALG (where clustering dominates) may be able to learn more efficiently.

7 Conclusions

We developed a region-based active learning framework A*HALR that *adaptively* constructs a hierarchical tree of regions and then learns classification models from the tree. We studied and compared the trade-off between two region splitting heuristics: *unsupervised heuristic* based on clustering and *supervised heuristic* based on classification. With this study, we were able to develop a near-optimal heuristic-choosing policy based on Multi-Arm Bandit with non-stationary rewards. Through experiments on multiple datasets, we demonstrated that our A*HALR can indeed learn decent models from a small number of region queries. Thus our approach is shown effective in learning good classification models while consuming very little human feedback.

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

[AN07] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[BGZ14] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *NIPS*, pages 199–207, 2014.

[DL10] Jun Du and Charles X Ling. Asking generalized queries to domain experts to improve learning. *Knowledge and Data Engineering, IEEE Transactions*, 22(6):812–825, 2010.

[LH18a] Zhipeng Luo and Milos Hauskrecht. Hierarchical active learning with group proportion feedback. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI'18*, pages 2532–2538, 2018.

[LH18b] Zhipeng Luo and Milos Hauskrecht. Hierarchical active learning with proportion feedback on regions. In *European Conference on Machine Learning, ECML'18*. (In process), 2018. Online version: http://www.ecmlpkdd2018.org/wp-content/uploads/2018/09/618.pdf.

[NVH14] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. Learning classification models with soft-label information. *Journal of the American Medical Informatics Association*, 21(3):501–508, 2014.

[QSCL09] Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10(Oct):2349–2374, 2009.

[RC11] Parisa Rashidi and Diane J Cook. Ask me better questions: active learning queries based on rule induction. In *Proceedings of the 17th ACM SIGKDD*, pages 904–912. ACM, 2011.

[Rue10] Stefan Rueping. Svm classifier estimation from group probabilities. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 911–918, 2010.

[Sli18] Aleksandrs Slivkins. Introduction to multi-armed bandits, 2018.

[VdV00] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[XH17] Yanbing Xue and Milos Hauskrecht. Active learning of classification models with likert-scale feedback. In *SIAM Data Mining Conference, 2017*. SIAM, 2017.

[XH19] Yanbing Xue and Milos Hauskrecht. Active learning of multi-class classification models from ordered class sets. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2019.