International Journal of Innovative Computing, Information and Control
Volume **18**, Number **2**, April **2022**

 ISSN 1349-4198
pp. 377–393

ACTIVE LEARNING OF THREE-WAY DECISION BASED ON NEIGHBORHOOD ENTROPY

Qiuyue Lv[1,2] and Minggang Dong[1,2,*]

[1]College of Information Science and Engineering
[2]Guangxi Key Laboratory of Embedded Technology and Intelligent System
Guilin University of Technology
No. 319, Yanshan Street, Guilin 541000, P. R. China
1020190579@glut.edu.cn; *Corresponding author: 2000025@glut.edu.cn

Received September 2021; revised January 2022

Abstract. *Addressing the issue that the entropy-based query strategy neglects the distribution of samples, a selection strategy based on neighborhood entropy is designed to combine sample distribution characteristics with entropy. The weighted entropy of all unlabeled samples in the neighborhood is employed as the measure, which can select both informative and representative unlabeled samples to label, so a more effective classifier can be trained by these samples. Furthermore, considering samples from different regions have different values, the neighborhood entropy is adopted as the decision function of the three-way decision, and an active learning of three-way decision based on neighborhood entropy is developed, namely ALTD_NE. The unlabeled dataset is divided into three regions by the value of the neighborhood entropy, and then the most valuable samples in different parts are selected for labeling and employed to train the classifier, which can further improve the classification performance. Experiments are conducted on several standard classification datasets of the KEEL and UCI databases and one real classification dataset. ALTD_NE has a better performance in most cases, which outperforms several comparative algorithms in ACC, F1 Value, and AUC.*

Keywords: Active learning, Unlabeled sample, Neighborhood entropy, Three-way decision

1. **Introduction.** In many classification tasks, the most valuable unlabeled samples need to be selected and manually annotated for achieving satisfactory classification accuracy [1]. However, this procedure requires a great quantity of professional domain knowledge and consumes a lot of labeling costs and material resources [2,31]. Active learning aims to reduce the number of labeled samples to train a high accurate classifier [3], where an appropriate query strategy can be applied to iteratively choosing the most valuable unlabeled sample, and a classification model trained with these selected samples can perform well on unseen data samples. The most desirable sample can be chosen by active learning, avoiding the selection of redundant data, and reducing the manpower and material resources required to label a large number of unlabeled samples [4].

Two types of sampling criteria are applied in active learning, i.e., informativeness and representativeness. Informativeness measures the ability of a sample to reduce the classification error of the classification model, whereas representativeness measures the ability of a sample to represent the entire data space [5,6]. Most popular active learning algorithms deploy the informativeness criterion to query the most informative samples, which can minimize the uncertainty of the classifier. These methods only focus on the information of a scarce number of samples but the data distribution of abundant unlabeled samples

DOI: 10.24507/ijicic.18.02.377

is neglected, leading to a sampling bias. Hence, several active learning algorithms select the representative samples by exploiting the structure of unlabeled samples, which can overcome the deficiency of the informativeness criterion. These algorithms prefer a selective sampling performed in areas close to cluster centers or in high-density areas, but the optimal cluster structure maybe not be found and the informativeness of the query samples is almost ignored [7,8].

Usually, active learning methods adopting a single criterion limit the performance of the classifier. Consequently, some algorithms [9-11,32] have been proposed to address the issue by combining two criteria to choose the valuable sample for annotating. These methods may be susceptible to a suboptimal performance as they are difficult to find a desirable trade-off and two criteria almost are inconsistent in sample selection [1,12,13]. Further, these algorithms always ignore the fact that the unlabeled samples in different parts have different worth for improving classification performance [2]. The informative samples located in high-density areas possess high labeling worth, which can make a great contribution to the improvement of classification performance. While for low-informative samples located in sparse areas, their labels can be easily determined by the classifier, which is not improved much for classification performance.

Considering that sample's representativeness is neglected when selecting samples for the entropy strategy [14], active learning based on neighborhood entropy (AL_NE) is proposed, which adopts the weighted entropy of all unlabeled samples in the neighborhood as a measurement for a selection of both informative and representative samples. Besides, taking into account that samples located in different regions possess different worth, we develop a novel active learning of three-way decision based on neighborhood entropy (ALTD_NE). More specifically, the neighborhood entropy is regarded as the decision function to divide the unlabeled samples into three regions (positive, negative, and boundary), then the samples in diverse regions can be manipulated separately to choose the most valuable samples, and the performance of the classifier can be further improved effectively. Results on benchmark datasets show a consistent and sizeable improvement of the classification accuracy with ALTD_NE and AL_NE, thus verifying the effectiveness of these methods.

The rest of this paper is organized as follows. Section 2 presents a brief review of some related work. Some related technologies are discussed in Section 3. Section 4 presents the proposed algorithm in detail. The experiment results with deep empirical analysis can be obtained in Section 5. Section 6 concludes the proposed approach and provides suggestions for potential future works.

2. **Related Work.**

Some active learning methods take advantage of the informativeness criterion to query unlabeled samples. Common methods include query-by-committee [15,16], uncertainty sampling [17-19], and expected error reduction based sampling [20]. Query-by-committee chooses the samples whose label information is most inconsistent based on several models. Uncertainty sampling aims at the reduction of classification uncertainty. Expected error reduction based sampling focuses on the samples that the classifier probably makes errors. These methods only select the most informative samples for labeling, where the distribution information of the unlabeled dataset is neglected and the bias of classification is prone to increase [1,33].

Some active learning algorithms deploy the representativeness criterion to query the most desirable sample. There are several algorithms [21-23] that use the clustering method to measure the representativeness of the unlabeled samples, whose efficiency depends on the quality of the clustering results. The others are optimal experimental design methods [24], which select the most representative sample according to the discovery of optimal decision boundary previously. The drawback of these methods is that the decision boundary can be found after the large amounts of samples need to be accessed [34,35].

Xu et al. [25] proposed an active learning method that combines the clustering structure of the samples with the classification boundary, which only clusters the unlabeled samples at the classification boundary and ignores the unlabeled samples outside the boundary. Donmez et al. [26] selected the most valuable sample by dynamically balancing the uncertainty of classifier and sample density. When the sample density is dense, this method favors the selection of the most informative unlabeled sample, while the sample density is sparse only the unlabeled samples closest to the boundary need to be selected. Hu et al. [2] introduced an active learning method based on the three-way decision model, where the margin strategy [19] is regarded as the decision function to divide the entire unlabeled dataset into three domains that need to be disposed of separately. This method can address the issue that samples in different regions have different worth, but it is prone to information redundancy, as its decision function ignores the sample distribution for a sample selection. At the same time, samples in different regions should be further processed since it also introduces a part of samples that make a slight contribution to classification performance.

Our work addresses these issues by employing neighborhood entropy, which queries both informative and representative unlabeled samples, as the uncertainty of the sample and its distribution are considered simultaneously. Further, the neighborhood entropy is applied as a decision function to processing the samples into various regions, and a more effective classifier can be trained since the selection of the most valuable sample in different regions is performed.

3. **Related Technology.** Hu et al. [28] proposed a formula for calculating the neighborhood, which is defined as follows:

$$\delta_B(x_i) = \{x_j | x_j \in U, dis(x_i, x_j) \leq \delta_i\} \tag{1}$$

U is the unlabeled dataset, B is the attribute set, δ_i is the neighborhood radius of sample x_i, and $dis(x_i, x_j)$ is the Euclidean distance between x_i and x_j. The Euclidean distance is defined as

$$dis(x_i, x_j) = \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2} \tag{2}$$

where x_{ik} is the value of the sample x_i on the kth attribute.

The three-way decision concepts are proposed by [27] according to the probabilistic rough set model and decision rough set theory. Given the decision function $f(x)$, the value of the decision function is called the decision state value, which can evaluate the quality of the entity. Then, a pair of thresholds (α, β) are set for the decision state values, where $\alpha \geq \beta$ and all entities can be mapped to three regions: positive domain, negative domain, and boundary domain by comparing their decision state values with thresholds. The detailed definition of three-way decisions [30] is as follows:

Given $\forall x \in U$, decision function $f(x)$, and the thresholds (α, β) where $\alpha \geq \beta$ then:

1) If $f(x) \geq \alpha$, then $x \in POS(X)$;
2) If $\beta < f(x) < \alpha$, then $x \in BND(X)$;
3) If $f(x) \leq \beta$, then $x \in REG(X)$.

The three-way decision theory allows active learners to select samples that have a significant contribution to the classification performance; however, its performance is determined by the decision function. The margin strategy is adopted as the decision function for the TWD_Active algorithm, which performs suboptimally in some cases, as the margin strategy only focuses on samples at decision boundary but the exploration of underlying data distribution always is ignored.

In this paper, the graph density strategy in [29] is applied to measuring the representativeness of a sample. This strategy adopts the Gaussian kernel function to establish a k-nearest neighbor graph centered on a sample. The specific formula is as follows:

$$W_{ij} = P_{ij} \exp\left(\frac{-dis(x_i, x_j)}{2\sigma^2}\right) \tag{3}$$

If $dis(x_i, x_j)$ is one of the k minimum distances of x_i, then $p_{ij} = 1$.

To reduce the influence of neighbor samples on its representativeness, the number of neighbors is exploited to normalize it:

$$Gra(x_i) = \frac{\sum_i W_{ij}}{\sum_i P_{ij}} \tag{4}$$

where the W_{ij} matrix is the weight of Gaussian kernel function between x_i and x_j, $Gra(x_i)$ is adopted to measure the representativeness of the sample x_i.

Further, to avoid selecting the same dense area multiple times, the formula in [29] is applied to reducing the representativeness for the direct neighbor x_j of the currently selected sample x_i:

$$Gra(x_j) = Gra(x_j) - Gra(x_i)P_{ij} \tag{5}$$

The graph density strategy can explore the high-density regions in the data distribution to select high representative samples, but the selected data may not be so effective in reducing the model uncertainty, since the informativeness of the samples is almost neglected.

4. Methodology.

Given $n = l + u$ d-dimensional samples, the unlabeled dataset $U = \{x_{l+1}, \ldots, x_n\}$, the labeled dataset $L = \{(x_1, y_1), \ldots, (x_l, y_l)\}$, and the labels $y \in \{1, \ldots, c\}$.

Considering the effectiveness of three-way decision theory is solely determined by the decision function, it allows for a decision function that can simultaneously measure both information and representativeness of the sample. Therefore, a query strategy of neighborhood entropy is developed, which takes account of the information entropy of a single data and data distribution characteristics simultaneously. Meanwhile, since the data located in different parts have different performance improvement worth for the classifier, it attaches great importance to further processing of data in different regions. The graph density strategy only considers data distribution and ignores data information when selecting data, which can be used to select the most representative samples in the informative sample space processed by decision functions.

4.1. Active learning based on neighborhood entropy.

In the machine learning community, the entropy value based on the posterior probability of sample is applied to measuring the degree of uncertainty and informativeness of the sample. The larger the entropy value of the sample, the greater its informativeness. The entropy strategy in [20] selects the most informative sample for labeling according to its entropy value, where the information of the surrounding samples is frequently ignored as this strategy only focuses on the informativeness of a single sample. Therefore, the entropy strategy usually selects multiple samples with high uncertainty that are located in the same high-density area, resulting in the redundancy of sample information.

An active learning method based on neighborhood entropy is proposed according to the entropy strategy, which considers the entropy value of a certain sample and the relationship of all samples in its neighborhood simultaneously. Hence, both informative and representative samples can be selected to query its label; instead of just focusing on its uncertainty and informativeness, a more effective classifier can be trained. First of all, to ensure the neighborhood never exceeds the boundary range, but also to find the optimal neighborhood, a parameter is used to dynamically adjust the radius of the field. Given $\forall x_i \in U$, the neighborhood radius of the sample x_i is as follows:

$$\delta_i = \lambda \min(dis(x_i, x_j)) + (1 - \lambda) \max(dis(x_i, x_j)) \tag{6}$$

where $\min_{x_j \in S}(dis(x_i, x_j))$ is the nearest distance between the sample x_i and the sample in the data space S, $\max_{x_j \in S}(dis(x_i, x_j))$ represents the farthest distance, and $S = \{x | x \in U, x \neq x_j\}$, and λ controls the size of the neighborhood radius and $0 \leq \lambda \leq 1$. The larger λ reduces the influence of neighborhood samples, which almost neglects the consideration of sample distribution characteristics. The smaller λ induces more consideration of data space distribution, while the information of a single sample is ignored, making it prone to an ineffective performance improvement. The detailed discussion about λ is shown in Section 5.3.

Secondly, to distinguish some neighbors with small entropy values, the weight of unlabeled samples in the neighborhood is adopted to normalize the neighbor samples. Given $\forall x_i \in U$, the neighborhood entropy of x_i in $\delta(x_i)$ can be defined as follows:

$$Entropy(x_i) = \frac{1}{|\delta(x_i)|} \sum_{x_j \in \delta(x_i)} entropy(x_j) \tag{7}$$

where $|\delta(x_i)|$ is the number of unlabeled samples in the neighborhood $\delta(x_i)$, $entropy(x_j)$ is the entropy values of sample x_j, and the entropy calculation formula is as follows:

$$entropy(x_i) = -\sum_{j=1}^{c} P(y_{ij}|x_i) \log P(y_{ij}|x_i) \tag{8}$$

where the y_{ij} means that sample x_i belongs to the class j, the $P(y_{ij}|x_i)$ is the posterior probability of attributing to class j and $\sum P(y_{ij}|x_i) = 1$.

Meanwhile, to avoid multiple selections of samples in the same dense area, we use the following formula to reduce the neighborhood entropy of the neighbors:

$$Entropy(x_j) = \frac{|\delta(x_j)|}{|\delta(x_j)| - 1} \left(Entropy(x_j) - \frac{entropy(x_i)}{|\delta(x_j)|} \right) \tag{9}$$

where $x_j = \{x | x \in \delta(x_i), x \neq x_i\}$.

The process of active learning based on neighborhood entropy is as follows: we first calculate the neighborhood entropy of all unlabeled samples through Equation (7) and sort them from large to small. Secondly, the sample with the largest neighborhood entropy value is selected, and then we reduce the neighborhood entropy for the neighbors of it according to Equation (9). Finally, the selected sample is marked by an oracle and used to train the classifier, which is then repeated multiple times until a termination criterion is satisfied, e.g., the annotation budget is exhausted.

4.2. **Active learning of three-way decision based on neighborhood entropy.**

Neighborhood entropy can be applied to measuring the informativeness and representativeness of an unlabeled sample. The greater the neighborhood entropy of the sample, the greater contribution to classifier performance improvement. Active learning of three-way decision based on neighborhood entropy is presented according to three-way decision theory, which adopts neighborhood entropy as the decision function. Then the unlabeled samples are divided into three parts according to the decision function value and processed separately. Finally, the most valuable samples for the classifier can be picked, since different solutions are selected for samples of different worth, so the classification performance will be further improved.

4.2.1. *The neighborhood entropy calculation.* The neighborhood entropy is applied as the decision function, where the unlabeled samples are divided into three domains according to the neighborhood entropy value. The larger neighborhood entropy value means that the sample is more valuable for marking, increasing the overall classification effect of the classifier. The processes of calculating the neighborhood entropy are as follows. Firstly, for $\forall x \in U$, the entropy $entropy(x)$ of sample x is calculated by Equation (8). Secondly, we calculate the Euclidean distance $dis(x_i, x_j)$ between every two samples in U according to Equation (2), then Equation (6) is applied to calculating the neighborhood radius δ, and then calculate the neighborhood $\delta(x)$ according to Equation (1). Finally, the neighborhood entropy $Entropy(x)$ of each unlabeled sample is calculated according to Equation (7).

4.2.2. *Regional division.* The unlabeled samples are divided into three parts according to the neighborhood entropy, corresponding to the negative domain, the boundary domain, and the positive domain. The processes of dividing unlabeled samples are as follows. Firstly, we sort the samples in U from large to small according to the neighborhood entropy $Entropy(x)$ and obtain the sequence $rank(x)$. Secondly, given a pair of thresholds (α, β), where $\alpha \geq \beta$, the unlabeled samples x can be divided into the following areas: if $rank(x) \geq \alpha$, then $x \in POS(X)$; if $\beta < rank(x) < \alpha$, then $x \in BND(X)$; if $rank(x) \leq \beta$, then $x \in REG(X)$.

Among them $0 \leq \beta \leq selectNum$, $selectNum$ is the number of unlabeled samples to be selected for each iteration. $\beta = selectNum/2$ and $\alpha = u - selectNum + 1$ are selected based on experimental experience.

4.2.3. *Separately processing of samples in different regions.* First of all, the labeling worth of positive domain samples is inferior as their classification result can easily be determined by the classifier, so the sample selection in this domain needs to be neglected.

Secondly, samples in the negative domain should be chosen by human experts for labeling since all of them are highly informative and representative.

Finally, for the samples of the boundary domain, the graph density strategy is employed to further calculate the representativeness of them, which facilitates the classification of surrounding samples. The specific details for boundary domain processing are as follows.

1) The Euclidean distance of pairwise samples is calculated according to Equation (2), and then the weight of the Gaussian kernel function between every two samples is computed based on Equation (3) to construct the matrix.

2) Equation (4) is used for redefining the representativeness of a sample to reduce the influence of surrounding samples on its representativeness.

3) To avoid selecting samples in the same dense area, we use Equation (5) to update the representativeness of direct neighbor sample points.

To sum up, the three-way decision theory is adopted to divide the unlabeled samples into three regions according to the neighborhood entropy and processed separately, which tackles the issue that samples in different regions have different worth. Meanwhile, the highly informative and representative samples are selected for labeling during the whole process and added to train the classifier, which can improve the performance of the classifier. The specific algorithm is shown in Algorithm 1.

Algorithm 1. The ALTD_NE algorithm

Input:
Unlabeled dataset U, labeled dataset L, number of selected samples *selectNum*;
Output:
Model ϕ;

for each *episode* $\in [1, N]$ **do**
  $L \Leftarrow \emptyset$;
  for $x \in U$ **do**
    Calculate the neighborhood entropy $Entropy(x)$ of unlabeled sample x;
    Divide the unlabeled samples into $REG(X)$, $BND(X)$ and $POS(X)$ according to $Entropy(x)$;
  end for
  Initialize the selected dataset $select = \emptyset$;
  for $x \in REG(X)$ **do**
    $select = select + x$
  end for
  for $x \in BND(X)$ **do**
    Get W_{ij} matrix based on Equation (3), and then calculate the representativeness $Gra(x)$ according to Equation (4);
  end for
  Sort $BND(x)$ from large to small according to $Gra(x)$;
  The $selectNum - \beta$ most informative samples in $BND(X)$ are added to *select*;
  Adopt Equation (5) to update the representativeness for direct neighbors of the selected samples;
  if $episode = N$ **then**
    break;
  else
    for $x \in select$ **do**
      Obtain the annotation y for x;
      $L \Leftarrow L + (x, y)$;
      $U \Leftarrow U - x$;
    end for
  end if
  Update model ϕ based on L
  Update *episode*;
end for

5. Experiments.

5.1. Experiment setup.

To verify the effectiveness of AL_NE and ALTD_NE, several standard classification datasets in the KEEL database and UCI platform are selected for experiments, including two-class and multi-class datasets. The specific information of the dataset is shown in Table 1.

Seven active learning methods are selected for experimental comparison. In addition to the two methods proposed in this paper, it also includes active learning based on entropy strategy (Entropy) [14], active learning based on graph density (Graph Density) [29], active learning based on random selection (Random), the query by committee algorithm (QBC) [15] and active learning based on the three-way decision (TWD_Active) [2]. In this paper, the dataset is divided into a training dataset, unlabeled dataset, and test dataset,

TABLE 1. Dataset information

Dataset	Feature	Instance	Class
austra	14	690	2
biodeg	41	1055	2
bupa	6	345	2
credit	20	1000	2
ecoli	7	336	8
heart	13	270	2
led7digit	7	500	10
page-blocks	10	5473	5
pima	8	768	2
segment	19	2310	7
sonar	60	208	2
steel	27	1941	7
vehicle	18	846	4
wine	13	178	3
yeast	8	1484	8

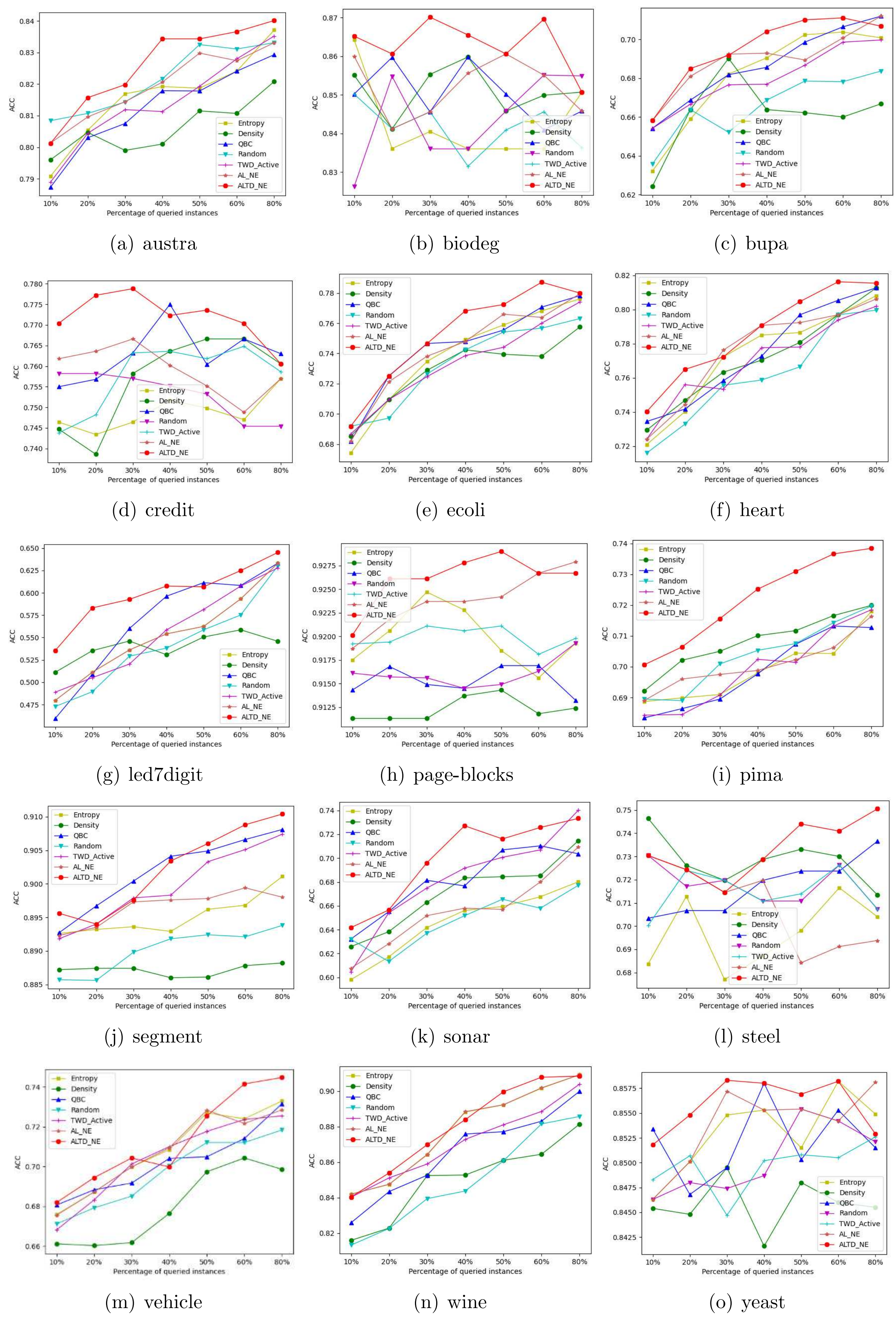
where the training dataset accounts for 10%, the unlabeled dataset accounts for 60%, and the test dataset accounts for 30%. The logistic regression classifier is chosen to calculate the posterior probability value of unlabeled samples in Equation (8). Parameter λ can adjust the neighborhood size, which allows finding an appropriate trade-off between informativeness and representativeness for data selection. The larger λ leads to the smaller neighborhood, which weakens the influence of data distribution characteristics on neighborhood entropy. Whereas the smaller λ induces the neglection of the informativeness of a single data, which results in little difference between samples. The experimental results in Section 5.3 show that $\lambda = 0.8$ will obtain the best results in this paper. To test the performance of the classifier, ACC, F1 Value, and AUC are picked as evaluation metrics. To ensure the authenticity of the experimental results, the experiment will be repeated 10 times, and the average value is taken as the final experimental result.

5.2. **Discussion and analysis.** To display the experimental results more intuitively, 10%, 20%, 30%, 40%, 50%, 60%, and 80% of the unlabeled samples are chosen as queries to compare the experiments respectively. Figure 1 shows the classification accuracy of different algorithms on each dataset, where the x-axis represents the percentage of the selected samples, with a unit of 10%. Table 2, Table 3, Table 4 and Table 5 are the experimental results of different algorithms on F1 Value and AUC respectively, where the best performance on each dataset is highlighted in boldface.

The following observation can be obtained.

1) The ACC values of AL_NE are significantly better than the Entropy on the dataset bupa, wine, and segment; AL_NE has an average increase in F1 Value on the dataset austra compared to the Entropy for 1.53%; Compared with Entropy, the average AUC of AL_NE increases 1.13% and 1.72% on the dataset pima and vehicle respectively, and is especially better than all comparison algorithms on the dataset pima.

2) ALTD_NE has significantly better ACC values on the test datasets ecoli, pima, and sonar than other algorithms in most cases; where the F1 Value increases by 6.78% and 5.28% on average for the datasets led7digit and sonar; where the AUC values increase on average 8.59%, 3.01%, and 5.68% for ecoli, heart, and vehicle compared to other methods.



(a) austra (b) biodeg (c) bupa

(d) credit (e) ecoli (f) heart

(g) led7digit (h) page-blocks (i) pima

(j) segment (k) sonar (l) steel

(m) vehicle (n) wine (o) yeast

Figure 1. Comparison on classification accuracy

TABLE 2. Comparison of F1 Value on austra, bupa, ecoli and heart, etc.

Dataset	Algorithm	10%	20%	30%	40%	50%	60%	80%	Mean
	Entropy	0.7810	0.7932	0.7963	0.7930	0.8034	0.8120	0.8219	0.8001
	Density	0.7856	0.7846	0.7840	0.7877	0.7927	0.8016	0.8066	0.7918
	QBC	**0.7919**	0.7925	0.7948	0.8003	0.8076	0.8055	0.8627	0.8079
austra	Random	0.7743	0.7791	0.7873	0.7854	0.7940	0.8022	0.8158	0.7912
	TWD_Active	0.7794	**0.8009**	0.8091	0.8114	0.8114	0.8141	0.8221	0.8069
	AL_NE	0.7838	0.7995	0.8024	0.8227	0.8302	0.8361	0.8333	0.8154
	ALTD_NE	0.7904	**0.8009**	**0.8150**	**0.8235**	**0.8344**	**0.8374**	**0.8386**	**0.8200**
	Entropy	0.5098	0.5346	0.5277	0.5283	0.5466	0.5359	0.5616	0.5349
	Density	0.5524	0.5412	0.5399	0.5533	**0.5880**	0.5874	0.5389	0.5573
	QBC	0.5405	0.5296	0.5174	0.5508	0.4980	0.5253	0.5184	0.5257
bupa	Random	**0.5531**	0.5343	0.5412	0.5461	0.5389	0.5435	0.5471	0.5435
	TWD_Active	0.5379	0.5082	0.5445	0.5531	0.5468	0.5475	0.5483	0.5409
	AL_NE	0.5300	0.5455	0.5425	0.5376	0.5547	0.5443	**0.5682**	0.5461
	ALTD_NE	0.5478	**0.5702**	**0.5646**	**0.5696**	0.5705	**0.5993**	0.5669	**0.5698**
	Entropy	0.5082	0.5238	0.5556	0.5742	0.5743	0.5796	0.5982	0.5591
	Density	0.4704	0.5142	0.5262	0.5676	0.6065	**0.6246**	0.6312	0.5630
	QBC	0.5172	**0.5364**	**0.5718**	0.5514	0.5604	0.5706	0.6186	0.5609
ecoli	Random	0.5058	0.5202	0.5274	0.5658	0.5946	0.5892	0.6198	0.5604
	TWD_Active	0.4890	0.5304	0.5460	0.5543	0.5694	0.5886	0.6354	0.5590
	AL_NE	0.5082	0.5244	0.5568	0.5754	0.5754	0.5797	0.6006	0.5601
	ALTD_NE	**0.5208**	0.5310	0.5688	**0.6024**	**0.6144**	0.6145	**0.6522**	**0.5863**
	Entropy	0.7102	0.7171	0.7389	0.7485	0.7502	0.7611	0.7777	0.7434
	Density	0.7092	0.7268	0.7337	0.7509	0.7459	0.7543	0.7734	0.7420
	QBC	0.7174	**0.7479**	0.7482	0.7492	0.7611	0.7727	0.7925	0.7556
heart	Random	0.7221	0.7383	0.7479	0.7789	0.7780	0.7962	0.8002	0.7659
	TWD_Active	**0.7224**	0.7406	0.7475	0.7396	0.7482	0.7555	0.7889	0.7490
	AL_NE	0.7103	0.7173	0.7389	0.7485	0.7502	0.7614	0.7777	0.7435
	ALTD_NE	0.7131	0.7462	**0.7535**	**0.7839**	**0.7945**	**0.8038**	**0.8088**	**0.7720**
	Entropy	0.4258	0.4444	0.4779	0.4805	0.5202	0.5413	0.5901	0.4972
	Density	0.4197	0.4682	0.4805	0.4779	0.4973	0.4788	0.4832	0.4722
	QBC	0.4373	0.4779	0.5158	0.5238	0.5794	0.5776	**0.5988**	0.5301
led7digit	Random	0.3853	0.4391	0.4558	0.4479	0.4461	0.4823	0.5300	0.4552
	TWD_Active	0.4179	0.4558	0.4673	0.5000	0.5158	0.5211	0.5405	0.4883
	AL_NE	0.4258	0.4445	0.4779	0.4805	0.5202	0.5414	0.5898	0.4972
	ALTD_NE	**0.4470**	**0.5150**	**0.5494**	**0.5926**	**0.5917**	**0.6120**	0.5970	**0.5578**
	Entropy	0.6339	0.6305	0.6338	0.6427	0.6420	0.6451	0.6637	0.6417
	Density	0.6079	0.6343	0.6298	0.6360	0.6387	0.6423	0.6540	0.6347
	QBC	0.6266	0.6394	0.6444	0.6502	0.6555	0.6519	0.6685	0.6481
pima	Random	0.6218	0.6324	0.6326	0.6350	0.6355	0.6454	0.6454	0.6354
	TWD_Active	0.6382	0.6350	0.6391	0.6442	0.6531	0.6675	0.6495	0.6467
	AL_NE	0.6300	0.6348	0.6396	0.6495	0.6492	0.6524	0.6588	0.6449
	ALTD_NE	**0.6471**	**0.6536**	**0.6593**	**0.6646**	**0.6759**	**0.6877**	**0.6966**	**0.6693**
	Entropy	0.8655	0.8799	0.8793	0.8775	0.8875	0.8857	0.8870	0.8803
	Density	0.8640	0.8644	0.8653	0.8623	0.8655	0.8686	0.8648	0.8650
	QBC	0.8922	0.9013	0.9036	0.9076	0.9053	0.9093	**0.9204**	0.9057
segment	Random	0.8801	0.8748	0.8735	0.8771	0.8687	0.8786	0.8816	0.8763
	TWD_Active	0.8621	0.8625	0.8860	0.8856	0.8818	0.8805	0.9009	0.8799
	AL_NE	0.8655	0.8801	0.8793	0.8829	0.8829	0.8852	0.8860	0.8803
	ALTD_NE	**0.9051**	**0.9225**	**0.9259**	**0.9223**	**0.9223**	**0.9223**	0.9142	**0.9192**
	Entropy	0.5552	0.6336	0.6502	0.6814	0.6778	0.6834	0.6821	0.6520
	Density	0.6203	0.6130	0.6443	0.6223	0.6459	0.6668	0.6794	0.6417
	QBC	0.5600	0.5997	0.6269	0.6588	0.6620	0.6701	0.6920	0.6385
sonar	Random	0.5651	0.5778	0.6116	0.6362	0.6602	0.6382	0.6774	0.6238
	TWD_Active	0.5286	0.6223	0.6209	0.6449	0.6774	0.6648	0.6894	0.6355
	AL_NE	0.5552	0.6336	0.6443	**0.6874**	0.6914	0.6814	0.6916	0.6550
	ALTD_NE	**0.6396**	**0.6801**	**0.6602**	**0.6874**	**0.7186**	**0.7399**	**0.7312**	**0.6939**
	Entropy	0.6689	0.6814	0.6935	0.7026	0.7132	0.7235	0.7344	0.7025
	Density	0.6704	0.6742	0.6729	0.6892	0.6907	0.6935	0.7107	0.6859
	QBC	0.6779	0.6861	0.6954	0.7113	0.7129	0.7288	0.7391	0.7074
vehicle	Random	**0.6829**	0.6882	0.6954	0.7070	0.7085	0.7166	0.7244	0.7033
	TWD_Active	0.6620	0.6717	0.6848	0.6857	0.7004	0.7129	0.7288	0.6923
	AL_NE	0.6689	0.6814	0.6935	0.7035	0.7134	0.7257	0.7385	0.7036
	ALTD_NE	0.6773	**0.6982**	**0.7092**	**0.7210**	**0.7329**	**0.7416**	**0.7529**	**0.7190**
	Entropy	0.8513	0.8523	0.8735	0.8804	0.9092	0.9122	**0.9260**	0.8864
	Density	0.8362	0.8726	0.8798	0.8919	0.8850	0.8873	0.9037	0.8795
	QBC	0.8719	0.8575	0.8726	0.8981	0.9040	0.9143	0.9220	0.8915
wine	Random	0.8578	0.8644	0.8729	0.8640	0.8686	0.8749	0.8749	0.8682
	TWD_Active	**0.8834**	0.8811	0.8883	0.8935	0.9020	0.9073	0.9151	0.8958
	AL_NE	0.8513	0.8523	0.8735	0.8804	0.9092	0.9122	**0.9260**	0.8864
	ALTD_NE	0.8821	**0.8870**	**0.9030**	**0.9151**	**0.9119**	**0.9260**	0.9246	**0.9071**

Table 3. Comparison of F1 Value on biodeg, credit, page-blocks, steel and yeast

Dataset	Algorithm	10%	20%	30%	40%	50%	60%	80%	Mean
biodeg	Entropy	0.7953	0.8016	0.8290	0.8329	0.8384	0.8369	0.8298	0.8234
	Density	0.7795	0.8174	0.8116	0.8361	0.8384	0.8308	0.8395	0.8219
	QBC	0.8022	0.8203	0.8329	0.8484	0.8298	0.8432	**0.8524**	0.8327
	Random	0.8169	0.8140	0.8171	0.8229	0.8329	0.8079	0.8424	0.8220
	TWD_Active	0.7953	0.8263	0.8237	0.8298	0.8271	0.8395	0.8453	0.8267
	AL_NE	0.8300	0.8340	0.8424	0.8397	0.8384	0.8369	0.8371	0.8369
	ALTD_NE	**0.8379**	**0.8529**	**0.8587**	**0.8495**	**0.8526**	**0.8468**	**0.8524**	**0.8501**
credit	Entropy	0.7184	0.7235	0.7201	0.7235	0.7235	0.7349	0.7366	0.7258
	Density	0.6868	0.7043	0.7153	0.7187	0.7222	0.7249	0.7253	0.7139
	QBC	0.6252	0.6844	0.7170	0.7198	0.7198	0.7249	0.7366	0.7040
	Random	0.7046	0.7101	0.7115	0.7153	0.7153	0.7153	0.7153	0.7125
	TWD_Active	0.7211	0.7222	0.7201	0.7198	0.7187	0.7153	0.7201	0.7196
	AL_NE	**0.7390**	0.7363	0.7318	0.7318	0.7338	0.7338	0.7356	0.7346
	ALTD_NE	0.7376	**0.7411**	**0.7411**	**0.7397**	**0.7397**	**0.7414**	**0.7398**	**0.7401**
page-blocks	Entropy	0.9595	0.9603	0.9584	0.9603	0.9608	0.9614	0.9620	0.9604
	Density	0.9626	0.9609	0.9658	**0.9658**	**0.9641**	0.9560	0.9590	0.9620
	QBC	0.9632	0.9560	0.9572	0.9603	**0.9641**	0.9620	0.9590	0.9603
	Random	0.9614	0.9597	0.9597	0.9619	0.9632	0.9649	0.9632	0.9620
	TWD_Active	0.9595	0.9554	0.9541	0.9541	0.9541	0.9553	0.9553	0.9554
	AL_NE	0.9589	0.9531	0.9524	0.9530	0.9512	0.9524	0.9535	0.9535
	ALTD_NE	**0.9658**	**0.9658**	**0.9666**	**0.9658**	**0.9641**	**0.9659**	**0.9640**	**0.9654**
steel	Entropy	0.6639	**0.7368**	**0.7446**	0.6941	0.7247	0.7247	0.7345	0.7176
	Density	0.7135	0.6537	0.7339	0.6941	0.7043	0.7247	**0.7349**	0.7084
	QBC	0.6732	0.6844	0.6844	0.6746	0.7349	0.7349	0.7266	0.7019
	Random	0.6115	0.6416	0.6941	0.7242	0.7463	0.7463	0.7266	0.6987
	TWD_Active	0.6333	0.6844	0.7150	0.6941	0.6435	0.6129	0.7052	0.6698
	AL_NE	0.6639	0.6241	0.6134	0.7155	0.7043	0.7349	0.7266	0.6832
	ALTD_NE	**0.7368**	**0.7368**	0.7261	**0.7665**	**0.7757**	**0.7466**	0.7266	**0.7450**
yeast	Entropy	0.7168	0.7168	0.7168	0.7540	0.7779	0.7913	0.8176	0.7559
	Density	0.7655	0.7044	0.7904	0.8018	0.8018	0.8152	0.8161	0.7850
	QBC	0.7283	0.7283	0.7531	0.7779	0.7779	0.8396	0.8161	0.7745
	Random	**0.8009**	0.7283	0.7531	0.7779	0.7540	0.7775	0.7669	0.7655
	TWD_Active	0.7044	0.7044	0.8411	0.8018	0.8147	0.8267	**0.8271**	0.7886
	AL_NE	0.7779	0.7899	0.8185	**0.8405**	**0.8520**	0.8396	0.8161	0.8192
	ALTD_NE	0.7655	**0.8295**	**0.8415**	**0.8405**	**0.8520**	**0.8400**	0.8176	**0.8267**

3) From the aspect of the user annotation amount, ALTD_NE can reduce the percentages of queried instances. Especially in biodeg and credit, ALTD_NE only requires 10% labeled training samples to achieve the best classification accuracy of other algorithms.

4) As the label of unlabeled samples is continuously queried, the classification performance is on the rise and tends to a certain fixed value.

5) Compared with other comparison algorithms, the AL_NE and ALTD_NE have shown little improvement in performance on most datasets in the early stage; while in the middle and late stages of algorithm execution, the classification performance of AL_NE and ALTD_NE has been significantly improved. Especially in the datasets ecoli, heart, and pima, this feature is particularly obvious.

6) The average values of ACC, F1 Value and AUC of the AL_NE algorithm have been slightly improved compared to the Entropy algorithm. This justifies that the neighborhood entropy strategy is more efficient in selecting the most valuable samples than the entropy strategy.

7) Under the same experimental environment, the average levels of ACC, F1 Value, and AUC of the ALTD_NE algorithm have been better than other algorithms. In general, the ALTD_NE has achieved obvious performance on most datasets, which validates that

TABLE 4. Comparison of AUC on biodeg, credit, page-blocks, steel and yeast

Dataset	Algorithm	10%	20%	30%	40%	50%	60%	80%	Mean
biodeg	Entropy	0.8153	0.8179	0.8224	0.8252	0.8388	0.8693	0.8763	0.8379
	Density	0.8190	0.8204	0.8278	0.8272	0.8506	0.8309	0.8424	0.8312
	QBC	0.8153	0.8179	0.8396	0.8368	0.8484	0.8577	0.8645	0.8400
	Random	0.8170	0.8060	0.8407	0.8238	0.8283	0.8280	0.8568	0.8287
	TWD_Active	0.8396	0.8365	0.8314	0.8410	0.8506	0.8518	0.8645	0.8451
	AL_NE	**0.8436**	0.8464	0.8504	0.8385	0.8444	0.8549	0.8588	0.8481
	ALTD_NE	**0.8436**	**0.8633**	**0.8656**	**0.8679**	**0.8707**	**0.8814**	**0.9001**	**0.8704**
credit	Entropy	0.6403	0.6379	0.6379	0.6388	0.6499	0.6268	0.6217	0.6362
	Density	**0.6592**	**0.6607**	0.6340	0.6415	0.6460	0.6460	0.6411	0.6469
	QBC	0.6319	0.6337	0.6169	0.5997	0.6021	0.6223	0.6229	0.6185
	Random	0.6376	0.6490	0.6379	0.6211	0.6738	0.6358	0.6472	0.6432
	TWD_Active	0.6334	0.6316	0.6223	0.6244	0.6054	0.5991	0.5991	0.6165
	AL_NE	0.6508	0.6544	0.6340	0.6310	0.6109	0.6343	0.6076	0.6319
	ALTD_NE	0.6562	0.6568	**0.6793**	**0.6760**	**0.6747**	**0.6655**	**0.6472**	**0.6651**
page-blocks	Entropy	0.9508	0.9432	0.9476	0.9439	0.9524	0.9496	0.9471	0.9478
	Density	0.9493	0.9482	0.9484	0.9483	0.9480	0.9482	0.9472	0.9482
	QBC	0.9523	0.9482	0.9393	0.9411	0.9517	0.9496	0.9478	0.9471
	Random	0.9500	0.9512	0.9427	0.9423	0.9524	0.9513	0.9504	0.9486
	TWD_Active	0.9408	0.9327	0.9349	0.9356	0.9450	0.9456	0.9485	0.9404
	AL_NE	0.9513	0.9506	0.9506	**0.9539**	**0.9533**	**0.9513**	0.9504	0.9516
	ALTD_NE	**0.9566**	**0.9562**	**0.9512**	0.9521	**0.9533**	0.9506	**0.9516**	**0.9531**
steel	Entropy	0.8043	**0.8074**	0.7959	**0.8034**	**0.8060**	0.7949	0.7888	0.8001
	Density	0.8109	0.7888	0.7943	0.7885	0.7906	0.7810	0.7898	0.7920
	QBC	0.7971	0.7993	0.7761	0.7885	0.7906	0.7749	0.7758	0.7860
	Random	0.7867	0.7876	0.7888	0.7812	0.7918	0.8008	0.7912	0.7897
	TWD_Active	0.7810	0.7818	0.7779	0.7779	0.7715	0.7636	0.7758	0.7756
	AL_NE	**0.8120**	0.7993	0.7958	0.7949	0.7946	0.8083	0.7974	0.8003
	ALTD_NE	0.7937	**0.8074**	**0.8077**	0.8007	**0.8060**	**0.8101**	**0.8011**	**0.8038**
yeast	Entropy	0.8489	**0.8527**	0.8498	0.8456	0.8433	0.8446	0.8480	0.8476
	Density	0.8307	0.8291	0.8280	0.8363	0.8413	0.8404	0.8436	0.8356
	QBC	**0.8504**	0.8494	0.8519	0.8459	0.8487	0.8487	0.8551	0.8500
	Random	0.8288	0.8414	0.8441	0.8375	0.8378	0.8375	0.8390	0.8380
	TWD_Active	0.8462	0.8473	0.8519	0.8481	0.8485	0.8500	0.8493	0.8488
	AL_NE	0.8462	0.8482	0.8478	0.8428	0.8430	0.8470	0.8434	0.8455
	ALTD_NE	0.8489	0.8509	**0.8551**	**0.8575**	**0.8562**	**0.8572**	**0.8570**	**0.8547**

ALTD_NE can further explore the data distribution while focusing on regions that are most informative.

According to the above observation, it can be concluded that the Entropy algorithm can select the most informative sample, but may lead to an information redundancy. The Density algorithm considers sample distribution information but often introduces some samples with low informativeness. The model created by the Random algorithm is unstable as it selects samples randomly. The QBC algorithm favors samples with the most inconsistent classification results by constructing multiple classifiers, but the distribution information of samples is ignored. The TWD_Active algorithm considers the problem that samples in different domains have different worth, but some informative samples located in the same density area will be chosen since the classification result depends on the decision function. The AL_NE algorithm takes neighborhood entropy as the selection criterion, which takes account of the informativeness and distribution information of a sample simultaneously. The neighborhood entropy is adopted as the decision function by the ALTD_NE algorithm, which has been improved over the baseline methods significantly.

5.3. **The influence of parameter on the algorithm.** There is an important parameter λ that controls the radius of the neighborhood in the proposed methods. A smaller λ leads

Table 5. Comparison of AUC on austra, bupa, ecoli, heart and lead7digit, etc.

Dataset	Algorithm	10%	20%	30%	40%	50%	60%	80%	Mean
austra	Entropy	0.7868	0.7949	0.7995	0.8067	0.8143	0.8149	0.8237	0.8058
	Density	0.7827	0.7782	0.7981	0.8005	0.8005	0.8102	0.8030	0.7962
	QBC	0.7857	**0.8091**	0.8140	0.8113	0.8140	0.8190	0.8269	0.8114
	Random	0.7872	0.7963	0.7996	0.8066	0.8100	0.8134	0.8280	0.8059
	TWD_Active	0.7840	0.7877	0.7945	0.8156	0.8167	0.8203	0.8251	0.8063
	AL_NE	0.7839	0.7852	0.8003	0.8104	0.8179	0.8239	**0.8401**	0.8088
	ALTD_NE	**0.7938**	0.8082	**0.8314**	**0.8271**	**0.8291**	**0.8316**	0.8356	**0.8224**
bupa	Entropy	0.5924	0.6098	0.6148	0.6253	0.6392	0.6400	0.6589	0.6258
	Density	0.5816	0.6136	0.6278	**0.6562**	**0.6644**	0.6420	0.6536	0.6342
	QBC	**0.6162**	0.6202	0.6221	0.6332	0.6457	**0.6711**	0.6638	0.6389
	Random	0.6122	0.6134	0.6352	0.6426	0.6550	0.6562	0.6587	0.6390
	TWD_Active	0.6006	0.6159	0.6278	0.6403	0.6471	0.6502	0.6397	0.6317
	AL_NE	0.5935	0.6097	0.6179	0.6304	0.6434	0.6417	0.6457	0.6260
	ALTD_NE	0.6026	**0.6312**	**0.6440**	0.6494	0.6638	0.6709	**0.6690**	**0.6473**
ecoli	Entropy	0.6999	0.6481	0.6211	0.6550	0.5692	0.5360	0.5554	0.6121
	Density	0.7553	0.6972	**0.7179**	0.5993	0.6889	0.6571	0.6322	0.6783
	QBC	0.6432	0.6315	0.5789	0.5429	0.6259	0.6398	0.5623	0.6035
	Random	0.6806	0.6176	0.6889	0.6266	0.6162	0.6695	0.6909	0.6558
	TWD_Active	**0.7636**	0.6840	0.5471	0.5581	0.5450	0.5319	0.5422	0.5960
	AL_NE	0.6999	0.6481	0.6211	0.6550	0.5692	0.5360	0.5554	0.6121
	ALTD_NE	0.7546	**0.7269**	0.7138	**0.6999**	**0.7013**	**0.7158**	**0.6730**	**0.7122**
heart	Entropy	0.6923	0.7069	0.7187	0.7321	0.7326	0.7440	0.7729	0.7285
	Density	0.6807	0.6965	0.7145	0.7141	0.7298	0.7377	0.7588	0.7189
	QBC	0.6719	0.6932	0.7039	0.7325	0.7414	0.7368	0.7581	0.7197
	Random	0.6965	0.7122	0.7368	0.7483	0.7525	0.7562	0.7641	0.7381
	TWD_Active	**0.7048**	**0.7275**	0.7298	0.7414	0.7738	**0.7877**	0.7840	0.7499
	AL_NE	0.6923	0.7069	0.7187	0.7321	0.7326	0.7440	0.7729	0.7285
	ALTD_NE	0.7002	0.7157	**0.7511**	**0.7724**	**0.7924**	**0.7877**	**0.8052**	**0.7607**
led7digit	Entropy	0.6919	0.7436	0.7478	0.7415	0.7539	0.7487	**0.7477**	0.7393
	Density	0.7911	**0.7762**	0.7652	**0.7766**	0.7084	0.7043	0.6994	0.7459
	QBC	0.7001	0.6836	0.6981	0.6164	0.6206	0.6722	0.6609	0.6646
	Random	0.5606	0.4686	0.4573	0.5348	0.6888	0.5058	0.5854	0.5430
	TWD_Active	0.7415	0.7343	0.6402	0.6216	0.6350	**0.7653**	0.6547	0.6847
	AL_NE	0.6919	0.7436	0.7560	0.7735	0.7539	0.7449	**0.7477**	0.7445
	ALTD_NE	**0.8045**	0.7074	**0.7756**	0.7746	**0.7611**	0.7453	0.7363	**0.7578**
pima	Entropy	0.6097	0.6130	0.6239	0.6308	0.6249	0.6286	0.6391	0.6243
	Density	0.6015	0.6108	0.6101	0.6095	0.6060	0.6154	0.6275	0.6115
	QBC	0.6079	0.6188	0.6251	0.6278	**0.6424**	0.6300	0.6498	0.6288
	Random	0.6103	0.6151	0.6253	**0.6317**	0.6280	0.6300	0.6456	0.6266
	TWD_Active	0.6097	0.6165	0.6251	0.6275	0.6206	0.6388	0.6415	0.6257
	AL_NE	**0.6134**	**0.6324**	0.6293	**0.6317**	0.6411	**0.6461**	**0.6552**	**0.6356**
	ALTD_NE	0.6047	0.6178	**0.6339**	0.6284	0.6288	0.6388	0.6486	0.6287
segment	Entropy	**0.8381**	0.8356	0.8348	0.8363	0.8301	0.8296	0.8362	0.8344
	Density	0.8356	0.8356	0.8356	0.8356	0.8356	0.8356	0.8347	0.8355
	QBC	0.8373	0.8286	0.8218	0.8322	0.8324	0.8345	0.8360	0.8318
	Random	0.8293	0.8305	0.8268	0.8316	0.8344	0.8364	0.8384	0.8325
	TWD_Active	0.8363	0.8338	0.8338	0.8338	0.8338	0.8370	0.8362	0.8350
	AL_NE	0.8347	0.8300	**0.8373**	0.8373	0.8323	0.8347	0.8330	0.8342
	ALTD_NE	0.8373	**0.8372**	0.8308	**0.8380**	**0.8380**	**0.8414**	**0.8414**	**0.8377**
sonar	Entropy	**0.6472**	0.6609	0.6794	0.6743	0.6894	0.7020	0.7035	0.6795
	Density	0.6055	0.6558	0.6693	0.7095	0.7160	0.7009	0.7251	0.6832
	QBC	0.6407	0.6608	0.6638	0.6787	0.6959	0.7004	0.7311	0.6816
	Random	0.6106	0.6437	0.6512	0.6869	0.7049	0.6924	0.6969	0.6695
	TWD_Active	0.6206	0.6578	0.6789	0.6824	0.6999	0.7040	**0.7321**	0.6822
	AL_NE	0.6437	0.6648	**0.6909**	0.6784	0.7040	0.7065	0.7165	0.6864
	ALTD_NE	0.6000	**0.6778**	0.6844	**0.7281**	**0.7180**	**0.7286**	0.7276	**0.6949**
vehicle	Entropy	0.5842	0.5857	0.5672	0.5371	0.5319	0.5450	0.5587	0.5585
	Density	0.5998	0.5783	0.5567	0.5678	0.5809	**0.6201**	0.5221	0.5751
	QBC	0.5724	0.5770	0.6064	0.6018	**0.6240**	0.5907	**0.6476**	0.6028
	Random	0.4881	0.5224	0.5750	0.5633	0.5319	0.5006	0.4581	0.5199
	TWD_Active	0.5986	0.5816	0.5652	0.4908	0.5149	0.5175	0.5300	0.5427
	AL_NE	0.5842	0.5857	0.5672	0.5417	0.5443	0.5848	0.6221	0.5757
	ALTD_NE	**0.6319**	**0.6071**	**0.6155**	**0.6240**	0.6208	0.6129	0.6221	**0.6192**
wine	Entropy	0.5007	0.5169	0.5166	0.5285	0.5341	0.5348	0.5447	0.5252
	Density	**0.5250**	0.5152	0.5191	0.5204	0.5309	0.5332	**0.5477**	0.5274
	QBC	0.4955	0.5184	0.5309	0.5245	0.5370	**0.5435**	0.5440	0.5277
	Random	0.5142	0.5086	0.5167	0.5240	0.5152	0.5214	0.5353	0.5193
	TWD_Active	0.5091	0.5060	0.5247	0.5186	0.5166	0.5277	0.5413	0.5206
	AL_NE	0.5007	0.5169	0.5166	0.5285	0.5341	0.5348	0.5447	0.5252
	ALTD_NE	0.5122	**0.5277**	**0.5331**	**0.5373**	**0.5440**	0.5417	0.5410	**0.5339**

TABLE 6. Influence of the parameter

λ	Algorithm	ACC	F1 Value	AUC
0.2	AL_NE	0.6775	0.6685	0.7590
	ALTD_NE	0.7019	0.6978	0.7663
0.4	AL_NE	0.6821	0.6772	0.7633
	ALTD_NE	0.7034	0.6911	0.7740
0.6	AL_NE	0.6987	0.6728	0.7740
	ALTD_NE	0.7126	0.6962	**0.7810**
0.8	AL_NE	0.7168	0.6938	**0.7763**
	ALTD_NE	**0.7179**	**0.7056**	0.7777
0.9	AL_NE	**0.7170**	**0.6990**	0.7737
	ALTD_NE	0.7159	0.7026	0.7793

to inefficiency of select samples, whereas the large λ seems difficult to combine distribution information for measuring the representativeness of samples. The neighborhood entropy is equal to information entropy when λ is set to 0. To further analyze the influence of parameters on the performance, the austra, led7digit, and sonar data are applied to testing the performance of the two algorithms. In Table 6, the λ has been set to 0.2, 0.4, 0.6, 0.8, and 0.9 respectively to show the average values of ACC, F1 Value, and AUC. Under different parameters, the best results on different metrics are shown in bold.

The experimental results show that 1) The performance of the algorithm increases gradually with the parameter increased, while performance begins to decrease when λ reaches 0.8; 2) The parameter 0.8 in the AL_NE and ALTD_NE algorithms obviously leads to a better and stable classification performance, which does not guarantee the best performance on all datasets but the differences are not that severe.

5.4. Application on ALTD_NE.

5.4.1. *Dataset of practical application.* The PC1 dataset (NASA) contains 1109 records of software defects, which comes from McCabe and Halstead features extractors of source code and contains 22 features that attempt to objectively characterize code features that are associated with software quality. In this case, each record is classified as either non-defective or defective.

5.4.2. *Setting.* The training/testing instances segmentation is consistent with the previous section, the comparing algorithms include Entropy, Density, QBC, Random, TWD_Active, AL_NE, and ALTD_NE. The classification accuracy (ACC) is applied to verifying the experimental results. To get rid of the influence of randomness, we average 10 times execution results as the final result.

5.4.3. *Comparison with state-of-the-art methods.* Figure 2 demonstrates the classification accuracy of our method compared to the baseline algorithms on the real problem, ALTD_NE achieves consistently better performance across the different numbers of labeled instances. The average ACC of ALTD_NE on dataset PC1 is 1.5% higher than various benchmark algorithms. The performance gap is clearer especially with increasing the number of queries where the selected instances contributed to model training. The Random and TWD_Active suffer from a significant gap in performance when compared with ALTD_NE. In general, empirical experiments demonstrate that the proposed ALTD_NE can achieve promising results in practical applications and exceed the established baseline.

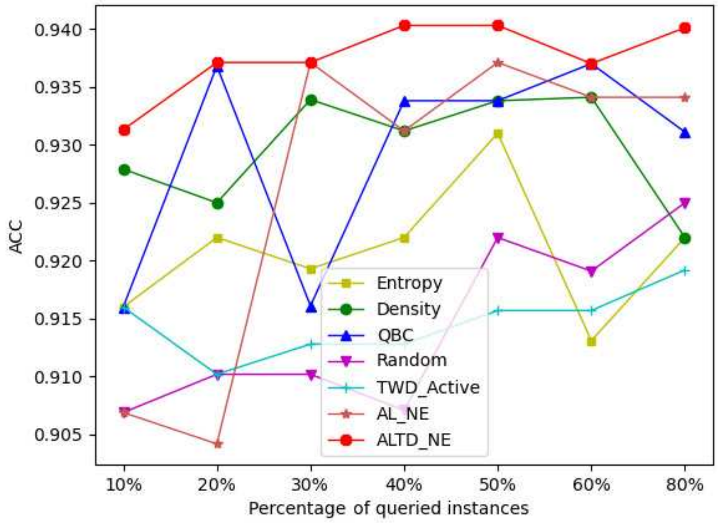


Figure 2. Comparison on ACC

6. **Conclusions.** An active learning method based on neighborhood entropy is proposed according to the entropy strategy, which favors the selection of both informative and representative samples by combining the sample distribution information. Further, we develop active learning of three-way decision based on neighborhood entropy, which takes the neighborhood entropy as the decision function of the three-way decision to select the most valuable sample in different domains, thereby improving the performance of the classifier. Results on benchmark datasets and the dataset of practical application show a slight improvement of the classification performance with AL_NE compared to the entropy method; the ALTD_NE outperforms state-of-the-art methods significantly in reducing the manual labeling cost while achieving the same classification performance. For future work, we intend to a faster computational efficiency by dimensionality reduction and feature selection, as the calculation of neighborhood entropy is more complicated for the high-dimensional datasets.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61563012), and the Guangxi Key Laboratory of Embedded Technology and Intelligent System Foundation (No. 2019-1-4).

REFERENCES

[1] S. Huang, R. Jin and Z. Zhou, Active learning by querying informative and representative examples, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.36, no.10, pp.1936-1949, 2014.
[2] F. Hu, M. Zhang and H. Yu, An active learning method based on three-way decision model, *Control and Decision*, vol.34, no.4, pp.718-726, 2019.
[3] B. Settles, *Active Learning Literature Survey*, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.
[4] Z. Zhou, A brief introduction to weakly supervised learning, *National Science Review*, vol.5, no.1, pp.44-53, 2018.
[5] K. Konyushkova, R. Sznitman and P. Fua, Learning active learning from data, *arXiv Preprint*, arXiv: 1703.03365, 2017.
[6] Z. Liu and D. Wu, Integrating informativeness, representativeness and diversity in pool-based sequential active learning for regression, *2020 International Joint Conference on Neural Networks*, pp.1-7, 2020.
[7] P. Bachman, A. Sordoni and A. Trischler, Learning algorithms for active learning, *International Conference on Machine Learning*, pp.301-310, 2017.

[8] B. Du, Z. Wang and L. Zhang, Exploring representativeness and informativeness for active learning, *IEEE Trans. Cybernetics*, vol.47, no.1, pp.14-26, 2015.

[9] J. Bappy, S. Paul and E. Tuncel, The impact of typicality for informative representative selection, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.771-780, 2017.

[10] W. Shao, S. Huang and M. Liu, Querying representative and informative super-pixels for filament segmentation in bioimages, *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol.17, no.4, pp.1394-1405, 2020.

[11] Z. Wang and J. Ye, Querying discriminative and representative samples for batch mode active learning, *ACM Trans. Knowledge Discovery from Data*, vol.9, no.3, pp.61-83, 2015.

[12] S. Huang, J. Chen and X. Mu, Cost-effective active learning from diverse labelers, *International Joint Conference on Artificial Intelligence*, pp.1879-1885, 2017.

[13] D. Wu, Pool-based sequential active learning for regression, *IEEE Trans. Neural Networks and Learning Systems*, vol.30, no.5, pp.1348-1359, 2018.

[14] A. Holub, P. Perona and M. Burl, Entropy-based active learning for object recognition, *IEEE Conference on Computer Vision and Pattern Recongnition*, pp.1-8, 2008.

[15] Y. Freund, H. Seung and E. Shamir, Selective sampling using the query by committee algorithm, *Machine Learning*, vol.28, no.2, pp.133-168, 1997.

[16] N. Abe and H. Mamitsuka, Query learning strategies using boosting and bagging, *Proc. of the 15th International Conference on Machine Learning*, pp.1-9, 1998.

[17] Y. Yang and M. Loog, Active learning using uncertainty information, *2016 23rd International Conference on Pattern Recognition*, pp.2646-2651, 2016.

[18] G. Wang, J. Hwang and C. Rose, Uncertainty-based active learning via sparse modeling for image classification, *IEEE Trans. Image Processing*, vol.28, no.1, pp.316-329, 2018.

[19] M. Balcan, A. Broder and T. Zhang, Margin based active learning, *Conference on Learning Theory*, pp.35-50, 2017.

[20] N. Roy and A. Mccallum, Toward optimal active learning through sampling estimation of error reduction, *Proc. of the 18th International Conference on Machine Learning*, pp.441-448, 2001.

[21] H. Nguyen and A. Smeulders, Active learning using pre-clustering, *Proc. of the 21st International Conference on Machine Learning*, pp.623-630, 2004.

[22] R. Chattopadhyay, Z. Wang and W. Fan, Batch mode active sampling based on marginal probability distribution matching, *ACM Trans. Knowledge Discovery from Data*, vol.7, no.3, pp.1-25, 2013.

[23] T. Wang, S. Huang and Z. Zhou, Towards identifying causal relation between instances and labels, *Proc. of the 2019 SIAM International Conference on Data Mining*, pp.289-297, 2019.

[24] K. Yu, J. Bi and V. Tresp, Active learning via transductive experimental design, *Proc. of the 23rd International Conference on Machine Learning*, pp.1081-1088, 2006.

[25] Z. Xu, K. Yu, V. Tresp, X. Xu and J. Wang, Representative sampling for text classification using support vector machines, in *Advances in Information Retrieval. ECIR 2003. Lecture Notes in Computer Science*, F. Sebastiani (ed.), Berlin, Springer-Verlag, 2003.

[26] P. Donmez, J. G. Carbonell and P. N. Bennett, Dual strategy active learning, in *Machine Learning: ECML 2007. ECML 2007. Lecture Notes in Computer Science*, Berlin, Springer-Verlag, 2008.

[27] Y. Yao, Three-way decisions with probabilistic rough sets, *Information Sciences*, vol.180, no.3, pp.341-353, 2010.

[28] Q. Hu, D. Yu and Z. Xie, Neighborhood classifiers, *Expert Systems with Applications*, vol.34, no.2, pp.866-876, 2008.

[29] S. Ebert, M. Fritz and B. Schiele, RALF: A reinforced active learning formulation for object class recognition, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.3626-3633, 2012.

[30] Y. Yao, Three-way decision and granular computing, *International Journal of Approximate Reasoning*, vol.103, pp.107-123, 2018.

[31] C. Zhang, S. Chai, L. Cui and B. Zhang, Road condition recognition in self-driving cars based on classification and regression tree, *ICIC Express Letters, Part B: Applications*, vol.10, no.12, pp.1115-1122, 2019.

[32] V. V. Meduri, L. Popa, P. Sen and M. Sarwat, A comprehensive benchmark framework for active learning methods in entity matching, *Proc. of the 2020 ACM SIGMOD International Conference on Management of Data*, pp.1133-1147, 2020.

[33] X. Cao, J. Yao, Z. Xu and D. Meng, Hyperspectral image classification with convolutional neural network and active learning, *IEEE Trans. Geoscience and Remote Sensing*, vol.58, no.7, pp.4604-4616, 2020.

[34] C. Li, C. Yang, L. Liang, Y. Yuan and G. Wang, On robust grouping active learning, *IEEE Trans. Emerging Topics in Computational Intelligence*, 2020.
[35] K. Madhawa and T. Murata, Active learning for node classification: An evaluation, *Entropy*, vol.22, no.10, DOI: 10.3390/e22101164, 2020.

Author Biography

Qiuyue Lv is a postgraduate student in the Department of Information Science and Engineering of Guilin University of Technology. Her main research interests include machine learning and data mining.

Minggang Dong obtained his Ph.D. degree in control science and engineering from Zhejiang University, China in 2012; he visited the University of Rhode Island for joint doctoral training from October 2016 to October 2017.

Prof. Dong is currently the vice dean of the Department of Information Science and Engineering at Guilin University of Technology. His main research interests include intelligent computing and application, and machine learning. He has published more than 50 papers in well-known journals at home and abroad. He is hosting some research projects funded from the National Natural Science Foundation of China.