# Cost-sensitive active learning with a label uniform distribution model

Yan-Xue Wu[a], Xue-Yang Min[b], Fan Min[a,*], Min Wang[c]

[a]*School of Computer Science, Southwest Petroleum University, Chengdu 610500, China*
[b]*College of Environment, Nanjing Tech University, Nanjing 211800, China*
[c]*School of Electrical Engineering, Southwest Petroleum University, Chengdu 610500, China*

**Abstract**

Active learning is a man-machine interaction scenario in which the machine acquires information actively from the expert. Cost-sensitive active learning balances the misclassification cost with the teacher cost paid for label queries. Inspired by granular computing (GrC) and three-way decision (3WD), this paper presents a new algorithm called cost-sensitive active learning through density clustering under a label uniform distribution model (CADU). CADU iteratively divides the universe, queries labels, and classifies instances until each label is queried or predicted. The density clustering technique is used to divide the universe into blocks. A label uniform distribution model is built to calculate the expected label distribution of each block. According to the teacher and misclassification cost settings, an optimization function is designed to determine the number of labels to be queried. Comparison study with 10 state-of-the-art algorithms are undertaken on 12 public datasets. Results show that CADU outperforms others in terms of average cost.

*Keywords:* Active learning; density clustering; granular computing; label uniform distribution; three-way decision.

## 1. Introduction

Active learning [10] is a special case of semi-supervised machine learning. It interactively queries instances that are more informative and beneficial to our learner. Then these instances are labeled by the expert/oracle and added to the training set. This process repeats until all labels are queried or predicted. This technique is widely used in text classification [60], information extraction [59], image classification [83], and speech recognition [76].

Cost-sensitive active learning [36, 40, 79] aims at minimizing the total cost of the learning process. Two types of costs are usually considered. One is the

*Corresponding author. Tel.: +86 135 4068 5200.
Email: minfanphd@163.com.

teacher cost of acquiring the class label from an expert [62]. The other is the misclassification cost of deciding that an object belongs to one class when its real class is another [61]. Most active learning scenarios require the user to specify the number of labels provided by the expert. In contrast, cost-sensitive active learning automatically determines this number to achieve a compromise between teacher and misclassification costs.

Inspired by granular computing (GrC) and three-way decision (3WD), this paper proposes a new algorithm called cost-sensitive active learning through density clustering under the label uniform distribution model (CADU). Figure 1 illustrates the CADU process using a running example. First, a master tree

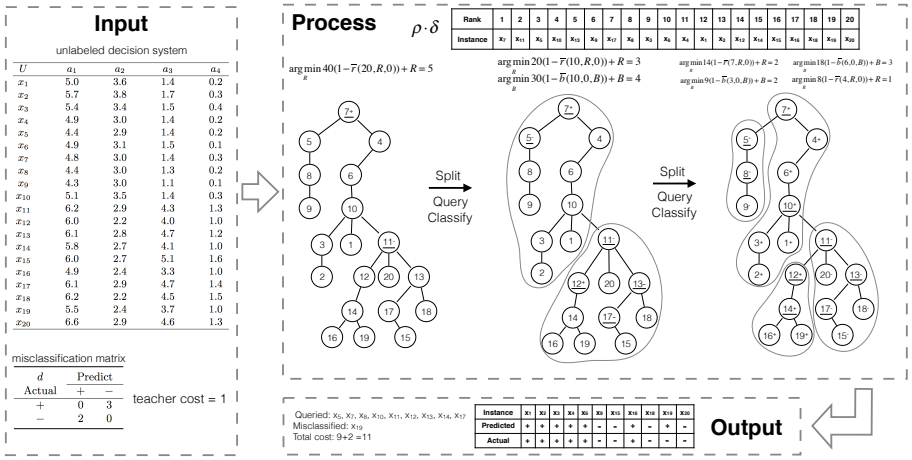



Figure 1: CADU framework

where each node represents an instance is built using the density clustering technique [50, 65]. Nodes with greater product of density and distance are considered more important. Second, the labels for a number of the most important nodes are queried. If they are the same, other instances will be classified accordingly. Otherwise the data is divided into two blocks. This split, query and prediction process terminates after all labels have been queried or predicted.

From the viewpoint of GrC [82], CADU iteratively divides large granules into small ones. Instances are classified according to the most suitable granule, which is as pure and coarse as possible. In this way, the classification ability is assured, and the teacher cost is saved. From the viewpoint of 3WD [73], there are three possible different actions for an instance in each round. These are label query, label prediction and delay decision. They are executed iteratively until all labels have been queried or predicted.

There are two key issues in this process: 1) When the labels being queried are not the same, should we divide the currently block? 2) How many labels should be queried before we classify other instances in the same block? We build a label uniform distribution (LUD) model to address these issues. Based on LUD, we calculate the expected ratio of positive/negative instances of blocks

with some known labels. It shows that once there is a label different from others, we need to query at least twice the number of labels to obtain the same ratio. Consequently, we divide the block into two parts immediately. For the second issue, we consider LUD and both types of costs to build the optimization problem. The optimization objective is to minimize the expected total cost of the current block. It can be solved by Direct Search or Fibonacci Method [9] since the function is unimodal. Through optimizing each block, the global optimal solution might be achieved.

Experiments are conducted on 12 public datasets. We compare the new algorithm with three sets of popular algorithms, namely cost-insensitive active learning, cost-sensitive learning, and cost-sensitive active learning algorithms. Results show that CADU outperforms all of these competing algorithms in terms of average cost. Specifically, compared with ALCE, CWMM, MEC and TALK, CADU reduces the average cost by 20.7%, 56.3%, 29.9% and 37.7% respectively.

The rest of the paper is organized as follows. Section 2 reviews the related works of GrC, 3WD, and cost-sensitive active learning. Section 3 presents the data model and problem definition. The label uniform distribution model is introduced in Section 4. Section 5 illustrates the pseudo code of CADU algorithm and analyzes its time complexity. Section 6 discusses the experimental results and Section 7 makes a conclusion.

## 2. Related works

This section reviews some related works concerning GrC, 3WD, and cost sensitive active learning.

### *2.1. Granular computing*

The concept of GrC was proposed by Yao [71], Lin [75], and Zadel [82]. In a narrow sense, it is about formation, processing and communicating information granules [70]. A granule [6] refers to a set of instances. There are indiscernibility relation [43], tolerance relation [63], or other types of relations between instances of a granule. Two issues, namely granulation and granule operation, are the foundations of GrC [64]. Granulation is a construction process for solution space, while granule operations are designed to conveniently solve these granule-based problems [48].

In a broad sense, it is a methodology of human cognition of complex problem solving. According to Zadeh [82], human cognition consists of three basic concepts: granulation, organization and causation. Granulation decomposes the whole into parts; organization integrates the parts into whole; and causation associates causes with effects. There are various supporting theories, including set theory, interval calculus [45], fuzzy sets [25], rough sets [39, 47], shadowed sets [44], quotient spaces [86], and probabilistic granules [19].

### 2.2. Three-way decision

3WD is an emerging methodology of divide and conquer [73] with tri-section and tri-action. It is also a class of effective ways and heuristics commonly used in human problem solving and information processing [72]. This methodology is an extension of decision-theoretical rough sets (DTRS) founded in 1992 [74].

Various theories and applications have been inspired by 3WD. Three-way formal concept analysis [51] and three-way cognition computing [29] provide concept learning and multi-granularity cognitive operations. Three-way fuzzy sets [14] construct a three-valued approximation with a pair of thresholds on the fuzzy membership function. Three-way decision-theoretic rough sets [87] consider the new risk measurement functions through the utility theory to improve classification correct rate. Three-way approximations [89] analyze the elevation and reduction errors produced by shadowed set. Three-way decision space [17, 18] unifies decision measurement, decision conditions and evaluation functions. Three-way classification [80, 88] builds three-way regions and measure its impurities with Gini coefficients. 3WD based multigranulation rough set theory [55] leads to the definition of decision-oriented aggregation operators. Three-way active learning [40, 65] takes advantage of sequential 3WD to improve classification accuracy. Tri-partition neighborhood covering reduction [81] facilitates robust classification. Three-way clustering [78] presents a new strategy for overlapping clustering. Three-way active clustering [77] investigates a new method via low-rank matrices that can improve clustering accuracy. Three-way spam filtering [91] reduces the email misclassification costs. Three-way medical decision [69] uses web-based decision support systems. Three-way face recognition [27] handles insufficient high quality facial image information. Three-way pattern discovery [41] presents an algorithm for a new type of pattern by dividing the alphabet into strong, medium and weak parts. Three-way recommender systems [20, 84, 85] provide the additional *promotion* option for the users and decrease the total cost. Three-way effective measures [23] facilitate movements of objects from unfavorable regions to favorable regions. Deep neural network-based 3WD [28] provides a novel feature extraction method for image data analysis. Sequential 3WD [46, 68] attribute reduction provides a new insight under dynamic granulation. Three-way decision is also used in extracting collective knowledge [7] from surveys and handle ordered decision systems [32].

### 2.3. Cost-sensitive active learning

According to the way of obtaining unlabeled instances, active learning algorithms are categorized into stream-based [42, 56] and pool-based ones [37, 57, 66]. Stream-based algorithms send unlabeled instances to the selector one by one [34]. Pool-based algorithms maintain an unlabeled set from which the selector labels important instances [52]. They are further categorized into uncertainty based [5], version space based [54] and generalization error reduction based [37] approaches in accordance with their selection criterion.

Clustering-based active learning becomes popular since it does not need an initial training set to construct the classifier. Kang et al. [24] proposed to cluster

all unlabeled instances and query the ones closest to the clusters' centroids, which were added into the initial training set. Mahajan et al. [35] proposed a general framework for clustering-based active learning in which feature selection is used for data reduction. Woo et al. [67] applied the hierarchical agglomerative clustering using Ward's linkage to the learning process. Krempl et al. [26] combined an optimized probabilistic active learning approach with clustering for evolving datastreams. Lorbach et al. [33] clustered unbalanced data by Dirichlet process Gaussian mixture model to obtain a more balanced training set. Xu et al. [15] proposed an approaches for the named entity recognition (NER) task, where the cluster is constructed by the candidate named entities. Wang et al. [65] proposed the active learning algorithm with density peaks clustering. The major drawback is that these approaches depend heavily on the quality of the clustering results [22].

Cost-sensitive active learning has attracted more and more research interest. Margineantu [36] introduced this problem and proposed to estimate the class probabilities over the unlabeled data. Sampling and decision making are based on these estimates. Settles et al. [53] assumed that instances may have different teacher costs. They considered the actual teacher costs involved with human annotators. Liu et al. [31] proposed the cost-sensitive active learning for spatial data. Zhao et al. [90] proposed a new approach to solve the unbalanced problem in the URL detection task. Chen et al. [8] and Agarwal [1] proposed two approaches with probabilistic models. Demir et al. [13] redefined active learning by assuming that the teacher cost during ground survey may vary. Huang et al. [21] embedded the cost information in the distance measure in a special hidden space by non-metric multidimensional scaling. Min et al. [40] applied the 3WD on cost-sensitive active learning considering both misclassification and teacher costs.

## 3. Problem statement

In this section, we describe the data model and the problem. Table 1 lists notations used throughout the paper.

### *3.1. Data model*

We consider the following data model.

**Definition 1.** A teacher-and-misclassification-cost-sensitive decision system (TMC-DS) [40] is the 7-tuple:

$$S = (U, C, d, V, I, m, t), \tag{1}$$

where $U$ is the finite set of instances, $C$ is the set of conditional attributes, $d$ is the decision attribute, $V = \cup_{a \in C \cup \{d\}} V_a$, $V_a$ is the set of values for attribute $a$, $I : U \times (C \cup \{d\}) \to V$ is the information function, $m : V_d \times V_d \to \mathbb{R}^+ \cup \{0\}$ is the misclassification cost function, and $t \in \mathbb{R}^+ \cup \{0\}$ is the teacher cost.

Table 1: Notations.

| Notation | Meaning |
|---|---|
| $U$ | The universe of instances |
| $C$ | The set of conditional attributes |
| $d$ | The decision attribute |
| $V_a$ | The set of values for each $a \in C \cup D$ |
| $V$ | The union of $V_a$ |
| $I$ | The information function |
| $m$ | The misclassification cost function |
| $t$ | The teacher cost |
| $x_i$ | The $i$-th instance in $U$ |
| $y_i$ | The actual label of $x_i$ |
| $l_i$ | The predicted label of $x_i$ |
| $N$ | The size of $U$ |
| $X$ | A subset of $U$ |
| $n$ | The size of $X$ |
| $R$ | The number of positive instances queried in $X$ |
| $B$ | The number of negative instances queried in $X$ |
| $A_i^j$ | The permutation of taking out $j$ elements from $i$ different ones |
| $P(R^*\|R, B; n)$ | The probability of $R^*$ positive instances in $X$ |
| $\overline{r}(n, R, B)$ | Expected proportion of positive instances in $X$ |
| $\overline{b}(n, R, B)$ | Expected proportion of negative instances in $X$ |
| $\sigma(n, R, B)$ | The standard deviation of the proportion of positive instances |
| $f$ | The cost function |
| $s$ | The optimal number of queried instances |

Tables 2 and 3 list an exemplary TMC-DS, where $U = \{x_1, \ldots, x_{20}\}$, $C = \{a_1, a_2, a_3, a_4\}$, $\forall a \in C, V_a \subset \mathbb{R}^+$, and $V_d = \{+, -\}$. Here $m$ and $t$ are application dependant. For example, $m(-, +) = 2$ indicates that the cost is 2 if a negative instance is misclassified to positive. $t = 1$ means that the cost is 1 if the instance is queried.

### *3.2. Problem*

Problem 1 presents the problem definition.

**Problem 1.** ***Cost-sensitive active learning***
***Input:*** *A TMC-DS $S = (U, C, d, V, I, m, t)$ where labels are unknown.*
***Output:*** *The set of queried instances $U_t \subset U$ and the predicted labels for $U - U_t$.*
***Optimization objective:*** $\min cost = \frac{t|U_t| + \sum_{i=1}^{|U|} m(l_i, y_i)}{|U|}$.

The input is a TMC-DS where the labels are unknown. The output has two parts. One is an object subset $U_t$ whose real labels are queried and provided by the expert. The other contains the predicted labels of the remaining objects.

The optimization objective is to minimize the average cost by considering both misclassification and teacher costs. Here $t \times |U_t|$ is the total teacher cost, and $\sum_{i=1}^{|U|} m(l_i, y_i)$ is the total misclassification cost. They are computed after $U_t$ and the predicted labels for $U - U_t$ are obtained.

Naturally, there is a tradeoff between these two types of costs. Note that the size of $U_t$ is not specified by the user. With the increase of $U_t$, the total teacher cost increases linearly, and the total misclassification cost might decrease.

Table 2: An exemplary decision system.

| $U$ | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $d$ |
|---|---|---|---|---|---|
| $x_1$ | 5.0 | 3.6 | 1.4 | 0.2 | + |
| $x_2$ | 5.7 | 3.8 | 1.7 | 0.3 | − |
| $x_3$ | 5.4 | 3.4 | 1.5 | 0.4 | + |
| $x_4$ | 4.9 | 3.0 | 1.4 | 0.2 | + |
| $x_5$ | 4.4 | 2.9 | 1.4 | 0.2 | + |
| $x_6$ | 4.9 | 3.1 | 1.5 | 0.1 | + |
| $x_7$ | 4.8 | 3.0 | 1.4 | 0.3 | + |
| $x_8$ | 4.4 | 3.0 | 1.3 | 0.2 | + |
| $x_9$ | 4.3 | 3.0 | 1.1 | 0.1 | − |
| $x_{10}$ | 5.1 | 3.5 | 1.4 | 0.3 | + |
| $x_{11}$ | 6.2 | 2.9 | 4.3 | 1.3 | − |
| $x_{12}$ | 6.0 | 2.2 | 4.0 | 1.0 | − |
| $x_{13}$ | 6.1 | 2.8 | 4.7 | 1.2 | − |
| $x_{14}$ | 5.8 | 2.7 | 4.1 | 1.0 | − |
| $x_{15}$ | 6.0 | 2.7 | 5.1 | 1.6 | − |
| $x_{16}$ | 4.9 | 2.4 | 3.3 | 1.0 | − |
| $x_{17}$ | 6.1 | 2.9 | 4.7 | 1.4 | − |
| $x_{18}$ | 6.2 | 2.2 | 4.5 | 1.5 | − |
| $x_{19}$ | 5.5 | 2.4 | 3.7 | 1.0 | − |
| $x_{20}$ | 6.6 | 2.9 | 4.6 | 1.3 | − |

Table 3: Misclassification cost matrix.

| $d$ | Predictive | |
|---|---|---|
| Actual | + | − |
| + | 0 | 3 |
| − | 2 | 0 |

## 4. The label uniform distribution model

In this section, we propose a label uniform distribution (LUD) model for our algorithm. This model estimates the label distribution of blocks. In contrast, most of the existing stochastic models require user-specified label distribution. Note that we only consider binary classification problems, where each instance can be positive or negative. More complicated models should be built for decision systems with more than two classes.

Given $X \subseteq U$ with $n$ instances, we obtain $R$ positive and $B$ negative instances under no return sampling. We try to address the following problem: What is the expected number of positive instances in $X$? Note that in our scenario, $n$ is often large (e.g., greater than $10^3$) while $R$ and $B$ are small (e.g., about 10). It is incorrect to say that $\frac{R}{R+B}$ of these instances are positive. We will explain this claim further through examples.

Since we will evaluate the overall distribution of labels, an appropriate assumption of the solution space distribution should be made.

**Assumption 1.** *(The discrete uniform distribution assumption) Suppose that no label is known, i.e., $R = B = 0$. The probability that there are $i$ positive instances in $X$ is the same for any $0 \leq i \leq n$. That is,*

$$\forall 0 \leq i \leq n, P(R^* = i) = \frac{1}{n+1}. \tag{2}$$

Note that the assumption is made on the block instead of individual instances. Although the distribution is "uniform," in most cases the block is imbalanced. For example, in 60% situations the difference between labels exceeds 7:3. This is because that $X$ is a cluster obtained by a clustering algorithm, instead of random sampling. Assumption 1 might be the simplest one considering the clustering quality.

We use some examples to explain this assumption.

**Example 1.** *Let $n = 100$. The probability of $i$ ($0 \leq i \leq 100$) positive instances in $X$ is always $\frac{1}{101}$. The expected number of positive instances is $\sum_{i=0}^{100} \frac{1}{101} i = 50$.*

The expected number of negative instances is also 50. Therefore, from a symmetrical point of view, this assumption is reasonable.

Now we take out some instances from $X$ and observe their labels. From the observation, we try to derive the label distribution of $X$.

**Theorem 1.** *Suppose that $R$ positive and $B$ negative instances are randomly taken from $X$. The probability that there are $R^*$ positive instances in $X$ is*

$$P(R^*|R, B; n) = \frac{A_{R^*}^R A_{n-R^*}^B}{\sum_{i=0}^{n} A_i^R A_{n-i}^B}. \tag{3}$$

Proof. According to the Bayes formula and Assumption 1,

$P(R^*|R, B; n) = \frac{P(R^*)P(R,B|R^*;n)}{\sum_{i=1}^{n} P(R^*=i)P(R,B|R^*=i;n)} = \frac{\frac{1}{n+1}P(R,B|R^*;n)}{\sum_{i=1}^{n} \frac{1}{n+1}P(R,B|R^*=i;n)} = \frac{P(R,B|R^*;n)}{\sum_{i=1}^{n} P(R,B|R^*=i;n)}$.

$P(R, B|R^* = i; n)$ is the probability that we take out $R$ positive and $B$ negative instances from a set with $i$ positive and $(n - i)$ negative instances. Hence $P(R, B|R^* = i; n) = \frac{A_i^R A_{n-i}^B}{A_n^{R+B}}$. Similarly, $P(R, B|R^*; n) = \frac{A_{R^*}^R A_{n-R^*}^B}{A_n^{R+B}}$.

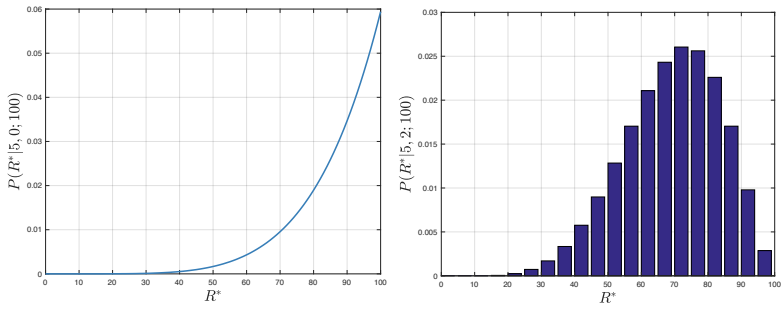
Finally, we have

$$P(R^*|R, B; n) = \frac{P(R,B|R^*;n)}{\sum_{i=0}^{n} P(R,B|R^*=i;n)} = \frac{\frac{A_{R^*}^R A_{n-R^*}^B}{A_n^{R+B}}}{\sum_{i=0}^{n} \frac{A_i^R A_{n-i}^B}{A_n^{R+B}}} = \frac{A_{R^*}^R A_{n-R^*}^B}{\sum_{i=0}^{n} A_i^R A_{n-i}^B}.$$

This completes the proof.

Figure 2 illustrates two distribution functions. Figure 2(a) shows that $P(R^*|5, 0; 100)$ increases with the increase of $R^*$. It shows that if all chosen instances are positive, then the probability of $n$ positive instances in $X$ is the maximal. Naturally, $P(R^*|5, 0; 100) = 0$ when $0 \leq R^* \leq 4$, in which case there are not enough positive instances.

From Figure 2(b) it is observed that $P(R^*|5, 2; 100)$ increases to the maximum and then decreases. This is because both positive and negative instances have been observed. In this figure, it is clearer that the probability function is discrete. Furthermore, $P(R^*|5, 0; 100) = 0$ when $0 \leq R^* \leq 4$ since we have already observed 5 positive instances. $P(R^*|5, 2; 100) = 0$ when $99 \leq R^* \leq 100$ since we have already observed 2 negative instances.




(a) (b)

Figure 2: $P(R^*|R,B;n)$ wrt. the change of $R^*$ where (a) $R=5$ and $B=0$, (b) $R=5$ and $B=2$.

**Theorem 2.** *The expected proportion of positive instances is*

$$\overline{r}(n,R,B) = \frac{\sum_{i=R}^{n-B} iA_i^R A_{n-i}^B}{n\sum_{i=R}^{n-B} A_i^R A_{n-i}^B}. \tag{4}$$

Proof. The expected number of positive instances is
$r(n,R,B) = \sum_{i=0}^{n} P(R^*=i|R,B;n)i = \sum_{i=0}^{n} P(R^*=i|R,B;n)i = \frac{\sum_{i=0}^{n} iA_i^R A_{n-i}^B}{\sum_{i=0}^{n} A_i^R A_{n-i}^B}$.

Since $A_i^j = 0$ when $i<j$, $r(n,R,B) = \frac{\sum_{i=R}^{n-B} iA_i^R A_{n-i}^B}{\sum_{i=R}^{n-B} A_i^R A_{n-i}^B}$.
$\overline{r}(n,R,B) = \frac{r(n,R,B)}{n}$, which gives Eq. (4).
This completes the proof.

Naturally, we have

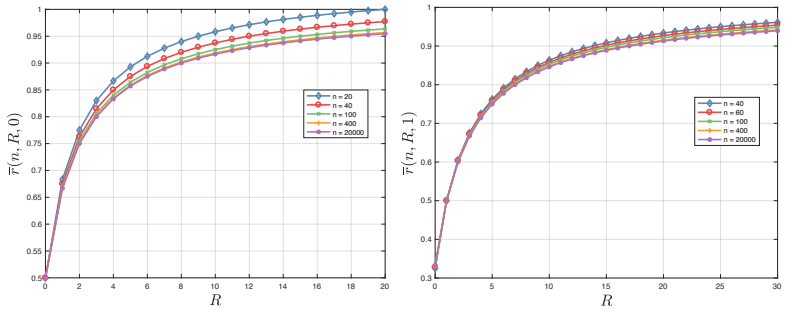
**Corollary 1.** *The expected proportion of negative instances is*

$$\overline{b}(n,R,B) = \overline{r}(n,B,R) = \frac{\sum_{i=B}^{n-R} iA_i^B A_{n-i}^R}{n\sum_{i=B}^{n-R} A_i^B A_{n-i}^R}. \tag{5}$$

Now we take some examples to have intuitive understanding. Figure 3(a) plots $\overline{r}(n,R,0)$ for $n=20$, 40, 100, 400, and $2\times10^4$. Here we observe that $\overline{r}(2\times10^4,R,0) > 0.95$ when $R \geq 18$. That is, only 18 instances are sufficient to evaluate a large unbalanced data block. It also shows that $\overline{r}(n,1,0)$ does not vary much with the change of $n$.

Figure 3(b) plots $\overline{r}(n,R,1)$ for $n=40$, 60, 100, 400, and $2\times10^4$. Here, we observe that with only one negative instance, the expected proportion of positive instances is significantly reduced.

We need to know the limit when $n\to+\infty$.
$\overline{r}(n,1,0) = \frac{\sum_{k=1}^{n} kA_k^1}{n\sum_{k=1}^{n} A_k^1} = \frac{1^2+2^2+\cdots+n^2}{n(1+2+\cdots+n)} = \frac{\frac{1}{6}n(n+1)(2n+1)}{\frac{1}{2}n^2(n+1)} = \frac{2}{3}+\frac{1}{3n}$.




(a) $B = 0$ (b) $B = 1$

Figure 3: $\overline{r}(n, R, B)$ wrt. the change of $R$

Hence $\lim_{n\to\infty} \overline{r}(n, 1, 0) = \frac{2}{3}$. Moreover,
$\overline{r}(n,2,0) = \frac{\sum_{k=2}^{n} kA_k^2}{n\sum_{k=2}^{n} A_k^2} = \frac{1^3+2^3+\cdots+n^3-\sum_{x=1}^{n} x^2}{n(1^2+2^2+\cdots+n^2-\sum_{x=1}^{n} x)} = \frac{\frac{1}{4}n^2(n+1)^2-\frac{1}{6}n(n+1)(2n+1)}{\frac{1}{6}n^2(n+1)(2n+1)-\frac{1}{2}n(n+1)}$
$= \frac{3n^2-n-2}{4n^2+10n}$. Hence $\lim_{n\to\infty} \overline{r}(n, 2, 0) = \frac{3}{4}$.

**Theorem 3.**

$$\lim_{n\to\infty} \overline{r}(n, R, 0) = \frac{R+1}{R+2}. \tag{6}$$

Proof. We know that $A_i^R = \Pi_{k=0}^{R-1}(i-k) = i^R + \sum_{k=0}^{R-1} c_k i^k$, where $c_k$ is independent of $i$. We also know that $\sum_{i=1}^{n} i^R = \sum_{k=1}^{R} \frac{S(R,k)A_{n+1}^{k+1}}{k+1}$, where $S(R,k)$ is the Stirling number.
$\lim_{n\to\infty} \overline{r}(n, R, 0)$
$= \lim_{n\to\infty} \frac{\sum_{i=R}^{n} iA_i^R}{n\sum_{i=R}^{n} A_i^R}$
$= \lim_{n\to\infty} \frac{\sum_{i=R}^{n} i(i^R+\sum_{k=0}^{R-1} c_k i^k)}{n\sum_{i=R}^{n}(i^R+\sum_{k=0}^{R-1} c_k i^k)}$
$= \lim_{n\to\infty} \frac{\sum_{i=R}^{n} i^{R+1}+\sum_{k=0}^{R-1} c_k \sum_{i=R}^{n} i^{k+1}}{n\sum_{i=R}^{n} i^R+n\sum_{k=0}^{R-1} c_k \sum_{i=R}^{n} i^k}$
$= \lim_{n\to\infty} \frac{\sum_{i=1}^{n} i^{R+1}+\sum_{k=0}^{R-1} c_k \sum_{i=1}^{n} i^{k+1}}{n\sum_{i=1}^{n} i^R+n\sum_{k=0}^{R-1} c_k \sum_{i=1}^{n} i^k}$
$= \lim_{n\to\infty} \frac{\sum_{k=1}^{R+1} \frac{S(R+1,k)A_{n+1}^{k+1}}{k+1}+\sum_{k=0}^{R-1} c_k \sum_{m=1}^{k+1} \frac{S(k,m)A_{n+1}^{m+1}}{m+1}}{n\sum_{k=1}^{R} \frac{S(R,k)A_{n+1}^{k+1}}{k+1}+n\sum_{k=0}^{R-1} c_k \sum_{m=1}^{k} \frac{S(k,m)A_{n+1}^{m+1}}{m+1}}$.

Obviously, both $\sum_{k=0}^{R-1} c_k \sum_{m=1}^{k+1} \frac{S(k,m)A_{n+1}^{m+1}}{m+1}$ and $n\sum_{k=0}^{R-1} c_k \sum_{m=1}^{k} \frac{S(k,m)A_{n+1}^{m+1}}{m+1}$ are polynomials of degree $R+1$, while both $\sum_{k=1}^{R+1} \frac{S(R+1,k)A_{n+1}^{k+1}}{k+1}$ and $n\sum_{k=1}^{R} \frac{S(R,k)A_{n+1}^{k+1}}{k+1}$ are polynomials of degree $R+2$.
In addition, we have $S(R,R) = S(R+1,R+1) = S(R-1,R-1) = 1$.
Thus $\lim_{n\to\infty} \overline{r}(n, R, 0) = \lim_{n\to\infty} \frac{\frac{S(R+1,R+1)A_{n+1}^{R+2}}{R+2}}{n\frac{S(R,R)A_{n+1}^{R+1}}{R+1}} = \frac{R+1}{R+2}$.

This completes the proof.

**Theorem 4.**

$$\overline{r}(n,R,0) > \overline{r}(n,2R-1,1). \tag{7}$$

PROOF. When $n$ is small, the inequalities can be validated by experiments.

Analogous to the proof of Eq. (6),

$$\begin{aligned}
&\lim_{n\to\infty} \overline{r}(n,2R-1,1)\\
&= \lim_{n\to\infty} \frac{\sum_{i=2R-1}^{n-1} iA_i^{2R-1}A_{n-i}^1}{n\sum_{i=2R-1}^{n-1} A_i^{2R-1}A_{n-i}^1}\\
&= \lim_{n\to\infty} \frac{\sum_{i=2R-1}^{n-1} i(n-i)i^{2R-1}}{n\sum_{i=2R-1}^{n-1}(n-i)i^{2R-1}}\\
&= \lim_{n\to\infty} \frac{n\sum_{i=1}^{n} i^{2R} - \sum_{i=1}^{n} i^{2R+1}}{n^2\sum_{i=1}^{n} i^{2R-1} - n\sum_{i=1}^{n} i^{2R}}\\
&= \lim_{n\to\infty} \frac{n\frac{S(2R,2R)A_{n+1}^{2R+1}}{2R+1} - \frac{S(2R+1,2R+1)A_{n+1}^{2R+2}}{2R+2}}{n^2\frac{S(2R-1,2R-1)A_{n+1}^{2R}}{2R} - n\frac{S(2R,2R)A_{n+1}^{2R+1}}{2R+1}}\\
&= \lim_{n\to\infty} \frac{\frac{1}{2R+1} - \frac{1}{2R+2}}{\frac{1}{2R} - \frac{1}{2R+1}}\\
&= \frac{R}{R+1}.
\end{aligned}$$

We also know that

$\lim_{n\to\infty} \overline{r}(n,R,0) = \frac{R+1}{R+2} > \lim_{n\to\infty} \overline{r}(n,2R-1,1) = \frac{R}{R+1}$.

This completes the proof.

Figure 4 illustrates the expect proportion of positive instances $\overline{r}(100,R,0)$, $\overline{r}(100,2R-1,1)$ and $\overline{r}(100,2R-2,2)$ wrt. the change of $R$. We observe that $\overline{r}(100,R,0) > \overline{r}(100,2*R-1,1)$. The gap between $\overline{r}(100,2*R-1,1)$ and $\overline{r}(100,2*R-2,2)$ is more obvious. Furthermore, the value of $\overline{r}$ becomes more and more stable wrt. the increase of $R$.

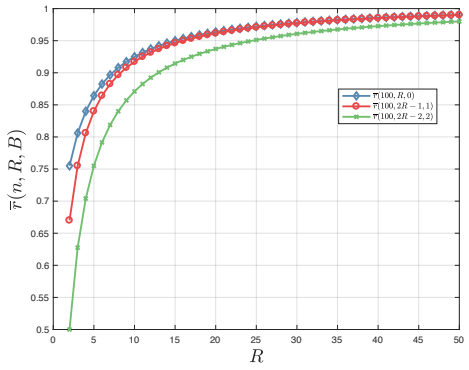
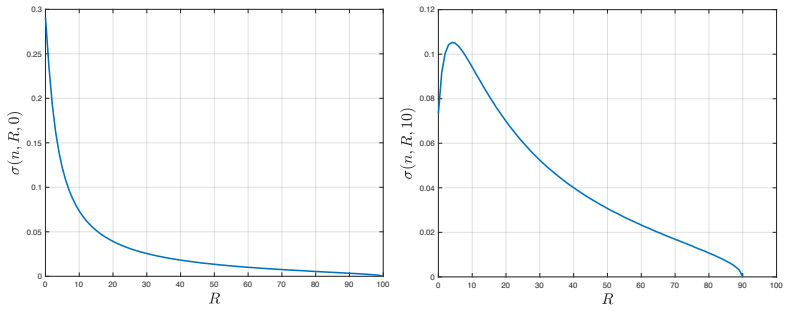



Figure 4: $\overline{r}(100,R,0)$, $\overline{r}(100,2R-1,1)$ and $\overline{r}(100,2R-2,2)$ wrt. the change of $R$

However, with only a few known labels, the estimation is not consistent. That is, the deviation is rather big.

**Theorem 5.** *The standard deviation of the proportion of positive instances is*

$$\sigma(n,R,B) = \sqrt{\sum_{i=0}^{n} P(R^*=i|R,B;n)[\frac{i}{n} - \overline{r}(n,R,B)]^2}. \tag{8}$$




(a) (b)

Figure 5: $\sigma(n, R, B)$ wrt. the change of $R$ where (a) $B = 0$, and (b) $B = 10$

For example, $\sigma(100, 0, 0) = 0.2915$, $\sigma(100, 2, 0) = 0.1937$, and $\sigma(100, 10, 0) = 0.073$.

Figure 5(a) illustrates that when $B = 0$, the standard deviation decreases with the increase of $R$. From Figure 5(b), it is observed that when $B = 10$, the standard deviation increases with the increase of $R$ from 0 to 4, then it decreases significantly.

## 5. Algorithm

In this section, we first discuss the algorithm framework. Then we describe major functions of the new algorithm. The implementation of the algorithm is available at http://www.fansmale.com/software.html. Finally we analyze the time complexity of the new algorithm.

### *5.1. Algorithm framework*

Our algorithm framework to Problem 1 has been illustrated in Figure 1. At the beginning, the whole universe is viewed as a block. Then a number of labels are queried. According to Theorem 2, once the queried labels are not the same, we should divide the block into two. Moreover, once there are enough number of positive (negative) labels, the other labels of the block can be predicted. This process terminates until each instance is queried or classified.

From the viewpoint of 3WD [73], there are three possible actions for an instance in each round. These are label query, label prediction and delay decision. From the viewpoint of sequential 3WD [27], impure blocks are split and the next round starts. The divide-and-conquer and GrC process executes until all labels have been queried or predicted.

Now we discuss the two key issues proposed in Section 1. The first key issue is addressed directly by Theorem 2. That is, once we have a different label, the block should be split. For the second key issue, we should consider Theorem 2, Corollary 1, and the optimization objective of Problem 1.

Given a block where all $R$ queried instances are positive, the expected total cost include the teacher cost and the expected misclassification cost. That is

$$f = tR + m(-,+)n[1 - \overline{r}(n, R, 0)]. \tag{9}$$

Similarly, if all $B$ queried instances are negative, the expected total cost is

$$f = tB + m(+,-)n[1 - \overline{b}(n, 0, B)]. \tag{10}$$

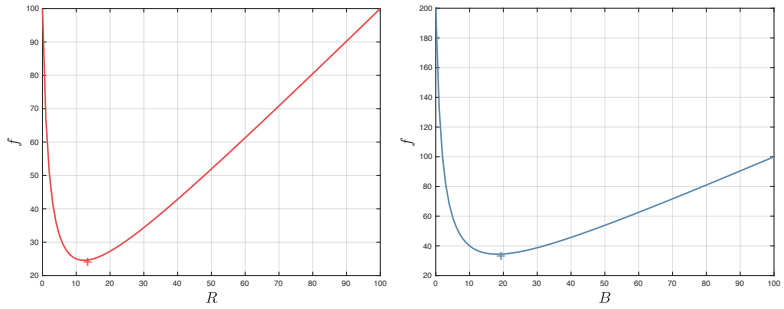
Figure 6 illustrates the expected cost wrt. the increase of the number of queried instances. The settings are as follows: $n = 100$, $t = 1$, $m(-,+) = 2$, and $m(+,-) = 4$. Here we observe that the function is unimodal.

To obtain the minimum expected total cost, the optimal number of queried instances is

$$s = \underset{R \textbf{ or } B}{\operatorname{argmin}} f, \tag{11}$$

which can be efficiently calculated by direct search.

There are two concrete issues while designing the algorithm: 1) How to split a block into two? and 2) Which instances should be queried? We will discuss them in Subsection 5.4.




(a) (b)

Figure 6: The expected total cost wrt. the increase of the number of queried instances

*5.2. Theoretical misclassification rate*

Now we analyze the theoretical misclassification rate.

**Theorem 6.** *Let the number of queried positive and negative instances in $X$ be $R$ and 0, respectively. The expected misclassification rate of $X$ is*

$$e_{-}^{+} = \frac{n[1 - \overline{r}(n, R, 0)]}{n - R}. \tag{12}$$

Proof. Since the expected proportion of positive instances is $\overline{r}(n, R, 0)$, the expected proportion of negative instances is $1 - \overline{r}(n, R, 0)$. In addition, there are $n$ instances in total and $R$ instances are already queried. Thus $e_{-}^{+} = \frac{n[1-\overline{r}(n,R,0)]}{n-R}$. This completes the proof.

Similarly, we have

**Corollary 2.** *Let the number of queried positive and negative instances for $X$ are 0 and $B$, respectively. The expected misclassification rate of $X$ is*

$$e_{+}^{-} = \frac{n[1 - \bar{b}(n, 0, B)]}{n - B}. \tag{13}$$

Note that in our scenario, one block is treated as *pure* iff enough queried instances belong to the same class. $e_{-}^{+}$ represents the misclassification rate of a *pure* positive block, and $e_{+}^{-}$ represents that of a *pure* negative one.

*5.3. Data organization with master tree*

We organize the data using a master tree, which is constructed with the density peaks clustering algorithm [50, 65].

Algorithm 1 lists the pseudo-code of master tree construction. The input of Algorithm 1 is cutoff distance $d_c$ and distance matrix $D$. The distance matrix $D$ is calculated from the original dataset, and $d_c$ is used to calculate the density array $\rho$. Line 3 sorts $\rho$ in descending order to obtain the indexed sequence $[o_i]_{N\times 1}$. In Line 4, $i$ starts with 2 instead of 1, because the master's index of max $\rho$ is set to $-1$ as default. Line 5 finds the instance with the smallest distance from the instance $o_i$. Lines 6 and 7 update $\delta_{o_i}$ and master's index $ms_{o_j}$, respectively.

**Algorithm 1** Master Tree Construction

**Input:** Cutoff distance $dc$ and distance matrix $D = [dist_{ij}]_{N\times N}$.
**Output:** Density array $\rho = [\rho_i]_{N\times 1}$, minimum distance array $\delta = [\delta_i]_{N\times 1}$ and master indices array $ms = [ms_i]_{N\times 1}$.
**Method:** buildMasterTree.

1: $\delta \leftarrow +\infty$, $ms \leftarrow -\mathbf{1}$ // Initialize
2: $\rho_i \leftarrow (\sum_j \chi(dist_{ij} - d_c)) - 1, i = 1, \ldots, N$ // Compute $\rho$
3: $[o_i]_{N\times 1} \leftarrow sort(\rho)$ // Obtain the index array according to $\rho$ in descending order
4: **for** $i \leftarrow 2$ to $N$ **do**
5: $\quad j^* \leftarrow \text{argmin}_{j\in\{1,\ldots,i-1\}} dist_{o_i,o_j}$ // Find the instance with minimum distance
6: $\quad \delta_{o_i} \leftarrow dist_{o_i,o_{j^*}}$
7: $\quad ms_{o_i} \leftarrow o_{j^*}$
8: **end for**
9: **return** $\{\delta, \rho, ms\}$

Example 2 illustrates how to build the master tree with the data listed in Table 2.

**Example 2.** *First, we build the Euclidean distance matrix between all the samples. Part of them are listed in Table 4. Second, we set hyper parameter*

Table 4: Exemplary distance matrix.

| $dist_{ij}$ | $x_1$ | $x_2$ | ... | $x_{19}$ | $x_{20}$ |
|---|---|---|---|---|---|
| $x_1$ | 0 | 0.7937 | ... | 2.7604 | 3.8079 |
| $x_2$ | 0.7937 | 0 | ... | 2.5475 | 3.3211 |
| ... | ... | ... | ... | ... | ... |
| $x_{19}$ | 2.7604 | 2.5475 | ... | 0 | 1.5362 |
| $x_{20}$ | 3.8079 | 3.3211 | ... | 1.5362 | 0 |

$d_c = 0.4608$. *According to Algorithm 1, we compute* $\rho$, $\delta$ *and* $ms$ *listed in Table 5. Finally,* $\rho \cdot \delta$ *are sorted in descending order and the rank array of all*

Table 5: Examples $\rho$, $\delta$ and $ms$.

| $U$ | $x_1$ | $x_2$ | ... | $x_7$ | ... | $x_{19}$ | $x_{20}$ |
|---|---|---|---|---|---|---|---|
| $\rho$ | 1 | 0 | ... | 4 | ... | 0 | 0 |
| $\delta$ | 0.1732 | 0.5477 | ... | 4.6076 | ... | 0.5831 | 0.5000 |
| $ms$ | $x_{10}$ | $x_3$ | ... | -1 | ... | $x_{14}$ | $x_{11}$ |

*instances is listed in Figure 1. The master tree is built according to* $ms$ *array. Here, the master of* $x_1$ *is* $x_{10}$ *and that of* $x_7$ *does not exist.*

### *5.4. The CADU algorithm*

Algorithm 2 lists the main process. Line 1 computes the distance matrix from the original dataset. Line 2 builds the master tree. Line 3 initializes the labels. Line 4 sorts $\rho \cdot \delta$ in descending order. Instances with larger $\rho_i \cdot \delta_i$ will be queried first. Line 5 queries the first instance to determine how many instances should be queried in the first iteration. Line 6 calls the recursive block splitting process, where the predicted and queried labels are gradually obtained.

**Algorithm 2** CADU

**Input:** A TMC-DS $S = (U, C, d, V, I, m, t)$ where labels are unknown, cutoff distance $d_c$.
**Output:** Predicted labels $L = [l_i]_{N\times 1}$, part of which are queried.
**Method:** activeLearning.

1: Compute distance matrix $D = [dist_{ij}]_{N\times N}$
2: $\{\delta, \rho, ms\} \leftarrow$ buildMasterTree$(d_c, D)$
3: $[l_i]_{N\times 1} \leftarrow -\mathbf{1}$ // Initialize all predict labels.
4: $b \leftarrow [b_1, \ldots, b_N] \leftarrow \text{sort}(\rho \cdot \delta)$ // Obtain the index sequence in descending order by $\rho \cdot \delta$.
5: query $l_{b_1}$
6: blockSplit$(b)$
7: **return** $L \leftarrow [l_i]_{N\times 1}$

Algorithm 3 lists the recursive block splitting process. It mainly receives an indexed array of instances sorted in descending order by $\rho \cdot \delta$. The array is iteratively split and the instances are predicted or queried during this process. Here, the instances in different sub-blocks are still sorted in $\rho \cdot \delta$ in descending order. Note that different blocks correspond to disjoint parts of the master tree. These parts are also organized as trees.

Line 1 acquires the number of instances that should be queried according to the first queried label. $s$ is the solution of Eq. (11). In Lines 2 through 10, important instances are queried one by one. If instance $x_{b_i}$ has a different label $l_{b_i}$ from the other instances, the block will be split immediately. Otherwise, we predict that the rest belongs to class $l_{b_1}$. Note that $b$ has similar meaning as $X$ discussed in Section 4. The difference lies in that $b$ is sorted.

**Algorithm 3** Recursive Block Splitting

**Input:** A sorted block $b \subseteq U$, the elements in $b$ are sorted by $\rho_{b_i} \cdot \delta_{b_i}$ in descending order.
**Method:** blockSplit.

```
1: s ← aquire(l_{b_1}, |b|, m, t)
2: for i ← 2 to s do
3:    query l_{b_i} if not queried
4:    if l_{b_i} ≠ l_{b_1} then
5:       split b into b' and b'', where b_1 is the root of b' and b_i is the root of b''
6:       blockSplit(b')
7:       blockSplit(b'')
8:       return
9:    end if
10: end for
11: ∀s + 1 ≤ k ≤ |b|, l_{b_k} ← l_{b_1}
```

### *5.5. Time complexity analysis*

Table 6 lists the time complexity of different steps of CADU. It takes $\Theta(|C|N^2)$ time to compute the distance matrix. Master tree construction takes $\Theta(N^2)$

Table 6: Time complexity of CADU.

| Description | Line | Complexity |
|---|---|---|
| Compute distances | 1 | $\Theta(\|C\|N^2)$ |
| Build master tree using Algorithm 1 | 2 | $\Theta(N^2)$ |
| Sort | 4 | $\Theta(NlogN)$ |
| Split block using Algorithm 3 | 6 | $O(N^2)$ |

time, and the sorting process takes $\Theta(NlogN)$ time by using Quicksort [16] or Mergesort [11]. The active learning process iteratively splits block, which

corresponds to traversing the master tree once. In each split, the optimization objective varies with the size of the block.

In the first iteration $n = N$. Then it decreases after block splitting. If $n$ is small, $\overline{r}(n, R, 0)$ can be directly calculated in linear time by Eq. (4), otherwise it can be approximated as $\frac{R+1}{R+2}$. In the best case, the optimization objective is calculated only once, and the time complexity of the active learning process is $\Theta(N)$. In the worst case, the optimization objective is calculated $N$ times where the size of a new block is 1 in each iteration. The corresponding time complexity is $\Theta(N^2)$. In summary, the time complexity of CADU is $\Theta(|C|N^2)$ and distance matrix computing is the performance bottleneck.

## 6. Experiments

In this section, we compare CADU with three sets of algorithms, namely active learning, cost-sensitive, and cost-sensitive active learning algorithms.

### *6.1. Dataset and cost setting*

Table 7 summarizes twelve benchmark datasets for our experimentation. These datasets include both real-world and artificial data. The real-world data are from different fields including botany, financial, computer, communication, biological, life and mass spectrometry. All of them can be downloaded from UCI machine learning repository [38] or IDA benchmark repository [1].

Table 7: Dataset information.

| ID | Dataset | $\|U\|$ | $\|C\|$ | Area |
|---|---|---|---|---|
| 1 | Allaml | 72 | 7129 | Biological |
| 2 | Arcene | 200 | 10000 | Mass Spectrometry |
| 3 | Banana | 5300 | 2 | Botany |
| 4 | Credit6000 | 5987 | 65 | Financial |
| 5 | German | 1000 | 20 | Financial |
| 6 | Heart | 270 | 13 | Life |
| 7 | Ionosphere | 351 | 34 | Physical |
| 8 | Jain | 373 | 2 | Artificial |
| 9 | Madelon | 2600 | 500 | Artificial |
| 10 | Sonar | 208 | 60 | Communication |
| 11 | Spambase | 4207 | 57 | Computer |
| 12 | Thyroid | 215 | 5 | Life |

We use the following cost settings: $m(-,+) = 2$, $m(+,-) = 4$, and $t = 1$. Since the average cost is $t$ when all instances are queried. Naturally, a cost-sensitive classification model is valid iff the average cost is less than $t$.

[1] http://www.raetschlab.org/Members/raetsch/benchmark

*6.2. Comparison with cost-insensitive active learning algorithms*

We study cost-insensitive active learning algorithms with three popular Query by Committee (QBC) strategies [2]:

1) VES: Vote entropy sampling;

2) CES: Consensus entropy sampling;

3) MDS: Max disagreement sampling.

The committee consists of three basic classifiers, namely Decision Tree [49], Random Forest [30] and SVM with RBF kernel [58, 12].

Table 8 compares CADU with these methods. The experiment is repeated five times with different data shuffling to get the average value. This is because the order may affect the results of some algorithms. We see that CADU outperforms others on 7 datasets, and the mean rank is the best. The average cost of CADU decreases by 11.9%, 10.5% and 16.2% compared with VES, CES and MDS, respectively.

Table 8: The average cost of CADU compared with cost-insensitive active learning algorithms.

| cost | Dataset | | | | | | | | | | | | MeanRank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Allaml | Arcene | Banana | Credit6000 | German | Heart | Ionosphere | Jain | Madelon | Sonar | Spambase | Thyroid | |
| VES | 1.0333 | 1.0970 | 0.3999 | 0.5900 | 0.8318 | 0.6896 | **0.4148** | 0.1164 | 1.0782 | **0.8952** | **0.3947** | 0.3005 | 2.0833 |
| CES | **0.7444** | 0.9990 | 0.5838 | 0.6182 | 0.8598 | 0.7193 | 0.5140 | 0.1303 | 1.1255 | 0.9452 | 0.4253 | **0.2781** | 2.7500 |
| MDS | 1.1500 | 1.0370 | 0.4478 | 0.6016 | 0.8742 | 0.7844 | 0.5470 | **0.1153** | 1.1702 | 0.9183 | 0.4606 | 0.3581 | 3.2500 |
| CADU | 0.8333 | **0.6850** | **0.3306** | **0.4154** | **0.6550** | **0.6630** | 0.7749 | **0.1153** | **0.9777** | 0.9087 | 0.7666 | 0.3023 | **1.8333** |

*6.3. Comparison with cost-sensitive learning algorithms*

We study the following three cost-sensitive supervised learning algorithms [3]:

1) CSLR: Example-dependent cost-sensitive logistic regression [2];

2) CSDT: Example-dependent cost-sensitive decision tree [4];

3) CSRF: Example-dependent cost-sensitive random forest [3].

These algorithms assume that the misclassification costs of each instance are different, thus they are the generalization of the cost-sensitive classification. Here, we set the misclassification costs for each instance to be the same.

The experiment is repeated five times with data shuffling. The number of training instances is the number of queried computed by CADU. In their initialization stage, we randomly query two instances with different labels. These instances serves as the initial training set.

Table 9 compares CADU with these algorithms. Here "-" indicates that the algorithm runs over 5 hours without final results. We see that CADU outperforms the others on 10 datasets. The $t$-test results of CADU versus the

[2] https://github.com/cosmic-cortex/modAL

[3] https://github.com/albahnsen/CostSensitiveClassification

Table 9: The average cost of CADU compared with cost-sensitive learning algorithms.

| cost | Dataset | | | | | | | | | | | | MeanRank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Allaml | Arcene | Banana | Credit6000 | German | Heart | Ionosphere | Jain | Madelon | Sonar | Spambase | Thyroid | |
| CSLR | 1.1944 | - | 1.8377 | 1.6395 | 1.2150 | 1.2407 | 0.8843 | 1.0375 | 1.7523 | 1.3644 | 1.6463 | 1.1805 | 3.9091 |
| CSDT | 1.1389 | 1.0350 | 0.7163 | 0.5927 | 1.0546 | 0.9593 | 0.6074 | 0.2836 | 1.3077 | 1.0644 | 0.4938 | 0.4995 | 2.5833 |
| CSRF | 1.4167 | 1.0150 | 1.5559 | 0.5100 | 1.1906 | 0.8896 | **0.5436** | 0.2450 | 1.1992 | 1.0375 | **0.3426** | 0.4679 | 2.1667 |
| CADU | **0.8333** | **0.6850** | **0.3306** | **0.4154** | **0.6550** | **0.6630** | 0.7749 | **0.1153** | **0.9777** | **0.9087** | 0.7666 | **0.3023** | **1.3333** |

other algorithms is based on 93% confidence level. The average cost of CADU decreases by 55.4%, 25.5% and 37.7% compared with CSLR, CSDT and CSRF, respectively.

### *6.4. Comparison with other cost-sensitive active learning algorithms*

Finally, we compare CADU with various cost-sensitive active learning algorithms. Four state-of-the-arts algorithms are compared [4]:

1) ALCE: Active learning with cost embedding [21];

2) CWMM: Cost-weighted minimum margin [8];

3) MEC: Maximum expected cost [8];

4) TALK: Tri-partition active learning through k-nearest neighbors [40].

For ALCE, CWMM and MEC, we repeat the experiment five times with data shuffling. Since the data order does not affect the results of TALK and CADU, we do not repeat the experiment. These four algorithms require an initial labeled set to train an basic classifier, while CADU does not. Two randomly selected positive and negative instances are used to train the basic classifier. They are treated as queried instances. Since the number of queries is a parameter in ALCE, CWMM and MEC, we set it to the CADU calculated value.

In the active learning process, we set all unlabeled instances as the training pool, and query a given number of instances to retrain the classifier. The remaining unlabeled instances are then classified and the total misclassification cost is calculated. Similar to CADU, the number of queried of TALK is determined by itself.

Table 10: The average cost of CADU compared with other cost-sensitive active learning algorithms.

| cost | Dataset | | | | | | | | | | | | MeanRank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Allaml | Arcene | Banana | Credit6000 | German | Heart | Ionosphere | Jain | Madelon | Sonar | Spambase | Thyroid | |
| ALCE | **0.6944** | 0.9250 | 0.4546 | 0.6486 | 0.6398 | **0.5533** | 1.0473 | 0.1174 | 1.4634 | 1.1221 | 0.9877 | **0.2260** | 2.2500 |
| CWMM | 1.9111 | 0.9250 | 0.6314 | 0.6649 | 0.6690 | 0.6659 | **0.5094** | 2.5003 | 1.4634 | 1.2990 | 1.0880 | 0.3042 | 3.8333 |
| MEC | 1.5111 | 0.9250 | 0.6735 | 0.6558 | 0.6666 | 0.5696 | 0.6541 | 0.3169 | 1.4634 | 1.2702 | 1.0897 | 0.2688 | 3.2500 |
| TALK | **0.6944** | 1.1200 | 0.8966 | **0.3207** | **0.6000** | 0.8889 | 1.2821 | 1.4799 | 1.0000 | 0.9327 | 0.7982 | 0.6047 | 3.1667 |
| CADU | 0.8333 | **0.6850** | **0.3306** | 0.4154 | 0.6550 | 0.6630 | 0.7749 | **0.1153** | **0.9777** | **0.9087** | **0.7666** | 0.3023 | **1.9167** |

Table 10 compares these cost-sensitive active learning algorithms. CADU outperforms others on six datasets. The $t$-test results of CADU versus the

[4]https://github.com/ej0cl6/csmcal

other cost-sensitive active learning algorithms is based on nearly 95% confidence level. The average cost of CADU decreases by 20.7%, 56.3%, 29.9% and 37.7% compared with ALCE, CWMM, MEC and TALK, respectively. The mean rank of CADU is also the best, demonstrating the superiority of CADU.

## 7. Conclusions and further works

This study has proposed a new cost-sensitive active learning algorithm called CADU. CADU explicitly builds a strong connection between active learning and sequential 3WD. To address a key issue of CADU, the LUD model is constructed to calculate the label distribution for each cluster. Experimental results show that CADU outperforms cost-insensitive active learning, cost-sensitive learning, and cost-sensitive active learning algorithms in terms of average cost.

The following research topics deserve further investigation:

1. New label distribution evaluation models for general classification problems. This study only considered the binary classification problems. For other classification problems we need more complicated models. New models for multi-label classification are also desired.

2. New label distribution evaluation models under different assumptions. The LUD model is based on Assumption 1, which takes into considering the quality of clustering. In reality, the quality of clustering depends not only on the algorithm but also on the data. More general assumptions can be made to adopt to different types of data.

3. New algorithms based on different clustering techniques. The CADU algorithm is based on the density peak clustering algorithm. Since different techniques are appropriate for different data, we can use other clustering techniques, or ensemble methods to enhance the clustering quality. In this way the total cost might be decreased further.

## Acknowledgements

This work is in part supported by the National Natural Science Foundation of China under Grant No. 61379089.

[1] A. Agarwal, Selective sampling algorithms for cost-sensitive multiclass prediction, in: ICML, 2013.

[2] A. C. Bahnsen, D. Aouada, B. Ottersten, Example-dependent cost-sensitive logistic regression for credit scoring, in: ICMLA, 2014.

[3] A. C. Bahnsen, D. Aouada, B. Ottersten, Ensemble of example-dependent cost-sensitive decision trees, arXiv preprint arXiv:1505.04637 (2015) 1–13.

[4] A. C. Bahnsen, D. Aouada, B. Ottersten, Example-dependent cost-sensitive decision trees, Expert Systems with Applications 42 (19) (2015) 6609–6619.

[5] M.-F. Balcan, A. Broder, T. Zhang, Margin based active learning, in: COLT, 2007.

[6] A. Bargiela, W. Pedrycz, Granular computing, in: Handbook on Computational Intelligence: Volume 1: Fuzzy Logic, Systems, Artificial Neural Networks, and Learning Systems, World Scientific, 2016, pp. 43–66.

[7] F. Cabitza, D. Ciucci, A. Locoro, Exploiting collective knowledge with three-way decision theory: cases from the questionnaire-based research, International Journal of Approximate Reasoning 83 (2017) 356–370.

[8] P.-L. Chen, H.-T. Lin, Active learning for multiclass cost-sensitive classification using probabilistic models, in: TAAI, 2013.

[9] E. K. P. Chong, S. H. Zak, An introduction to optimization, vol. 76, John Wiley & Sons, 2013.

[10] D. A. Cohn, Z. Ghahramani, M. I. Jordan, Active learning with statistical models, Journal of Artificial Intelligence Research 4 (1996) 129–145.

[11] R. Cole, Parallel merge sort, SIAM Journal on Computing 17 (4) (1988) 770–785.

[12] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000.

[13] B. Demir, L. Minello, L. Bruzzone, Definition of effective training sets for supervised classification of remote sensing images by a novel cost-sensitive active learning method, IEEE Transactions on Geoscience and Remote Sensing 52 (2) (2014) 1272–1284.

[14] X.-F. Deng, Y. Y. Yao, Decision-theoretic three-way approximations of fuzzy sets, Information Sciences 279 (2014) 702–715.

[15] X. Han, C. K. Kwoh, J. J. Kim, Clustering based active learning for biomedical named entity recognition, in: IJCNN, 2016.

[16] C. A. R. Hoare, Quicksort, The Computer Journal 5 (1) (1962) 10–16.

[17] B.-Q. Hu, Three-way decisions space and three-way decisions, Information Sciences 281 (2014) 21–52.

[18] B.-Q. Hu, H. Wong, K. C. Yiu, On two novel types of three-way decisions in three-way decision spaces, International Journal of Approximate Reasoning 82 (2017) 285–306.

[19] Q.-H. Hu, D.-R. Yu, Z.-X. Xie, J.-F. Liu, Fuzzy probabilistic approximation spaces and their information measures, IEEE Transactions on Fuzzy Systems 14 (2) (2006) 191–201.

[20] J.-J. Huang, J. Wang, Y. Y. Yao, N. Zhong, Cost-sensitive three-way recommendations by learning pair-wise preferences, International Journal of Approximate Reasoning 86 (2017) 28–40.

[21] K.-H. Huang, H.-T. Lin, A novel uncertainty sampling algorithm for cost-sensitive multiclass active learning, in: ICDM, 2016.

[22] S.-J. Huang, R. Jin, Z.-H. Zhou, Active learning by querying informative and representative examples, in: NIPS, 2010.

[23] C.-M. Jiang, Y. Y. Yao, Effectiveness measures in movement-based three-way decisions, Knowledge-Based Systems.

[24] J. Kang, K. R. Ryu, H. C. Kwon, Using cluster-based sampling to select initial training set for active learning in text classification, in: PAKDD, 2004.

[25] G. Klir, B. Yuan, Fuzzy sets and fuzzy logic, vol. 4, Prentice hall New Jersey, 1995.

[26] G. Krempl, T.-C. Ha, M. Spiliopoulou, Clustering-based optimised probabilistic active learning (COPAL), in: International Conference on Discovery Science, 2015.

[27] H.-X. Li, L.-B. Zhang, B. Huang, X.-Z. Zhou, Sequential three-way decision and granulation for cost-sensitive face recognition, Knowledge-Based Systems 91 (2016) 241–251.

[28] H.-X. Li, L.-B. Zhang, X.-Z. Zhou, B. Huang, Cost-sensitive sequential three-way decision modeling using a deep neural network, International Journal of Approximate Reasoning 85 (2017) 68–78.

[29] J.-H. Li, C.-C. Huang, J.-J. Qi, Y.-H. Qian, W.-Q. Liu, Three-way cognitive concept learning via multi-granularity, Information Sciences 378 (2017) 244–263.

[30] A. Liaw, M. Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22.

[31] A. Liu, G. Jun, J. Ghosh, Spatially cost-sensitive active learning, in: SIAM, 2009.

[32] D. Liu, D. Liang, Three-way decisions in ordered decision system, Knowledge-Based Systems 137 (2017) 182–195.

[33] M. Lorbach, R. Poppe, E. van Dam, L. Noldus, R. C. Veltkamp, Clustering-based active learning in unbalanced rodent behavior data, in: Proceedings of the International Workshop on VAIB, 2016.

[34] C. C. Loy, T. M. Hospedales, T. Xiang, S.-G. Gong, Stream-based joint exploration-exploitation active learning, in: CVPR, 2012.

[35] P. Mahajan, R. Kandwal, R. Vijay, General framework for cluster based active learning algorithm, International Journal on Computer Science & Engineering 3 (1) (2011) 307–312.

[36] D. D. Margineantu, Active cost-sensitive learning, in: IJCAI, vol. 5, 2005.

[37] A. K. McCallumzy, K. Nigamy, Employing em and pool-based active learning for text classification, in: ICML, 1998.

[38] C. J. Merz, UCI repository of machine learning databases, http://archive. ics. uci. edu/ml/datasets.html.

[39] D.-Q. Miao, Y. Zhao, Y. Y. Yao, H.-X. Li, F.-F. Xu, Relative reducts in consistent and inconsistent decision tables of the pawlak rough set model, Information Sciences 179 (24) (2009) 4140–4150.

[40] F. Min, F.-L. Liu, L.-Y. Wen, Z.-H. Zhang, Tri-partition cost-sensitive active learning through kNN, Soft Computing 7 (2017) 1–16.

[41] F. Min, Z.-H. Zhang, W.-J. Zhai, R.-P. Shen, Frequent pattern discovery with tri-partition alphabets, Information Sciences.

[42] A. Narr, R. Triebel, D. Cremers, Stream-based active learning for efficient and adaptive classification of 3d objects, in: ICRA, 2016.

[43] Z. Pawlak, Rough sets, International Journal of Computer & Information Sciences 11 (5) (1982) 341–356.

[44] W. Pedrycz, Shadowed sets: representing and processing fuzzy sets, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28 (1) (1998) 103–109.

[45] W. Pedrycz, B.-J. Park, S.-K. Oh, The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition, Pattern Recognition 41 (12) (2008) 3720–3735.

[46] J. Qian, C.-Y. Dang, X.-D. Yue, N. Zhang, Attribute reduction for sequential three-way decisions under dynamic granulation, International Journal of Approximate Reasoning 85 (2017) 196–216.

[47] Y.-H. Qian, J.-Y. Liang, C.-Y. Dang, Incomplete multigranulation rough set, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 40 (2) (2010) 420–431.

[48] Y.-H. Qian, J.-Y. Liang, W.-Z. Wu, C.-Y. Dang, Information granularity in fuzzy binary GrC model, IEEE Transactions on Fuzzy Systems 19 (2) (2011) 253–264.

[49] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.

[50] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[51] B.-B. Sang, Y.-T. Guo, D.-R. Shi, W.-H. Xu, Decision-theoretic rough set model of multi-source decision systems, International Journal of Machine Learning & Cybernetics (2017) 1–14.

[52] B. Settles, Active learning, Synthesis Lectures on Artificial Intelligence and Machine Learning 6 (1) (2012) 1–114.

[53] B. Settles, M. Craven, L. Friedland, Active learning with real annotation costs, in: Proceedings of the NIPS workshop on cost-sensitive learning, 2008.

[54] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, Proceedings of the 5th Annual Workshop on Computational Learning Theory (1992) 287–294.

[55] Y.-H. She, X.-L. He, H.-X. Shi, Y.-H. Qian, A multiple-valued logic approach for multigranulation rough set model, International Journal of Approximate Reasoning 82 (2017) 270–284.

[56] J. Smailović, M. Grčar, N. Lavrač, M. Žnidaršič, Stream-based active learning for sentiment analysis in the financial domain, Information sciences 285 (2014) 181–203.

[57] M. Sugiyama, S. Nakajima, Pool-based active learning in approximate linear regression, Machine Learning 75 (3) (2009) 249–274.

[58] J. A. K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (3) (1999) 293–300.

[59] C. A. Thompson, M. E. Califf, R. J. Mooney, Active learning for natural language parsing and information extraction, in: ICML, 1999.

[60] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, Journal of Machine Learning Research 2 (Nov) (2001) 45–66.

[61] P. D. Turney, Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm, Journal of Artificial Intelligence Research 2 (1995) 369–409.

[62] P. D. Turney, Types of cost in inductive concept learning, in: Proceedings of the Workshop on Cost-Sensitive Learning at the 17th ICML, 2002.

[63] G.-Y. Wang, L.-H. Guan, W.-Z. Wu, F. Hu, Data-driven valued tolerance relation based on the extended rough set, Fundamenta Informaticae 132 (3) (2014) 349–363.

[64] G.-Y. Wang, J. Yang, J. Xu, Granular computing: from granularity optimization to multi-granularity joint problem solving, Granular Computing 2 (3) (2017) 105–120.

[65] M. Wang, F. Min, Z.-H. Zhang, Y.-X. Wu, Active learning through density clustering, Expert Systems with Applications 85 (2017) 305–317.

[66] S. Wang, J.-J. Wang, X.-H. Gao, X.-Z. Wang, Pool-based active learning based on incremental decision tree, in: ICMLC, vol. 1, 2010.

[67] H. Y. Woo, C. H. Park, Active learning based on hierarchical clustering, KIPS Transactions on Software & Data Engineering 2 (10) (2013) 705–712.

[68] X. Yang, T.-R. Li, H. Fujita, D. Liu, Y. Y. Yao, A unified model of sequential three-way decisions and multilevel incremental processing, Knowledge-Based Systems 134 (2017) 172–188.

[69] J.-T. Yao, N. Azam, Web-based medical decision support systems for three-way medical decision making with game-theoretic rough sets, IEEE Transactions on Fuzzy Systems 23 (1) (2015) 3–15.

[70] J.-T. Yao, A. V. Vasilakos, W. Pedrycz, Granular computing: perspectives and challenges, IEEE Transactions on Cybernetics 43 (6) (2013) 1977–1989.

[71] Y. Y. Yao, Granular computing: basic issues and possible solutions, in: Proceedings of the 5th Joint Conference on Information Sciences, vol. 1, 2000.

[72] Y. Y. Yao, A partition model of granular computing, in: Transactions on Rough Sets I, Springer, 2004, pp. 232–253.

[73] Y. Y. Yao, Three-way decision: an interpretation of rules in rough set theory, in: RSKT, 2009.

[74] Y. Y. Yao, S. K. Wong, A decision theoretic framework for approximating concepts, International Journal of Man-Machine Studies 37 (6) (1992) 793–809.

[75] T. Young, Data mining and machine oriented modeling: A granular computing approach, Applied Intelligence 13 (2) (2000) 113–124.

[76] D. Yu, B. Varadarajan, L. Deng, A. Acero, Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion, Computer Speech & Language 24 (3) (2010) 433–444.

[77] H. Yu, X.-C. Wang, G.-Y. Wang, X.-H. Zeng, An active three-way clustering method via low-rank matrices for multi-view data, Information Sciences.

[78] H. Yu, Y. Wang, Three-way decisions method for overlapping clustering, in: RSCTC, 2012.

[79] S.-L. Yu, H. Zhao, Rough sets and laplacian score based cost-sensitive feature selection, PloS one 13 (2018) 1–23.

[80] X.-D. Yue, Y.-F. Chen, D.-Q. Miao, H. Fujita, Fuzzy neighborhood covering for three-way classification, Information Sciences.

[81] X.-D. Yue, Y.-F. Chen, D.-Q. Miao, J. Qian, Tri-partition neighborhood covering reduction for robust classification, International Journal of Approximate Reasoning 83 (2017) 371–384.

[82] L. A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (2) (1997) 111–127.

[83] C. Zhang, T. Chen, An active learning framework for content-based information retrieval, IEEE Transactions on Multimedia 4 (2) (2002) 260–268.

[84] H.-R. Zhang, F. Min, Three-way recommender systems based on random forests, Knowledge-Based Systems 91 (2016) 275–286.

[85] H.-R. Zhang, F. Min, B. Shi, Regression-based three-way recommendation, Information Sciences 378 (2017) 444–461.

[86] L. Zhang, B. Zhang, Fuzzy reasoning model under quotient space structure, Information Sciences 173 (4) (2005) 353–364.

[87] Q.-H. Zhang, Q. Xie, G.-Y. Wang, A novel three-way decision model with decision-theoretic rough sets using utility theory, Knowledge-Based Systems.

[88] Y. Zhang, J.-T. Yao, Gini objective functions for three-way classifications, International Journal of Approximate Reasoning 81 (2017) 103–114.

[89] Y. Zhang, J.-T. Yao, Game theoretic approach to shadowed sets: a three-way tradeoff perspective, Information Sciences.

[90] P.-L. Zhao, S. C. H. Hoi, Cost-sensitive online active learning with application to malicious URL detection, in: SIGKDD, 2013.

[91] B. Zhou, Y. Y. Yao, J.-G. Luo, Cost-sensitive three-way email spam filtering, Journal of Intelligent Information Systems 42 (1) (2014) 19–45.