IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 29, 2019, accepted April 20, 2019, date of publication May 1, 2019, date of current version May 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914263

Active Learning Through Multi-Standard Optimization

MIN WANG[1], YING-YI ZHANG[1], AND FAN MIN[2,3], (Member, IEEE)
[1]School of Electrical Engineering and Information, Southwest Petroleum University, Chengdu 610500, China
[2]School of Computer Science, Southwest Petroleum University, Chengdu 610500, China
[3]Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu 610500, China

Corresponding author: Fan Min (minfanphd@163.com)

This work was supported in part by the Natural Science Foundation of Sichuan Province under Grant 2017JY0190, in part by the Scientific Innovation Group for Youths of Sichuan Province under Grant 2019JDTD0017, in part by the State Administration of Work Safety project under Grant Sichuan-0008-2016AQ and Grant Sichuan-0009-2016AQ, and in part by the Ministry of Education Innovation Project under Grant 201801140013 and Grant 201801006094.

ABSTRACT Active learning selects the most critical instances and obtains their labels through interaction with an oracle. Selecting either informative or representative unlabeled instances may result in sampling bias or cluster dependency. In this paper, we propose a multi-standard optimization active learning (MSAL) algorithm that considers the informativeness, representativeness, and diversity of instances. Informativeness is measured by the soft-max predicted entropy, whereas representativeness is measured by the probability density function obtained by a non-parametric estimation. The multiplex of the two is used as an optimization objective to reduce model uncertainty and explore the distribution of unlabeled data. Diversity is measured by the difference between the selected critical instances. This is used as a constraint to prevent the selection of instances that are too similar. Learning experiments were performed with 12 datasets from various domains. The results of significance tests verify the effectiveness of MSAL and its superiority over state-of-the-art active learning algorithms.

INDEX TERMS Active learning, diversity, informativeness, representativeness.

I. INTRODUCTION

Active learning [1], [2] is a subfield of machine learning in which the algorithm is able to interactively query an *oracle* to obtain labels. It is widely employed in applications where the query incurs a heavy manual labeling cost [3], [4]. Initially, the training set is small, or even empty. Some critical instances are then selected and added to the training set to update the classifier. This process is repeated until the classifier achieves the desired accuracy or the maximum labeling cost is reached. Therefore, the active learner needs to consider a key issue: Which instances are critical?

Various active learning algorithms have been proposed to handle this issue. One popular approach is to query the most informative instances, such as in the query-by-committee [5], uncertainty sampling [6], and optimal experimental design [7] methods. Sun *et al.* [5] used a typical correlation analysis to find highly informative instances, while Tong and Chang [6] explored valuable instances through version space splitting. These approaches are unable to exploit the abundance of unlabeled data, making them prone to sample bias. Another direction in active learning is to select the instances that are most representative of the unlabeled data [8]–[10]. For example, Zhao *et al.* [8] utilized the structure information of unlabeled instances to choose representative samples. Wang *et al.* [11] built a master tree to express the cluster structure and designed a deterministic instance selection strategy. These approaches are heavily dependent on the quality of the clustering results.

Several active learning algorithms attempt to combine informativeness with representativeness to find the optimal query instances. Zhao *et al.* [8] proposed a sampling algorithm that exploits both the cluster information and the classification margins, and Donmez *et al.* [12] extended the active learning approach by dynamically balancing uncertainty and instance density. Huang *et al.* [13] developed a systematic approach for using the information of both labeled and unlabeled instances.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Guo.

In this paper, we propose a multi-standard optimization active learning (MSAL) algorithm that considers the informativeness, representativeness, and diversity of instances. The contributions of this paper are fourfold. First, we use the soft-max predictive entropy to measure the instance informativeness. Using the ''clustering by fast search and find of density peaks'' (CFDP) algorithm [14], we select the central instances as the initial training set for building a soft-max regression model. Soft-max regression is then used to obtain the probability that each instance belongs to each category. Finally, we calculate the information entropy for each instance.

Second, we adopt a non-parametric estimation method to obtain a probability density function for measuring the instance representativeness. Non-parametric estimation can be used with arbitrary distributions without assuming the form of the underlying density. We calculate the number of instances that fall into the window function. The Gaussian kernel function and window width are then selected. Finally, we estimate the probability density function with the statistical probability.

Third, we calculate the difference as a constraint and evaluate the diversity of critical instances. With the norm of the vector, we define the difference between the instances and set the difference threshold. The difference is used as a constraint to avoid choosing too many similar instances. An instance can only be queried if its difference is greater than the threshold. Finally, we define the diversity evaluation function.

Fourth, we design a multi-standard optimization active learning (MSAL) algorithm. Fig. 1 illustrates the MSAL process using a running example. The top part shows the input Seeds dataset, which is often used in standard machine learning tasks. The middle part shows the multi-standard optimization method, which considers the informativeness, representativeness, and diversity of instances. We choose the instance with the largest multiplex of informativeness and representativeness. If the instance satisfies the difference constraint, we query its label. Once the given N labels have been used, the loop terminates. The bottom part shows the output. With the selected critical instances, we use k-Nearest Neighbors (kNN) to classify the remaining instances. In this way, all labels are either queried or predicted.

Experiments are undertaken on 12 UCI datasets. Seven of the datasets are selected from different application areas (e.g., botany, material, iconology, and so on), and the other five are generated artificially. We compare the MSAL algorithm with popular classifiers and state-of-the-art active learning algorithms. We use a Friedman test and a Nemenyi post-hoc test to verify the significance of the differences between MSAL and the other algorithms. The results show that MSAL outperforms all of the other algorithms in terms of classification accuracy.

The remainder of this paper is organized as follows. In Section II, we briefly review three typical critical instance selection strategies. Section III presents a detailed description of the proposed approach. Section IV introduces the pseudocode of the MSAL algorithm and computes its time complexity. Section V presents and analyzes the experimental results, before Section VI summarizes our conclusions.

II. RELATED WORK

This section reviews three typical active learning critical instance selection strategies, namely informativeness-based [15], representativeness-based [11], [16], and hybrid selection strategies [13].

A. INFORMATIVENESS-BASED INSTANCE SELECTION STRATEGY

Querying the most informative instances is probably the most popular approach for active learning. Exemplar approaches include query-by-committee [15], [17], optimal experimental design [18], [19], and uncertainty sampling [3], [20]. Seung *et al.* [15] proposed the query-by-committee (QBC) algorithm. The word ''committee'' emphasizes that the choice of each critical instance is determined by a group of classifiers. The most inconsistent instances have the highest informativeness. To measure the level of disagreement, there are two main approaches. The first is the vote entropy [21]:

$$x^*_{VE} = \arg\max_x -\sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}, \tag{1}$$

where $V(y_i)$ is the number of ''votes'' that a label receives from among the committee members' predictions and C is the size of the committee.

Another proposed disagreement measure is the average Kullback–Leibler (KL) divergence [22]:

$$x^*_{KL} = \arg\max_x \frac{1}{C} \sum_{c=1}^{C} D(P_{\theta(c)} \| P_c), \tag{2}$$

where

$$D(P_{\theta(c)} \| P_c) = \sum_i P_{\theta(c)}(y_i|x) \log \frac{P_{\theta(c)}(y_i|x)}{P_c(y_i|x)}. \tag{3}$$

Here, $\theta^{(c)}$ represents a particular model in the committee and C represents the committee as a whole; thus, $P_c(y_i|x) = \frac{1}{C}\sum_{c=1}^{C} P_{\theta(c)}(y_i|x)$ is the ''consensus'' probability that y_i is the correct label.

The QBC algorithm suffers from high time complexity [17]. Gilad-Bachrach *et al.* [17] proposed the kernel QBC algorithm (KQBC) to decrease the runtime. Their key idea was to project the version space into a low-dimensional space.

B. REPRESENTATIVENESS-BASED INSTANCE SELECTION STRATEGY

Another school of active learning is to select the instances that are most representative of the unlabeled data. These approaches aim to exploit the clustering structure of unlabeled data [16], [23], generally using some clustering method. Zhao *et al.* [8] considered the clustering structure and selected the clustering centers for labeling, while

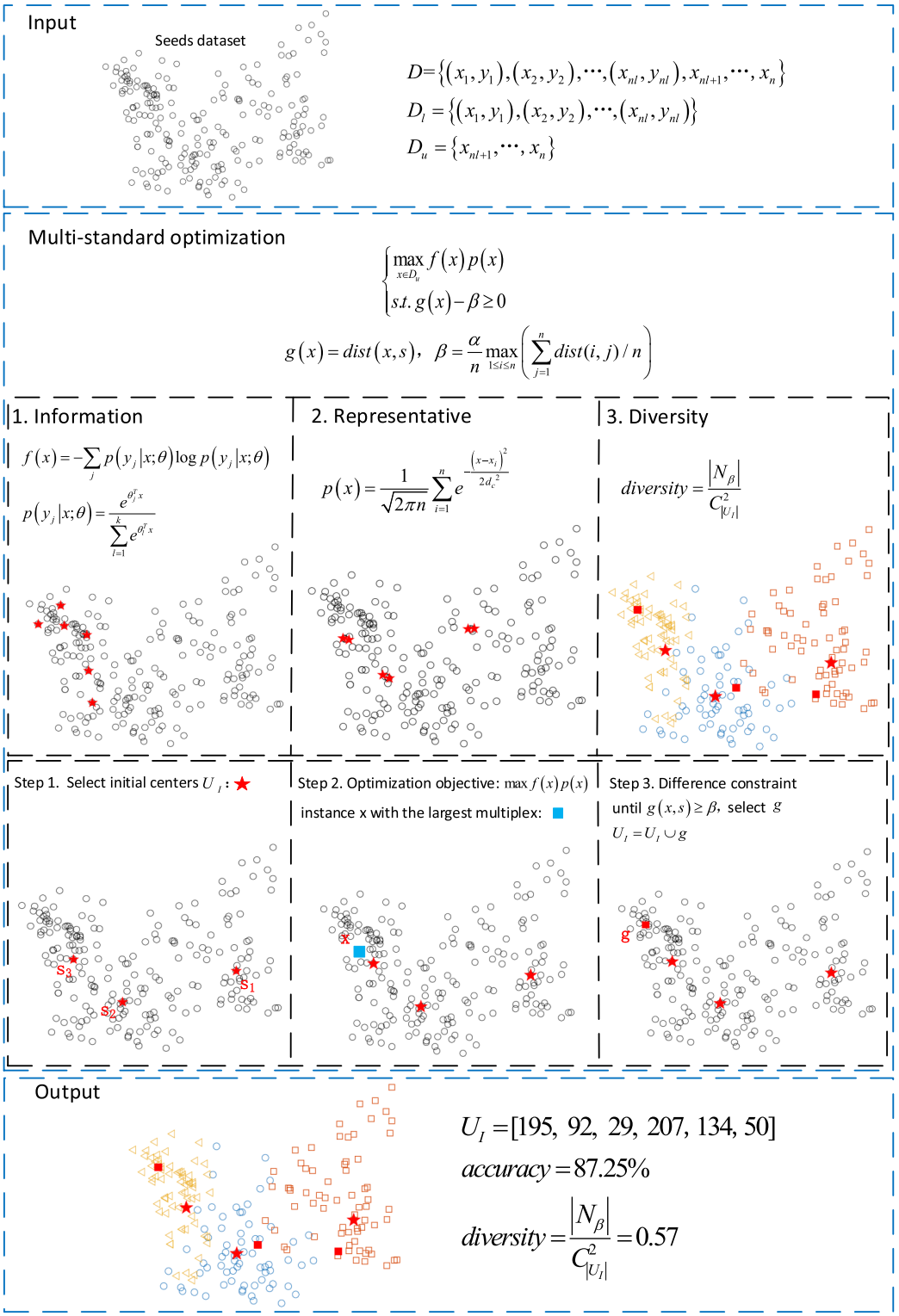

FIGURE 1. MSAL framework.

Nguyen and Smeulders [23] incorporated clustering into active learning and sampled a few randomly chosen instances in each cluster. Tuia *et al.* [24] searched the tree for pruning and sampled the most uncertain clusters.

Wang *et al.* [11] presented a typical clustering-based active learning algorithm (ALEC) for selecting instances with the largest density and distance product. ALEC includes three main stages. First, the dataset is clustered using CFDP [14].

No class labels are required at this time. Second, a deterministic strategy is employed for instance selection. In each cluster, the instances are sorted according to the multiplex of density and distance. The top $\sqrt{N}$ representative instances are selected for labeling, where N is the maximal number of labels provided by the oracle. Third, tri-partitioning [25], [26] is employed to determine the action to be taken on each instance during a specific iteration. If an instance is critical, it will be labeled by the oracle. If an instance is in a pure cluster, it will be classified directly. Otherwise, the algorithm waits for the next clustering operation.

C. HYBRID INSTANCE SELECTION STRATEGY

Several active learning algorithms [8], [12], [27], [28] have been proposed to find both informative and representative unlabeled instances. Huang *et al.* [13] proposed the query informative and representative instances (QUIRE) method. This selects an instance x_s from the pool of unlabeled data to query its class label. This criterion can be approximated by

$$s^* = \arg\min_{nl<s\leq n} L(D_l, D_u, y_u, x_s), \tag{4}$$

where

$$L(D_l, D_u, y_u, x_s) = \max_{y_s=\pm 1} \min_{f\in H} \frac{\lambda}{2}|f|_H^2 + \sum_{i=1}^{nl} l(y_i, f(x_i)). \tag{5}$$

Here, D_l denotes the labeled data and D_u denotes the unlabeled data, and y_l, y_s, y_u are the class labels assigned to D_l, x_s, D_u, respectively. The class assignment y_u is unknown. According to the manifold assumption, Huang *et al.* [13] expected a good solution for y_u to result in a small value of $L(D_l, D_u, y_u, x_s)$. Therefore, the solution for y_u can be obtained by minimizing $L(D_l, D_u, y_u, x_s)$.

III. PROBLEM STATEMENT AND ANALYSIS

In this section, we present a new constraint optimization problem that considers the informativeness, representativeness, and diversity of the instances.

Table 1 lists the notation used throughout this paper.

A. PROBLEM STATEMENT

We consider the case relevant to some active learning applications in which an oracle provides a fixed number of labels. Let N be the number of labels that the *oracle* can provide. For example, for an actual label task, the total label budget may be \$100k and each label costs \$1k, so $N = 100$. The key to the question is: How can we select the most critical N instances so as to obtain the highest classification accuracy?

The dataset is denoted by $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{nl}, y_{nl}), x_{nl+1}, \cdots, x_n\}$, where each instance $x_i = [x_{i1}, x_{i2}, \cdots, x_{id}]^T$ is a d-dimensional vector and $y_i \in \{1, 2, \cdots, k\}$ is the class label of x_i. D includes the training set $D_l = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{nl}, y_{nl})\}$ and the unlabeled set $D_u = \{x_{nl+1}, \cdots, x_n\}$. On each iteration, the active learner selects one instance x_s from the unlabeled set D_u and queries its label. When the number of queries reaches N, the process terminates.

TABLE 1. Notation.

Notation	Meaning
D	Dataset $D = \{(x_1, y_1), \ldots, (x_{nl}, y_{nl}), x_{nl+1}, \ldots, x_n\}$
n	The number of instances
N	The number of labels that the oracle can provide
k	Number of categories
d	Data dimension
g	Instance that satisfies the different constraint
s	The last critical instance that has been selected
$f(x)$	Informativeness of x
$p(x)$	Representativeness of x
$g(x)$	Constraint function of x
U_{I}	The instances labeled by the oracle
U_{II}	The instances classified by the active learner
β	Difference threshold
N_β	The significant difference set
$div(U_{\mathrm{I}})$	The diversity evaluate function

We define the following constraint optimization problem:

$$\begin{cases} \max\limits_{x\in D_u} f(x)p(x), \\ s.t.\ g(x) - \beta \geq 0, \end{cases} \tag{6}$$

where $f(x)$ denotes the informativeness, $p(x)$ denotes the representativeness, $g(x)$ denotes the difference, and β is the difference threshold.

This problem provides a systematic way for combining the informativeness, representativeness, and diversity of an instance. We will present specific functions for this problem in the following subsections.

B. INFORMATIVENESS

Informativeness can be used to reduce the uncertainty of the model. We use the information entropy [29] to measure the informativeness of $x \in D_u$. This is defined as

$$f(x) = -\sum_j P(y_j|x;\theta)\log P(y_j|x;\theta), \tag{7}$$

where $P(y_j|x;\theta)$ indicates the probability that instance x belongs to class j. Considering the multi-classification problem, we use soft-max regression to obtain $P(y_j\,|x\,;\theta)$. Given any instance x, the conditional probability of x belonging to y_j is

$$P(y_j|x;\theta) = \frac{e^{\theta_j^T x}}{\sum_{l=1}^k e^{\theta_l^T x}}. \tag{8}$$

The key to calculating $P(y_j|x;\theta)$ is to determine the parameter θ. The solution process mainly includes the following three steps.

Step 1: Determine the cost function $J(\theta)$.

The cost function $J(\theta)$ represents the deviation between the predicted value and the true value. The cost function is

$$J(\theta) = -\frac{1}{nl}\left[\sum_{i=1}^{nl}\sum_{j=1}^{k} 1\{y_i = j\}\log\frac{e^{\theta_j^T x_i}}{\sum_{l=1}^k e^{\theta_l^T x_i}}\right], \tag{9}$$

where $i \in \{1, 2, \cdots, nl\}, j \in \{1, 2, \cdots, k\}$, i represents the ith instance, and j represents the category. $1\{\cdot\}$ is an indicative function, that is, when the argument is true, the result is 1; otherwise, the result is 0.

Step 2: Solving the cost function $J(\theta)$ to obtain the optimal parameter θ. Minimizing the cost function, we obtain the optimal model parameter θ. We use an iterative optimization algorithm such as gradient descent [30] or Quasi-Newton [31] to solve $J(\theta)$. After some derivation, we obtain the gradient

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{nl} \sum_{i=1}^{nl} [x_i(1\{y_i = j\} - P(y_i = j \,|x_i; \theta))], \quad (10)$$

which represents the partial derivative of the loss function. By iteratively solving for the parameters θ_j, we obtain the model parameters

$$\theta_j := \theta_j - \alpha' \nabla_{\theta_j} J(\theta), \quad j = 1, 2, \ldots, k, \quad (11)$$

which converges to $J(\theta)$, where α' is the step size.

Step 3: Compute the hypothesis function $h_\theta(x)$. For each input x, the hypothesis function gives the probability value for each class j, i.e., $P(y_j|x; \theta), j \in \{1, 2, \ldots, k\}$. The hypothesis function is

$$h_\theta(x) = \begin{bmatrix} p(y) = 1|x; \theta \\ p(y) = 2|x; \theta \\ \vdots \\ p(y) = k|x; \theta \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x}} \begin{bmatrix} e^{\theta_1^T x} \\ e^{\theta_2^T x} \\ \vdots \\ e^{\theta_k^T x} \end{bmatrix}, \quad (12)$$

where $\theta_1, \theta_2, \ldots, \theta_k \in R^{d+1}$ are the model parameters, and $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x}}$ is used to normalize the probability distribution.

C. REPRESENTATIVENESS

Representativeness can be used to represent the overall feature of all unlabeled data. We use the probability density function to estimate representativeness. Traditional parameter estimation methods assume the parameters of the probability density function [32]. However, we cannot make accurate assumptions because this function may have various forms. In particular, the parameter of all classical density functions has a single-mode form [33], whereas the real situation is usually multimodal. Therefore, we use a non-parametric approach to obtain the probability density function $p(x)$. This can handle arbitrary probability distributions regardless of the form of the parameters.

Proposition 1: Give the dataset $D = \{x_1, x_2, \ldots, x_n\}$, window bandwidth d_c, and volume $V_n = \frac{1}{\sqrt{n}}$, the probability density of any instance x is

$$p(x) = \frac{1}{\sqrt{2\pi n}} \sum_{i=1}^{n} e^{-\frac{(x-x_i)^2}{2d_c^2}}. \quad (13)$$

Proof: The probability that instance x will fall into a region R is given by

$$P = \int_R p(x)dx, \quad (14)$$

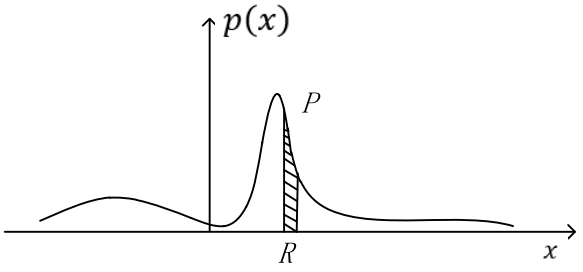

FIGURE 2. Probability density function $p(x)$.

which is a smoothed or averaged version of the density function $p(x)$. We can obtain the probability density function $p(x)$ by estimating the probability P. The detailed process is illustrated in Fig. 2 and (15)–(20).

The probability that k of these n fall into R is given by the binomial law

$$P_k = C_n^k P^k (1 - P)^{(n-k)}. \quad (15)$$

According to the properties of the binomial law, the mean of the k is

$$E(k) = k = nP. \quad (16)$$

When the value of n is sufficiently large,

$$P = \frac{k}{n}. \quad (17)$$

The ratio k/n will be a very good estimate for the probability P. Therefore, we have

$$P = \int_R p(x)dx = \frac{k}{n}, \quad (18)$$

where R is an area containing the instance x with a volume of V_n. If we assume that $p(x)$ is continuous and that the region R is so small that $p(x)$ does not vary appreciably within it, then

$$\int_R p(x)dx \approx p(x)V_n. \quad (19)$$

According to (17)–(19), we can estimate the probability density as

$$p(x) = \frac{k}{nV_n}. \quad (20)$$

Next, we fix V_n to determine k from the data. This leads to a kernel density estimation (KDE), such as the Parzen window estimation method [34]. The kernel function $\phi(\frac{x-x_i}{d_c})$ is

$$\phi(\frac{x - x_i}{d_c}) = \begin{cases} 1, & \left|\frac{x - x_i}{d_c}\right| \le \frac{1}{2}, \quad i = 1, 2, \ldots, n; \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

The quantity $\phi((x - x_i)/d_c)$ is equal to one if x_i is inside a hypercube with side d_c centered on x, and zero otherwise. The total number of instances k inside the hypercube is

$$k = \sum_{i=1}^{n} \phi(\frac{x - x_i}{d_c}). \quad (22)$$

We adopt the Gaussian kernel function

$$\phi(\frac{x-x_i}{d_c}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-x_i}{d_c})^2}, \tag{23}$$

where d_c is the window bandwidth, $d_c = 0.1 \max(dist(i,j))$, $1 \le i \le n, 1 \le j \le n$.

Let the volume $V_n = \frac{1}{\sqrt{n}}$. The probability density of any instance x is

$$\begin{aligned} p(x) &= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-x_i}{d_c})^2} \\ &= \frac{1}{\sqrt{2\pi n}} \sum_{i=1}^{n} e^{-\frac{(x-x_i)^2}{2d_c^2}}. \end{aligned} \tag{24}$$

This completes the proof.

D. DIVERSITY

Diversity is used to ensure that selected critical instances cover all categories, thus improving the classification accuracy. It is measured by the difference among the currently selected critical instances. The diversity evaluation function has three aspects.

1) THE DISTANCE METRIC

Given the dataset $D = \{x_1, x_2, \ldots, x_n\}$, the *distance* between x_i and x_j is

$$dist(i,j) = \sqrt[p]{\sum_{k=1}^{d} \left|x_{ik} - x_{jk}\right|^p}. \tag{25}$$

We use the norm of the vector to calculate the distance between x_i and x_j. For differently distributed datasets, p will take a different value. When $p = 1$, $dist(i,j)$ is the Manhattan distance between x_i and x_j; when $p = 2$, $dist(i,j)$ is the Euclidean distance.

2) THE CONSTRAINT FUNCTION $G(X)$

As active learning is a process, the computation of $g(x)$ relies on critical instances that have been queried. Let s be the last critical instance that has been queried. We define

$$g(x) = dist(x,s), \tag{26}$$

and so the constraint $g(x) - \beta \ge 0$ in (6) is equivalent to $dist(x,s) \ge \beta$. Only if $g(x) = dist(x,s) > \beta$ can the instance be queried.

Next, we set the *difference threshold*

$$\beta = \frac{\partial}{n} \max_{1 \le i \le n} \sum_{j=1}^{n} dist(i,j), \tag{27}$$

where ∂ is a coefficient, usually taken as 0.5.

3) THE DIVERSITY OF INSTANCES

Each selected critical instance satisfies the difference constraint, ensuring some diversity among the entire set of critical instances. We construct a significant difference set N_β.

Definition 1: Let U_{I} be the critical instance set. The significant difference set with respect to the threshold β is

$$N_\beta = \{\langle x_i, x_j\rangle \in U_{\mathrm{I}} \times U_{\mathrm{I}} \,|dist(i,j) > \beta\}. \tag{28}$$

Next, we evaluate the diversity of the entire set of critical instances. $div(U_I)$ is the fraction of instance pairs on U_I that satisfies the difference constraint.

Definition 2: The diversity of the critical instance set U_I is

$$div(U_{\mathrm{I}}) = \frac{\left|N_\beta\right|}{C^2_{|U_{\mathrm{I}}|}}, \tag{29}$$

where $C^2_{|U_{\mathrm{I}}|} = \frac{|U_{\mathrm{I}}||U_{\mathrm{I}}-1|}{2}$ is the permutation of U_{I}.

IV. MSAL ALGORITHM

This section presents the MSAL algorithm. First, we describe the process of the MSAL algorithm, and then we elaborate on two key sub-problems. Finally, we analyze the time complexity.

A. ALGORITHM DESCRIPTION

Lines 1 and 2 of Algorithm 1 correspond to the initialization stage. The set of instances labeled by the oracle is $U_{\mathrm{I}} = \emptyset$, and the set of instances classified by the classifier is $U_{\mathrm{II}} = U$. Line 3 performs CFDP clustering and updates U_{I}. We select the k centers as the initial training set. Line 4 updates U_{I} to be the set of k cluster centers. Line 5 updates the set U_{II}, $U_{\mathrm{II}} \leftarrow U - U_{\mathrm{I}}$. Line 6 records the last instance of U_{I}.

Lines 7–19 select critical instances. For each instance, we consider informativeness, representativeness, and diversity. The loop terminates when no more labels are available ($|U_{\mathrm{I}}| \ge N$). Line 8 trains the model parameters θ based on the initial training set U_{I}. Line 11 calculates the information entropy according to (7). Line 12 calculates the probability density function according to (13). Line 14 calculates the product of $f(x)$, $p(x)$ and sorts the results. Line 15 considers the difference constraint using $constraint(o, s, \beta)$, a function that computes the constraint condition. The input includes the sorted objective vector $[o]_{1\times|U_{\mathrm{II}}|}$, the last instance s in U_{I}, and the difference threshold β. The output is an instance g that satisfies the constraint. g is the critical instance to be queried, $g \in o$. Lines 17–18 update s, U_{I}, and U_{II}.

Finally, Lines 20–21 classify the instances in U_{II} using kNN.

B. TWO SUB-PROBLEMS IN ALGORITHM DESIGN

The MSAL algorithm selects critical instances based on informativeness, representativeness, and diversity. We now address the following two sub-problems: 1) How can non-unique solutions be avoided? 2) How should overflow and underflow situations be handled?

Algorithm 1 Multi-Standard Optimization Active Learning (MSAL)

Input: The dataset D, the cutoff distance d_c, the maximal number of labels provided by the oracle N, and the diversity threshold β.

Output: Predicted label $L \leftarrow [l_i]_{n\times 1}$.

$U_{\mathrm{I}} = \emptyset$; $U_{\mathrm{II}} = U$;
$[l_i]_{n\times 1} \leftarrow -1$;
//Step 1. Select the initial training set U_I.
$[c_1, c_2, \ldots, c_k] \leftarrow densityClustering(D, d_c)$;
$U_{\mathrm{I}} \leftarrow [c_1, c_2, \ldots, c_k]$;
$U_{\mathrm{II}} \leftarrow U - U_{\mathrm{I}}$;
$s \leftarrow c_k$; //s is the last critical instance, $s \in U_{\mathrm{I}}$
//Step 2. Select critical instances.
while $(|U_{\mathrm{I}}| < N)$ **do**
  $[\theta]_{k\times(d+1)} \leftarrow softmaxTrain(U_{\mathrm{I}})$;
  for $(i \leftarrow 1$ to $|U_{\mathrm{II}}|)$ **do**
    //Step 2.1. Compute informativeness
    $f_i \leftarrow -\sum_j P(y_j|x_i;\theta)\log P(y_j|x_i;\theta)$;
    //Step 2.2. Compute representativeness
    $p_i = \frac{1}{\sqrt{2\pi}|U_{\mathrm{II}}|}\sum_{j=1}^{|U_{\mathrm{II}}|} e^{-\frac{(x_i-x_j)^2}{2d_c^2}}$;
  end for
  $[o]_{1\times|U_{\mathrm{II}}|} \leftarrow sort(f \cdot p)$; //$o_1 \geq o_2 \geq \cdots \geq o_{|U_{\mathrm{II}}|}$
  //Step 2.3. Compute the difference constraint
  $g \leftarrow constraint(o, s, \beta)$; // g is the instance that satisfies the constraint.
  query l_g;
  $U_{\mathrm{I}} \leftarrow U_{\mathrm{I}} \cup g$, $U_{\mathrm{II}} \leftarrow U_{\mathrm{II}} - g$; // Update U_{I} and U_{II}
  $s \leftarrow g$; // Update s
end while
//Step 3. Classify the remaining instances.
$[l_i]_{n\times 1} \leftarrow kNNClassify(U_{\mathrm{I}}, U_{\mathrm{II}})$;
return $L \leftarrow [l_i]_{n\times 1}$;

1) NON-UNIQUE SOLUTIONS

Proposition 2: The optimization parameter θ is not the only solution.

Proof: Let θ_j be an optimal solution. We have

$$P(y_j|x;\theta) = \frac{e^{\theta_j^T x}}{\sum_{l=1}^{k} e^{\theta_l^T x}}. \tag{30}$$

According to the operational rules of exponential functions,

$$\frac{e^{(\theta_j-\varphi)^T x}}{\sum_{l=1}^{k} e^{(\theta_l-\varphi)^T x}} = \frac{e^{\theta_j^T x}e^{-\varphi^T x}}{\sum_{l=1}^{k} e^{\theta_l^T x}e^{-\varphi^T x}} = \frac{e^{\theta_j^T x}}{\sum_{l=1}^{k} e^{\theta_l^T x}}. \tag{31}$$

Hence, $\theta_j - \varphi$ is also an optimal solution. Thus, the solution θ_j is not unique.

This completes the proof.

The loss function at this time is not strictly non-convex, as there is a "flat" area near the local minimum point.

TABLE 2. Computational complexity of Algorithm 1.

Lines	Complexity	Description
Lines 3–6	$O(dn^2)$	Select the initial training set U_{I}
Lines 7-19	$O(Nn'^2)$	Select critical instances
Lines 20-21	$O(dn)$	Classify the instances in U_{II}
Total	$O(dn^2) + O(n) + O(dn) = O(dn^2)$	

$n' = |U_{\mathrm{II}}|$, $n' < n$.

In practice, the usual approach is to add the weight attenuation term $\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{d}\theta_{ij}^2$ [35]. The improved cost function

$$J(\theta) = -\frac{1}{nl}\left[\sum_{i=1}^{nl}\sum_{j=1}^{k} l\{y_i = j\}\log\frac{e^{\theta_j^T x_i}}{\sum_{l=1}^{k} e^{\theta_l^T x_i}}\right] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{d}\theta_{ij}^2 \tag{32}$$

is a strictly convex function. We can then guarantee a unique solution. The gradient descent method [36] guarantees convergence to the global optimal solution.

2) OVERFLOW AND UNDERFLOW

When we calculate $P(y_j|x;\theta)$ in Algorithm 1, overflow and underflow may occur. Underflow occurs when numbers close to zero are rounded to zero [37], and overflow occurs when numbers with large magnitudes are approximated as ∞ or $-\infty$ [37]. Both overflow and underflow can be resolved by instead evaluating $e^{\theta_j^T x - M}$, $M = \max(\theta_j^T x), j \in 1, 2, \ldots, k$ [38]. That is, M is the largest of all $\theta_j^T x$.

$$\frac{e^{\theta_j^T x - M}}{\sum_{l=1}^{k} e^{\theta_l^T x - M}} = \frac{\frac{e^{\theta_j^T x}}{e^M}}{\frac{\sum_{l=1}^{k} e^{\theta_l^T x}}{\sum_{l=1}^{k} e^M}} = P(y_j|x;\theta). \tag{33}$$

For any x, if we subtract M, the maximum value of the exponential function is zero. Therefore, no overflow will occur. Additionally, the denominator contains at least one item with a value of 1. Therefore, the denominator will not suffer from underflow.

C. COMPLEXITY ANALYSIS

The time complexity of the MSAL algorithm is analyzed in Table 2.

V. EXPERIMENTS

We conducted experiments to analyze the effectiveness of the MSAL algorithm and answer the following questions:

1) Is the MSAL algorithm more accurate than supervised classification algorithms such as kNN, J48 (C4.5), Random Forest, and Bagging?
2) Is the MSAL algorithm more accurate than state-of-the-art active learning algorithms, including informativeness-based KQBC, representativeness-based ALEC, and the hybrid QUIRE?
3) Is the MSAL algorithm efficient?

TABLE 3. Dataset information.

ID	Name	Source	Domain	n	d	k
1	Appendicitis	KEEL	Medical	106	7	2
2	Iris	UCI	Botany	150	4	3
3	Sonar	UCI	Physical	208	60	2
4	Seeds	UCI	Biological	210	7	3
5	Haberman	UCI	Life	306	3	2
6	Ionosphere	UCI	Physical	351	34	2
7	Compound	Synthetic	N/A	399	2	6
8	R15	Synthetic	N/A	600	2	15
9	D31	Synthetic	N/A	3100	2	31
10	Banana	Synthetic	N/A	5300	2	2
11	Texture	UCI	Material	5500	40	11
12	Twonorm	KEEL	Historical	7400	20	2

The computations were performed on a Windows 10 64-bit operating system with 8 GB RAM and Intel (R) Core 2Quad CPU Q9500@2.83 GHz processors using Matlab software. The source code is available at *www.fansmale.com/software.html*.

A. DATASETS

Table 3 summarizes the 12 datasets used in our experiments. These include four artificial datasets obtained from [14], six from the University of California at Irvine (UCI) ML repository [39], and two obtained from [40]. The number of instances ranged from 106–7400, the number of attributes ranged from 2–60, and the number of classes ranged from 2–31. The domains of the data are listed in the table.

The performance of an active learner is evaluated by the accuracy:

$$accuracy = \frac{n - N - e}{n - N} \times 100\%, \tag{34}$$

where e is the number of misclassified instances.

B. TWO-DIMENSIONAL VISUALIZATION OF THE SELECTED CRITICAL INSTANCES

Fig. 3 shows a two-dimensional visualization of the selected critical instances. We compare this against the following three active learning algorithms. Detailed descriptions of these methods can be found in Section II.

1) Informativeness-based algorithm: Kernel query by committee (KQBC) [17];
2) Representativeness-based algorithm: Active learning through density clustering (ALEC) [11]; [1]
3) Hybrid algorithm: Active learning by querying informative and representative examples (QUIRE) [13] [2].

We consider four typical datasets: Iris, Seeds, Compound, and R15. Iris and Seeds are probably the most well-known datasets in the pattern recognition and classification community [41]. Iris contains three classes. One of these classes is linearly separable from the other two, whereas the latter two are not linearly separable from each other. The Seeds dataset includes kernels belonging to three different wheat varieties: Kama, Rosa and Canada, each with 70 elements. Compound and R15 are synthetic datasets with typical shape distributions [14]. Compound contains petals and scatter patterns, whereas R15 consists of three rings formed by 15 evenly distributed clusters.

KQBC tends to select the most informative instances. Figs. 3(e) and 3(i) show some selected instances near the edge of the classification. This suggest that outliers are easily selected. ALEC selects representative instances based on clusters. The performance is heavily dependent on the quality of the clustering results. For the Iris and Seeds datasets, the representative instances selected by ALEC do not cover all categories. The QUIRE approach favors both representative and informative instances. This approach may select too many similar instances in the same cluster, ignoring small clusters of lower density. For Iris, Compound, and R15, QUIRE cannot avoid selecting instances that are too similar. For example, the distance between the two closest instances in the Iris and R15 datasets is only 0.2 and 0.3, respectively. To effectively reduce the number of queries, such similar instances should not be selected. MSAL considers the informativeness, representativeness, and diversity of instances. The selected critical instances are evenly distributed across different categories, exhibiting good representativeness and diversity.

For $N = 0.03|U|$, we compare the diversity of the selected critical instances as calculated using (2). For Iris, the diversity values of KQBC, ALEC, QUIRE, and MSAL are 0.05, 0.12, 0.30, and 0.40, respectively. For Seeds, the diversity values of KQBC, ALEC, QUIRE, and MSAL are 0.00, 0.48, 0.27, and 0.57, respectively. The MSAL algorithm guarantees the diversity of selected critical instances. Thus, the proposed method can effectively reduce the number of queries.

C. COMPARISON WITH SUPERVISED CLASSIFICATION ALGORITHMS

We compared the MSAL algorithm with nine well-known supervised classification algorithms: kNN [42], J48 [43], ClassificationViaClustering (CVC) [44], Random Forest (RF) [45], AdaBoostM1 (ABM) [46], Classification Via Regression (CVR) [47], Logit Boost (LB) [48], Bagging [49], and Multiclass Classifier (MCC) [50]. These algorithms are tested using Weka's built-in codes.

Table 4 compares the accuracy of MSAL and these nine supervised classification algorithms when $N = 0.1|U|$. The average ranks were obtained by applying the Friedman procedure. The Friedman test [51] is the most well-known nonparametric test when there are more than two related samples. The p value calculated by the Friedman test is 1.70E-5. The best results are highlighted in boldface. MSAL is generally superior to the existing supervised classification algorithms. Through significance analysis, the average rank of MSAL was found to be 2.5417. The proposed algorithm ranked first for six of the datasets. MSAL generally outperformed the existing supervised classification algorithms. In some cases,

[1] http://www.fansmale.com/software.html
[2] http://parnec.nuaa.edu.cn/huangsj

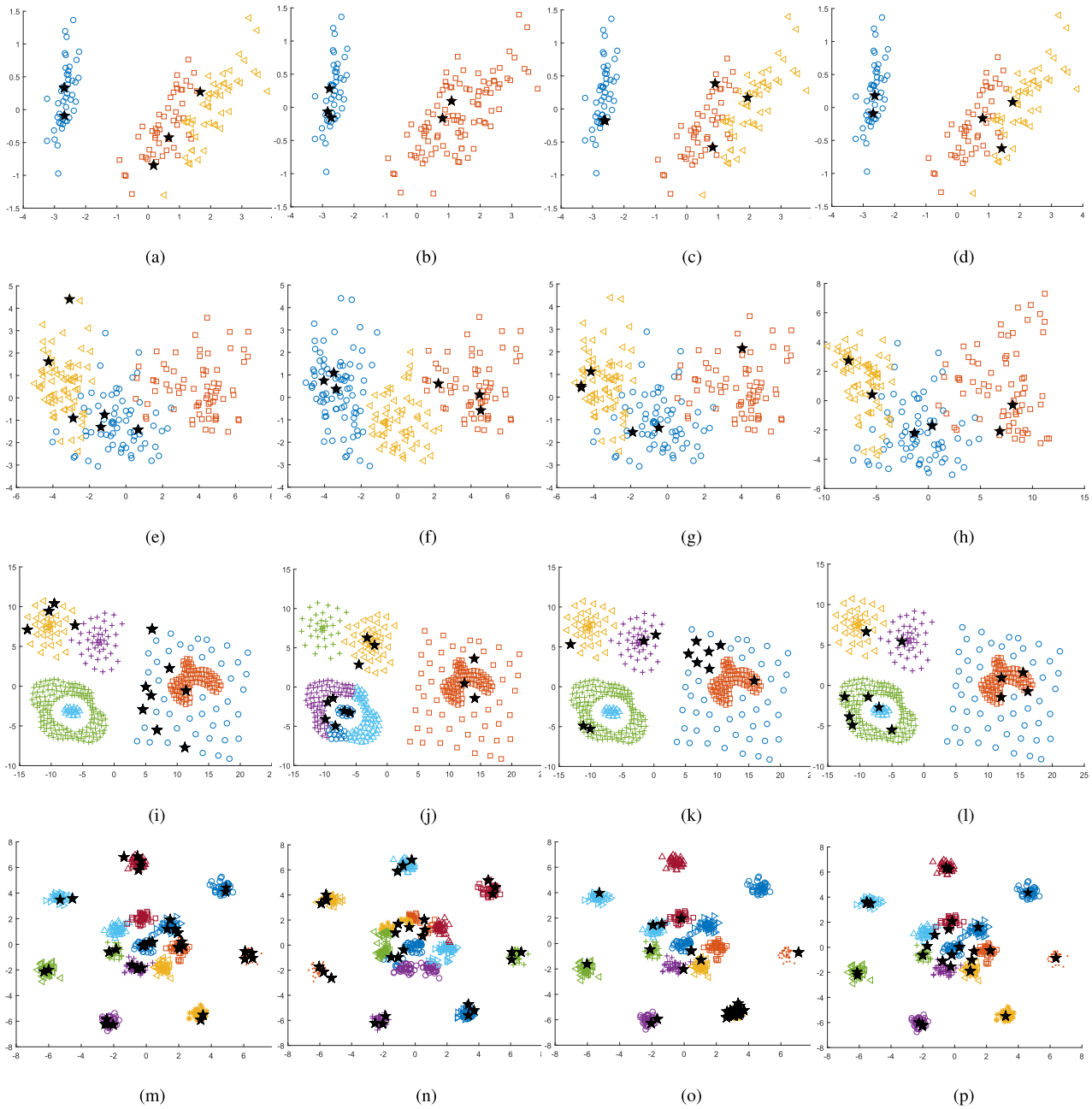

FIGURE 3. Comparison of four typical critical instance selections. Black stars represent the selected critical instances. The Iris, Seeds, Compound, and R15 datasets contain 5, 6, 12, and 30 critical instances, respectively. (a) Iris-KQBC. (b) Iris-ALEC. (c) Iris-QUIRE. (d) Iris-MSAL. (e) Seeds-KQBC. (f) Seeds-ALEC. (g) Seeds-QUIRE. (h) Seeds-MSAL. (i) Compound-KQBC. (j) Compound-ALEC. (k) Compound-QUIRE. (l) Compound-MSAL. (m) R15-KQBC. (n) R15-ALEC. (o) R15-QUIRE. (p) R15-MSAL.

the performance of MSAL is worse than that of the other algorithms. For example, the kNN algorithm is ranked first with the Seeds and Ionosphere datasets, and the J48 algorithm is ranked first with the Haberman dataset.

According to the Friedman statistics, the assumption that "all algorithms have the same performance" can be rejected. Statistical results show that the performance of these algorithms is significantly different. We used a post-hoc Nemenyi test to further compare the algorithms at a significance level of $\alpha = 0.05$. Table 5 presents the results. The MSAL algorithm is significantly better than the CVC, ABM, J48, CVR, MCC, and LB algorithms. There is no significant difference between MSAL and the Bagging, RF, and kNN algorithms.

D. COMPARISON WITH ACTIVE LEARNING ALGORITHMS

In this section, we compare the MSAL algorithm with three state-of-the-art active learning algorithms, namely

TABLE 4. Accuracy of MSAL and nine supervised classification algorithms. The average ranks were obtained using Friedman's test.

	kNN	J48	CVC	RF	ABM	CVR	LB	Bagging	MCC	MSAL
Appendicitis	0.7368	0.6526	0.6000	0.6525	0.6632	0.8105	0.7158	0.8000	0.4947	**0.8737**
Iris	0.9037	0.9185	0.5926	0.8889	0.9259	0.7037	0.9259	0.7851	0.8593	**0.9630**
Sonar	0.6631	0.5133	0.4010	0.6364	**0.6952**	0.5828	0.6631	0.6791	**0.6952**	0.6791
Seeds	**0.9153**	0.5026	0.5873	0.8359	0.7989	0.8201	0.8253	0.6667	0.8412	0.8942
Haberman	0.7018	**0.7236**	0.4836	0.7018	0.7200	**0.7236**	**0.7236**	**0.7236**	0.7163	**0.7236**
Ionosphere	**0.8449**	0.8291	0.6170	0.8386	0.7848	0.7468	0.8006	0.7974	0.8259	0.8222
Compound	0.8162	0.7771	0.6100	0.8133	0.6100	0.8050	0.8133	0.8133	0.8105	**0.9276**
R15	0.9814	0.7389	0.1333	0.8833	0.1203	0.7129	0.7500	0.7389	0.6351	**0.9944**
D31	0.9559	0.9121	0.0620	0.9366	0.0616	0.8939	0.8272	0.9477	0.6433	**0.9581**
Banana	0.8641	0.8486	0.5886	0.8708	0.7061	0.8685	0.7526	**0.8730**	0.5467	0.8132
Texture	0.9515	0.8347	0.1759	0.8954	0.1759	0.8909	0.8881	0.8515	**0.9793**	0.8988
Twonorm	0.9249	0.8087	**0.9761**	0.9191	0.8613	0.8743	0.8643	0.8973	**0.9761**	0.9332
Average rank	3.4167	6.4583	8.7917	4.7917	7.2500	6.0000	5.2500	4.9167	5.5833	**2.5417**

TABLE 5. Adjusted *p*-values computed by the post-hoc tests.

MSAL v.s.	$z = (R_0 - R_i)/SE$	p
CVC	5.056499	0
ABM	3.809229	0.000139
J48	3.168739	0.001531
CVR	2.797929	0.005143
MCC	2.460829	0.013862
LB	2.191150	0.028441
Bagging	1.921470	0.054673
RF	1.820340	0.068707
kNN	0.707910	0.479001

KQBC [17], ALEC [11],[3] and QUIRE [13].[4] KQBC selects the most informative instances, ALEC selects the most representative instances, and QUIRE selects informative and representative instances.

Fig. 4 illustrates the accuracy of MSAL and the three active learning algorithms. The number of labels N provided by the oracle ranges from $0.03|U|$ to $0.12|U|$. The MSAL algorithm is significantly better than the other three active learning algorithms in the following two aspects. First, the MSAL algorithm is more accurate than the others. For the Appendicitis, Haberman, Ionosphere, Compound, R15, D31, and Texture datasets, the classification accuracy of MSAL is significantly higher than that of the other algorithms for the entire range of N. For Iris and R15, the classification accuracy of MSAL reaches 0.9448 and 0.9038 with just 3% labeled instances. The MSAL algorithm also achieves the highest accuracy faster than the KQBC, ALEC, and QUIRE algorithms. In some cases, the performance of MSAL is worse than that of the other algorithms. For example, with the Seeds dataset, the ALEC algorithm is the most accurate when $N > 0.05|U|$, and with the Twonorm dataset, ALEC is more accurate than MSAL.

Second, the MSAL algorithm is more stable than the other three active learning algorithms. For 10 datasets, the MSAL accuracy increases steadily as the value of N increases. The QUIRE algorithm exhibits significant fluctuations with six datasets, including Appendicitis, R15, and D31. The KQBC algorithm also fluctuates on the Appendicitis, Iris, Sonar, Seeds, and Haberman datasets.

[3]http://www.fansmale.com/software.html

[4]http://parnec.nuaa.edu.cn/huangsj

TABLE 6. Accuracy of MSAL and three active learning algorithms. The average ranks were obtained using Friedman's test.

	KQBC	ALEC	QUIRE	MSAL
Appendicitis	0.6800	0.7579	0.5484	**0.8737**
Iris	0.8993	**0.9645**	0.9556	0.9630
Sonar	0.4754	0.6684	**0.7219**	0.6791
Seeds	0.6916	**0.9101**	**0.9101**	0.8942
Haberman	0.7007	0.7018	0.6971	**0.7236**
Ionosphere	0.7810	0.6305	0.4749	**0.8222**
Compound	0.3735	0.8886	0.6006	**0.9276**
R15	0.6946	0.9852	0.7815	**0.9944**
D31	0.1008	0.8998	0.5305*	**0.9581**
Banana	**0.8776**	0.6864	0.5073*	0.8132
Texture	0.8072	**0.8993**	0.5717*	0.8988
Twonorm	**0.9628**	0.9379	0.5150*	0.9332
Average rank	3.0833	2.0417	3.2083	1.6667

$*$ indicates 10% sampling of the original dataset.

Table 6 presents the accuracy of the MSAL algorithm and the three active learning algorithms when $N = 0.1|U|$. For QUIRE, memory overflows occur when testing large datasets. Therefore, for the D31, Banana, Texture, and Twonorm datasets, we sampled 10% of the instances for each class and formed new datasets. The average ranks were obtained by applying Friedman's procedure. The p value calculated by the Friedman test is 0.005651. The best results are highlighted in boldface. According to our significance analysis, MSAL generally outperforms the other algorithms, with an average rank of 1.6667. Of the 12 datasets, the MSAL algorithm achieves the highest accuracy on six. The mean classification accuracy of MSAL is 0.8734. In some cases, the performance of MSAL is worse than that of other algorithms. For example, the ALEC algorithm is ranked first with the Iris, Seeds, and Texture datasets.

According to the Friedman statistics, the assumption that "all algorithms have the same performance" can be rejected. The statistical results show that the performance of these algorithms is significantly different. We used a post-hoc Nemenyi test to further compare the algorithms at a significance level of $\alpha = 0.05$. Table 7 presents the results. The MSAL algorithm is significantly better than the

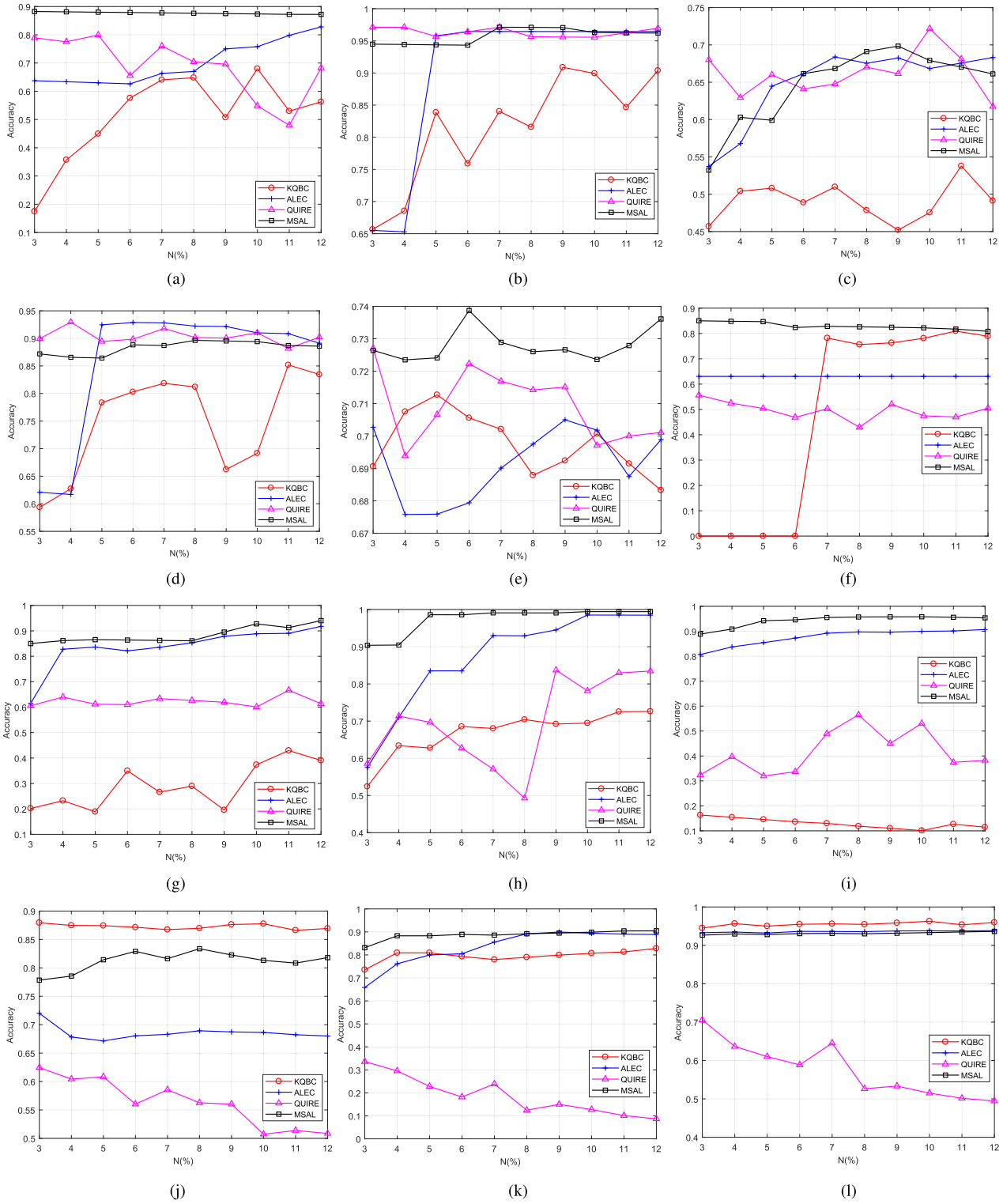

FIGURE 4. Comparison with three active learning algorithms. N is the number of labels provided by the oracle; N ranges from $0.03|U|$ to $0.12|U|$. (a) Appendicitis. (b) Iris. (c) Sonar. (d) Seeds. (e) Haberman. (f) Ionosphere. (g) Compound. (h) R15. (i) D31. (j) Banana. (k) Texture. (l) Twonorm.

QUIRE and KQBC algorithms. The second-ranked ALEC algorithm is also significantly better than the QUIRE and KQBC algorithms. There is no significant difference between the MSAL and ALEC algorithms. In addition, there is no significant difference between the QUIRE and KQBC algorithms.

We used the Twonorm dataset to quantify the efficiency of MSAL. Fig. 5 shows the relationship between the training

TABLE 7. Adjusted *p*-values computed by the post-hoc tests.

Algorithm	$z = (R_0 - R_i)/SE$	p
MSAL vs. QUIRE	2.925107	0.003443
MSAL vs. KQBC	2.687936	0.007190
ALEC vs. QUIRE	2.213594	0.026857
ALEC vs. KQBC	1.976424	0.048107
MSAL vs. ALEC	0.711512	0.476767
KQBC vs. QUIRE	0.237171	0.812524

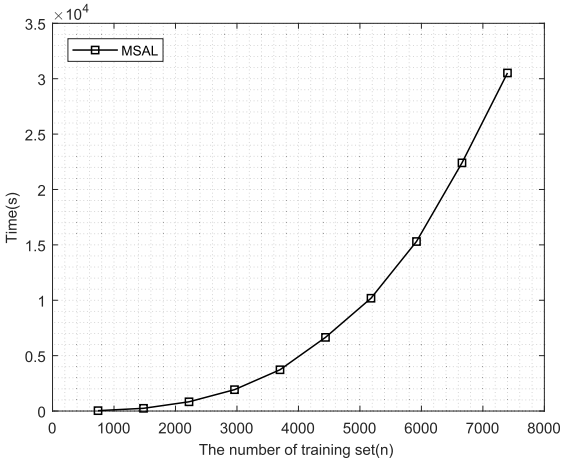

FIGURE 5. Runtime as a function of the training set size *n*.

set size n and the runtime. We observe that the runtime is proportional to n^2. Therefore, the MSAL algorithm is efficient and scalable.

E. DISCUSSION

We are now able to answer the questions proposed at the beginning of this paper.

1) MSAL is more accurate than several popular supervised learning algorithms. It is effective for most datasets that have different distributions, shapes, and cluster classes. The validity and advantages of active learning are confirmed.
2) MSAL is more accurate than several state-of-the-art active learning algorithms. It achieves high accuracy using only a small number of labeled instances.
3) MSAL is efficient and scalable.

Note that the values of d_c and β are essential to the performance of our algorithm. For the same dataset, different values of d_c and β will lead to different classification accuracies. We first set d_c and β according to the Rodriguez's [14] recommendation. We then considered the distribution of the actual data to adjust the values of d_c and β.

VI. CONCLUSION AND FURTHER WORK

In this paper, we have considered active learning from a constrained optimization perspective to address the issue of which instances are critical. First, we used the soft-max model to calculate the information entropy and obtain information for each instance. Second, considering the arbitrary distribution, we used a non-parametric estimation method to obtain a probability density function. Third, we designed diversity constraints to avoid selecting instances that are too similar. Finally, we designed a multi-standard optimization active learning (MSAL) algorithm. The results of significance tests verify the superiority of MSAL over several state-of-the-art active learning algorithms.

The following research topics deserve further investigation:

1) Simplifying the parameter settings. The accuracy of MSAL depends on the cutoff distance d_c and the difference threshold β, which are difficult to determine. One solution is to establish certain rules for different situations. A better solution is to avoid the parameter setting altogether without sacrificing the predication accuracy.
2) Reducing the time complexity. The runtime of the MSAL algorithm becomes rather long as the dataset size increases. The bottleneck in the algorithm is the computation of the distance between instance pairs. The time complexity may be reduced by using divide-and-conquer approaches such as that described in [52].
3) Revising MSAL for cost-sensitive active learning. Currently, MSAL can be applied to problems with a fixed number of labels provided by the oracle. As active learning is cost-sensitive, it would be natural to modify MSAL for problems in which the labeling and misclassification costs are known.

REFERENCES

[1] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 705–712, 1996.
[2] B. Settles, "Active learning literature survey," *Univ. Wisconsinmadison*, vol. 39, no. 2, pp. 127–131, 2009.
[3] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, no. 1, pp. 999–1006, 2002.
[4] D. Yu, B. Varadarajan, L. Deng, and A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," *Comput. Speech, Lang.*, vol. 24, no. 3, pp. 433–444, 2010.
[5] S.-L. Sun and D. R. Hardoon, "Active learning with extremely sparse labeled examples," *Neurocomputing*, vol. 73, nos. 16–18, pp. 2980–2988, 2010.
[6] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. ACM MM*, 2001, pp. 107–118.
[7] V. Fedorov, "Optimal experimental design," *Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 5, pp. 581–589, Sep. 2010.
[8] X. Zhao, Y. Kai, V. Tresp, X.-W. Xu, and J.-Z. Wang, "Representative sampling for text classification using support vector machines," in *Proc. ECIR*, 2003, pp. 393–407.
[9] F. Min, F.-L. Liu, L.-Y. Wen, and Z.-H. Zhang, "Tri-partition cost-sensitive active learning through kNN," *Soft Comput.*, vol. 23, no. 5, pp. 1557–1572, 2019.
[10] Y.-X. Wu, X.-Y. Min, F. Min, and M. Wang, "Cost-sensitive active learning with a label uniform distribution model," *Int. J. Approx. Reasoning*, vol. 105, pp. 49–65, Feb. 2019.
[11] M. Wang, F. Min, Y.-X. Wu, and Z.-H. Zhang, "Active learning through density clustering," *Expert Syst. Appl.*, vol. 85, pp. 305–317, Nov. 2017.
[12] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *Proc. ECML*, 2007, pp. 116–127.
[13] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," in *Proc. NIPS*, 2010, pp. 892–900.

[14] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[15] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. 5th Workshop Comput. Learn. Theory*, vol. 284, 1992, pp. 287–294.
[16] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *Proc. ICML*, 2008, pp. 208–215.
[17] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Kernel query by committee (KQBC)," Leibniz Center, Hebrew Univ., Jerusalem, Israel, Tech. Rep. 2003-88, 2004.
[18] Y. Kai, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proc. ICML*, 2006, pp. 1081–1088.
[19] S. Zhao, X. Sun, J. Chen, Z. Duan, Y.-P. Zhang, and Y.-W. Zhang, "Relational granulation method based on quotient space theory for maximum flow problem," *Inf. Sci.*, 2018.
[20] X.-Z. Wang, R. Wang, and C. Xu, "Discovering the relationship between generalization and uncertainty by incorporating complexity of classification," *IEEE Trans. Syst., Man, Cybern.*, vol. 48, no. 2, pp. 703–715, Feb. 2018.
[21] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proc. Mach. Learn.*, 1995, pp. 150–157.
[22] A. Mccallum and K. Nigam, "Employing EM and pool-based active learning for text classification," in *Proc. ICML*, 1998, pp. 350–358.
[23] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proc. ICML*, 2004, p. 79.
[24] D. Tuia, M. Kanevski, J. M. Mari, and G. Campsvalls, "Cluster-based active learning for compact image classification," in *Proc. Geosci., Remote Sens. Symp.*, 2010, pp. 2824–2827.
[25] Y. Y. Yao, "Interval sets and three-way concept analysis in incomplete contexts," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 3–20, 2017.
[26] J. Chen, Y.-P. Zhang, and S. Zhao, "Multi-granular mining for boundary regions in three-way decision theory," *Knowl. Based Syst.*, vol. 91, pp. 287–292, Jan. 2016.
[27] R. Wang, X.-Z. Wang, S. Kwong, and C. Xu, "Incorporating diversity and informativeness in multiple-instance active learning," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1460–1475, Dec. 2017.
[28] S. C. H. Hoi, J. Rong, J. Zhu, and M. R. Lyu, "Semi-supervised SVM batch mode active learning for image retrieval," in *Proc. CVPR*, 2008, pp. 1–7.
[29] J. A. Núñez, P. M. Cincotta, and F. C. Wachlin, "Information entropy," *Celestial Mech., Dyn. Astron.*, vol. 64, nos. 1–2, pp. 43–53, 1996.
[30] Y. C. Liu, "Gradient descent method," *J. Nanjing Univ. Sci., Technol.*, vol. 68, no. 2, pp. 13–22, 1993.
[31] J. E. Dennis and J. J. Moré, "Quasi-Newton methods, motivation and theory," *SIAM Rev.*, vol. 19, no. 1, pp. 46–89, 1977.
[32] L. Verde *et al.*, "First-year *Wilkinson Microwave Anisotropy Probe* (WMAP) observations: Parameter estimation methodology," *Astrophys. J. Suppl.*, vol. 148, no. 1, pp. 195–211, 2003.
[33] R. D. Harcourt, "Angular eigenvalues and some classical probability density functions for the helium isoelectronic sequence," *Phys. Lett. A*, vol. 200, no. 2, pp. 144–148, 1995.
[34] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962.
[35] R. Memisevic, C. Zach, G. Hinton, and M. Pollefeys, "Gated softmax classification," in *Proc. NIPS*, 2010, pp. 1603–1611.
[36] L. C. Baird and A. W. Moore, "Gradient descent for general reinforcement learning," in *Proc. NIPS*, vol. 11, 1999, pp. 968–974.
[37] G. Heindl, V. Kreinovich, and A. V. Lakeyev, "Solving linear interval systems is NP-Hard even if we exclude overflow and underflow," *Reliable Comput.*, vol. 4, no. 4, pp. 383–388, 1998.
[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[39] C. Blake and C. J. Merz. (1998). *UCI Repository of Machine Learning Databases*. [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html
[40] J. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.
[41] H. Chiroma, A. Y. Gital, A. Abubakar, and A. Zeki, "Comparing performances of Markov blanket and tree augmented Naïve-Bayes on the IRIS dataset," *Lect. Notes Eng. Comput. Sci.*, vol. 2209, no. 1, pp. 328–331, 2014.
[42] J.-W. Han and M. Kamber, *Data Mining Concept and Techniques*. Amsterdam, The Netherlands: Elsevier, 2006.
[43] Z.-Y. Xiang and L. Zhang, "Research on an optimized C4.5 algorithm based on rough set theory," in *Proc. ICMeCG*, 2012, pp. 272–274.
[44] M. I. Lopez, J. M. Luna, C. Romero, and S. Ventura, "Classification via clustering for predicting final marks based on student participation in forums," in *Proc. Int. Educ. Data Mining Soc.*, 2012, pp. 148–151.
[45] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 23, no. 23, pp. 18–22, 2002.
[46] E. A. Cortés, M. G. Martínez, and N. G. Rubio, "Multiclass corporate failure prediction by Adaboost.M1," *Int. Adv. Econ. Res.*, vol. 13, no. 3, pp. 301–312, 2007.
[47] Y.-X. Ruan, H.-T. Lin, and M.-F. Tsai, "Improving ranking performance with cost-sensitive ordinal classification via regression," *Inf. Retr. J.*, vol. 17, no. 1, pp. 1–20, 2014.
[48] Y.-D. Cai, "Using logitboost classifier to predict protein structural classes," *J. Theor. Biol.*, vol. 238, no. 1, pp. 172–176, 2006.
[49] J. R. Quinlan, "Bagging, boosting, and C4.5," in *Proc. AAAI*, 1996, pp. 725–730.
[50] S. Afshar, M. Mosleh, and M. Kheyrandish, "Presenting a new multiclass classifier based on learning automata," *Neurocomputing*, vol. 104, pp. 97–104, Mar. 2013.
[51] O. Reyes, A. H. Altalhi, and S. Ventura, "Statistical comparisons of active learning strategies over multiple datasets," *Knowl. Based Syst.*, vol. 145, pp. 274–288, Apr. 2018.
[52] J. J. M. Cuppen, "A divide and conquer method for the symmetric tridiagonal eigenproblem," *Numerische Mathematik*, vol. 36, no. 2, pp. 177–195, 1980.

MIN WANG is currently an Associate Professor with Southwest Petroleum University. She presided over a number of longitudinal tasks including the Sichuan Science and Technology Foundation. She has authored over 10 refereed papers in various journals and conferences, including the Expert Systems with applications and the IEEE Access. She has obtained several patent and software copyright. Her research interests include data mining and active learning.

YING-YI ZHANG is currently pursuing the master's degree with Southwest Petroleum University, Chengdu, China.

FAN MIN (M'09) received the M.S. and Ph.D. degrees from the School of Computer Science and Engineering, University of Electronics Science and Technology of China, Chengdu, China, in 2000 and 2003, respectively.

He visited the University of Vermont, Burlington, Vermont, from 2008 to 2009. He is currently a Professor with Southwest Petroleum University, Chengdu. He has published more than 110 refereed papers in various journals and conferences, including the *Information Sciences*, the *International Journal of Approximate Reasoning*, and *Knowledge-Based Systems*. His current research interests include data mining, recommender systems, active learning, and granular computing.