Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A novel active learning framework for classification: Using weighted rank aggregation to achieve multiple query criteria

Yu Zhao, Zhenhui Shi, Jingyang Zhang, Dong Chen, Lixu Gu*

Image-Guided Surgery and Virtual Reality Lab, School of Biomedical Engineering, Shanghai Jiao Tong University, Dongchuan Road 800, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 9 January 2018
Revised 30 January 2019
Accepted 30 March 2019
Available online 4 May 2019

Keywords:
Multiple query criteria active learning
Integration criteria strategy
Sample query criterion
Weighted rank aggregation

ABSTRACT

Multiple query criteria active learning methods have a higher potential performance than conventional active learning methods in which only one criterion is deployed for sample selection. A central issue related to multiple query criteria active learning methods concerns the development of an integration criteria strategy that makes full use of all criteria. The conventional integration criteria strategies adopted in relevant research facilitate the desired effects, but several limitations still must be addressed. For instance, some of the strategies are not sufficiently scalable during the design process, and the number and type of criteria involved are dictated. Thus, it is challenging for the user to integrate other criteria into the original process unless modifications are made to the algorithm. Other strategies are too dependent on empirical parameters, which can be acquired only by experience or cross-validation and thus lack generality; additionally, these strategies are counter to the intention of active learning, as samples need to be labeled in the validation set before the active learning process can begin.

To address these limitations, we propose a novel multiple query criteria active learning method for classification tasks that employs a third strategy via weighted rank aggregation. The proposed method serves as a heuristic means to select high-value samples of high scalability and generality and is implemented through a three-step process: (1) the transformation of the sample selection to sample ranking and scoring, (2) the computation of the self-adaptive weights of each criterion, and (3) the weighted aggregation of each sample rank list. Ultimately, the sample at the top of the aggregated ranking list is the most comprehensively valuable and must be labeled. Several experiments generating 419 wins, 226 ties and 55 losses against other state-of-the-art multiple query criteria-based methods are conducted to verify that the proposed method can achieve superior results.

1. Introduction

1.1. Motivation

Active learning (AL) is a subfield of machine learning technology that is used to minimize the amount of annotation work that must be executed before training an accurate classification or regression model [1]. AL methods are unique in their use of various sample query criteria (SQC). These methods can help the user select a fraction of the most 'valuable' samples for querying labels from massive volumes of unlabeled data [2–4]. On the basis of differences in the definition of 'valuable', AL methods can be broadly divided into two categories: representativeness and informativeness measure-based approaches [5-6].

As illustrated in Fig. 1, plenty of comparative studies [5,7] of AL methods have shown that most representativeness measure-based AL methods perform better when the number of labeled samples is few, whereas others, especially those that are informativeness measure-based, will usually overtake the former after substantial sampling. In this paper, the above phenomenon is referred to as 'the timeliness of AL'. The main explanation for this phenomenon is that representativeness measure-based AL methods can obtain the entire structure of a database upon their first use. However, these AL methods are not sensitive to samples that are close to the decision boundary, notwithstanding the fact that such samples are probably more important to the prediction model. In addition, informativeness measure-based AL methods always search for 'valuable' samples around the current decision boundary, and the optimal decision boundary cannot be found unless a certain number of samples have already been labeled [8]. In other words, the single

* Corresponding author.
E-mail addresses: lereinion@163.com (Y. Zhao), shizhenhui90@gmail.com (Z. Shi), J.Y.Zhang@sjtu.edu.cn (J. Zhang), chendong8707@126.com (D. Chen), gulixu@sjtu.edu.cn (L. Gu).

https://doi.org/10.1016/j.patcog.2019.03.029
0031-3203/© 2019 Published by Elsevier Ltd.

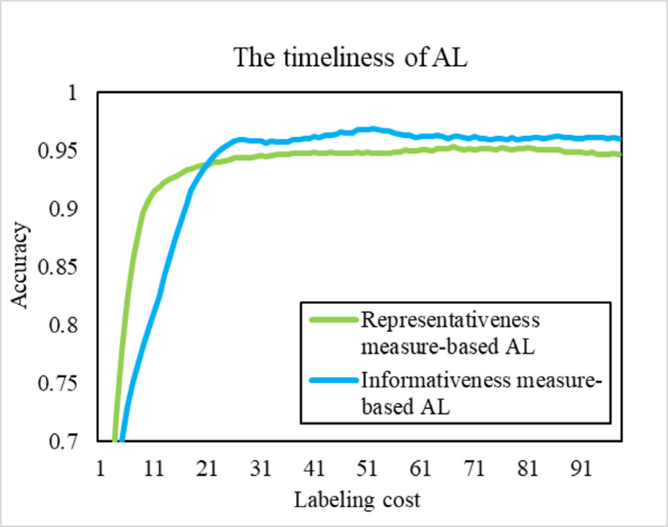


Fig. 1. The timeliness of AL.

query criterion can only guarantee its optimal performance over a period of time in the entire AL process, and the optimal period differs for each criterion.

Considering the above complementary characteristics, recent research reports have similarly proposed that the AL method could likely be improved if more than one SQC were deployed through one AL process to leverage the strengths of all methods [9]. The multiple query criteria AL method (MQCAL) has been developed for this purpose. The MQCAL method can combine most complementary information for each SQC through a special integration criteria strategy. A small number of samples that meet all involved criteria are selected for querying labels. The MQCAL method can in theory be more reliable, effective and resistant to interference because it takes several factors into account rather than focusing on one selection criterion, as is done in conventional AL methods. However, there are still some limitations that exist in MQCAL that require further effort to resolve (e.g., manual weight setting, the impossible combination), which will be described in detail below.

1.2. Related work

Most existing AL methods are based on a single query criterion. Representativeness and informativeness measure-based AL methods are the two main branches of single criterion-based AL methods as shown in Fig. 2. The AL algorithms in the first category rely on the native data structure, and the samples that represent the majority of all samples are regarded as the most representative. According to the data structure expression, representativeness-based AL methods can be further subdivided into three classes that include Clustering Analysis (e.g., *Cluster* [6,10]), Sample Connection (e.g., *Diversity* [11], *Dissimilarity* [5], *Density* [12]), and Experimental Design (e.g., *TED* [13], *MAED* [14], *Random Walks* [15]). In contrast, informativeness measure-based AL methods always select a sample that has a high degree of uncertainty or is able to impart the greatest change to the current model. Based on the number of involved models, this method can also be further subdivided into two classes that include Certainty Based (e.g., *Margin* [1,16], *Entropy* [17], *EER* [18]) and Committee Based (e.g., *QBC* [19] and *Multiple View* [20]).

Compared with the traditional single criterion-based AL methods, existing research about the MQCAL is relatively sparse. Across these few studies, the selection and design of appropriate SQC for combining are usually their main foci of research rather than how to integrate all involved SQC together. After a careful review of existing methods, only four kinds of integration criteria strategies have been found as shown in Fig. 3.

Baram et al. [21] proposed the earliest form of the MQCAL method, as shown in Fig. 3(A). For each iteration of this MQCAL, only one of the involved SQC with the highest criterion selection parameters is applied to choose samples. The criterion selection parameter is a variant of the multi-armed bandit algorithm proposed in [22]. Lughofer E [6] designed a two-phase AL process. In the first phase, the most representative samples based on clustering are selected, and a certainty-based AL approach is applied in the second phase. These MQCAL methods are beneficial primarily in terms of their high levels of efficiency. However, since only one involved query criterion is used in each iteration, their integration criteria strategies are more like criteria selection rather than criteria integration; hence, we refer to such strategies as 'CSAL' for short in this paper.

Shen et al. [23] developed two other integration criteria strategies: parallel-form (shown in Fig. 3(B)) and serial-form (shown in Fig. 3(C)), both of which have been widely used in subsequent studies.

Serial-form MQCAL ('SMQCAL' for short) employs each SQC to select a certain number of samples from the selection results of the previous SQC in sequence as a multilayer filter. On the basis of Shen's work [23], previous reports [4,11] further developed

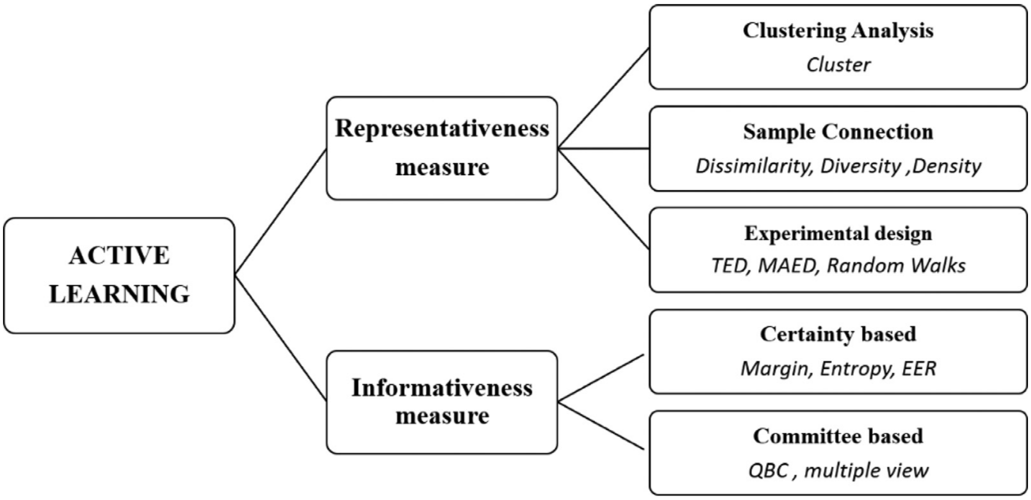


Fig. 2. The categories of traditional AL methods based on a single criterion.

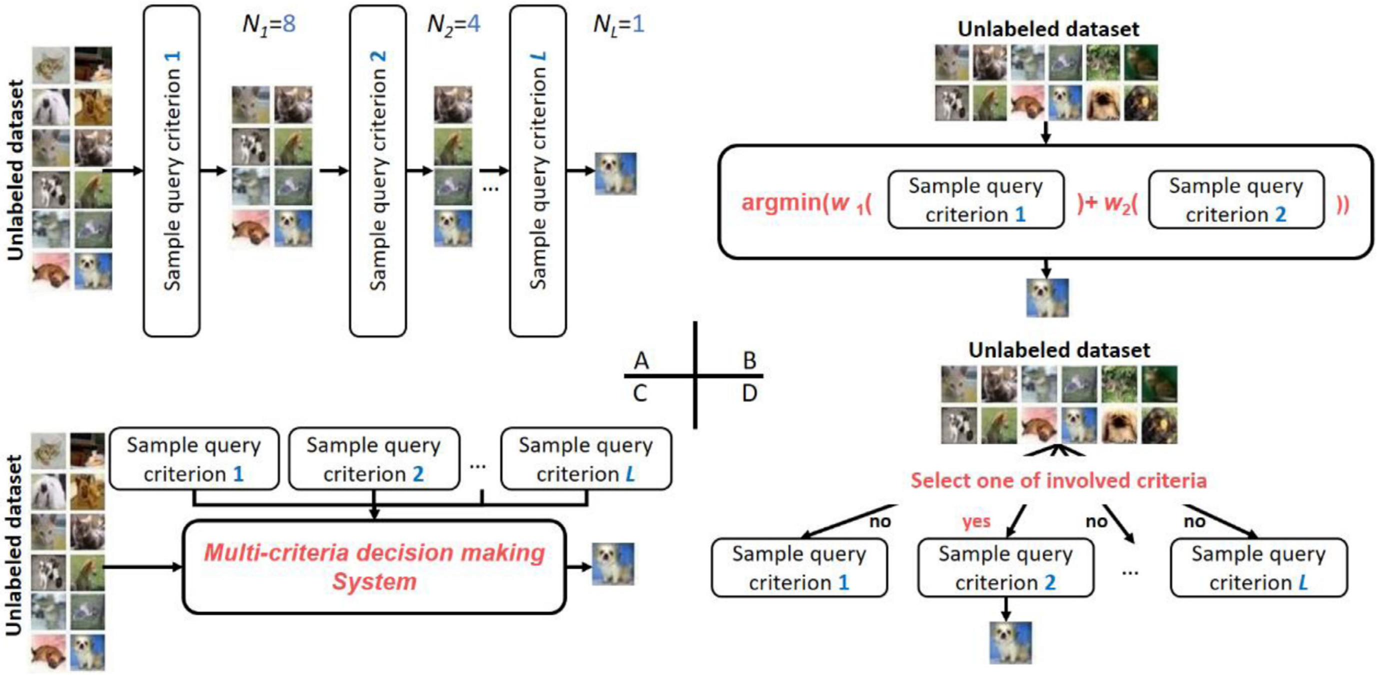


Fig. 3. The existing MQCAL process (A: the process of CSAL, B: the process of PMQCAL, C: the process of SMQCAL, D: the process of MCDMAL).

this approach by combining clustering and uncertainty-based SQCs. Another report [24] applied this method to connect a K nearest neighbor-based cluster algorithm, an SVM margin algorithm and a genetic algorithm to propose an improved AL method for hyperspectral image classification. Moreover, Demir et al. [25] proposed SMQCAL for remote sensing images, which involves SQCs based on uncertainty and diversity. Similar work includes a paper [26] in which samples in low-density regions were selected among the most uncertain samples; low-density regions are determined by exploiting the topological properties of SOM. This approach achieved fast convergence and performed well in both real multispectral remote sensing image classification tasks and hyperspectral remote sensing image classification tasks. In addition to the above classification tasks, Demir et al. [27] demonstrated that their serial-form MQCAL framework can perform well in regression tasks by efficiently identifying most of the diverse samples from high-density regions.

SMQCAL is efficient and operable and is widely used to address practical problems. In addition, the user can directly add several additional SQCs on the basis of the original process, which can be regarded as strongly scalable. However, SMQCAL relies too heavily on two important settings, including the sequence of the applied SQCs and the number of samples selected from each layer (N_i in Fig. 3(C)), which are not generalized.

Parallel-form MQCAL (‘PMQCAL’ for short) can select optimal samples with regard to two different SQCs using a weighted-sum optimization function. Based on this characteristic, previous studies [23,28] have effectively combined uncertainty and diversity to name entity recognition and natural language processing tasks, respectively. In addition, Huang et al. [5] also employed the weighted-sum optimization function to combine the early stage-based SQCs with the representativeness measure-based SQCs (dissimilarity) to acquire satisfactory AL selection results. Other similar studies include recent papers [29,30]. Although the respective criteria used to measure the values of the samples in each are not the same, they all yield satisfactory results using the same basic mechanism, as shown in Fig. 3(B): the weight parameters w_1 and w_2 are used to balance the trade-off between each involved SQC. Notably, although the integration strategy in paper [31] is rendered as the product of two involved SQCs with a high exponent, it still can be regarded as the deformation of a weighted-sum optimization function and classified as a PMQCAL. To further improve parallel MQCAL, Huang et al. [8] developed another systematic way of measuring and combining representativeness and informativeness in the same SVM framework using the min-max AL view. This technique can be regarded as a state-of-the-art MQCAL with strong theoretical capacities.

However, PMQCAL also has two limitations. First, PMQCAL is not scalable. Thus, it is challenging for the user to integrate other SQCs into the original process unless modifications are made to the algorithm. Even so, the optimization function of this extended version may be unsolvable. Second, PMQCAL also places too much reliance on weight parameters. Using the wrong settings can result in suboptimal performance. Most of the above papers suggest that the user can directly use their recommended value [23,28] or obtain the optimal weights through cross-validation [29,30]. Therefore, it is clear that it is also not generalizable for different applied data sets and is even slightly contrary to the original ideal of AL, because it requires the user to prepare some extra labeled samples as a validation set in the cross-validation process. In light of this problem, our group recently published an article [32] designing a double-strategy AL method that is useful for mammographic mass classification, in which the combination weight is selected from predefined candidate values. Of course, this approach is not the best solution because it lacks a fine-tuning procedure. Similarly, due to the expectation maximization concept underlies it, this solution is designed for only two SQCs, further contributing to a lack of scalability.

Additionally, Wang et al. [9] proposed a fourth MQCAL, as shown in Fig. 3(D), by transforming the problem of integration criteria into a multicriteria decision-making system (termed ‘MCDMAL’ here), which also yields good results in the multiple-instance learning environment rather than in a classification task. However, this method has high algorithm complexity. Its implementation and execution are quite difficult, and not every kind of SQC can be integrated into their MQCAL process.

Furthermore, an unsolved problem concerns the establishment of a mechanism that allows for a dynamic and adaptive tradeoff between each SQC that is used for each AL iteration [8,11]. This problem is addressed in most of the above works as a suggested avenue for future research. To our knowledge, only Donmez et.al. [7] has proposed a means of tuning the weights of two SQCs in each iteration by calculating the estimated future residual error reduction level.

1.3. Our approach – the main concept

We realized that the sample selection problem in AL methods is also a sample ranking problem and were inspired by the recommendation technologies that have developed in recent years. Hence, in this manuscript, we develop a novel weighted rank aggregation-based MQCAL for classification tasks, which can be regarded as a fifth form of the integration criteria strategy, which we term 'RMQCAL'.

To implement the proposed method, three additional steps are added to the framework of the original AL process, as shown in Fig. 4. In any iteration of the AL process, all involved SQCs first need to be tweaked in order to invert the problem of sample selection into sample ranking and scoring. Next, every pair of ranking and scoring lists based on their corresponding SQCs can be obtained from the remaining unlabeled samples (see Section 2.2). Third, using the best-versus-second-best (BVSB) strategy, the weights of each SQC for every iteration of the AL process can be dynamically obtained from the current score lists (see Section 2.3). Then, the rank lists of all SQCs involved are weighted and combined as a comprehensive ranked list through our improved weighted rank aggregation method (see Section 2.4). The sample ranked highest in this comprehensive ranked list is then considered to be the most comprehensively valuable and the most in need of labeling for this iteration.

The innovation of this article manifests in both the originality of the study object and the proposed solution. The study object of the proposed RMQCAL focuses on the design of an integration criteria strategy that can integrate each SQC involved rather than designing several specific SQCs and adding them together using empirical weight-parameter settings. Moreover, the solution of the proposed RMQCAL treats criteria integration as a special rank aggregation problem to be solved using a Markov chain; this methodology differs completely from that of earlier studies. In terms of the algorithm itself, RMQCAL has the following advantages over the existing MQCAL. *Scalability:* Similar to serial-form MQCAL, any number and type of SQC can be easily introduced into our RMQCAL process without establishing a more complex optimization function. *Uniformity:* The uniformity of each SQC can be guaranteed by converting sample selection, the purpose of each SQC, into sample ranking. *Generality:* Our method no longer employs any empirical parameters; instead, each tradeoff behind SQCs is self-adaptive. *Dynamics:* As in most papers, except in their future work, the tradeoffs between each SQC used in our method are dynamic and change according to their differential contributions in each iteration of the AL process.

Moreover, RMQCAL offers a potential predominance in practical applications. The aim of AL methods is to reduce the annotation work of unlabeled samples in hand. However, when coping with an unlabeled dataset in real-word problems, in order to select the most appropriate AL method and acquire optimal empirical

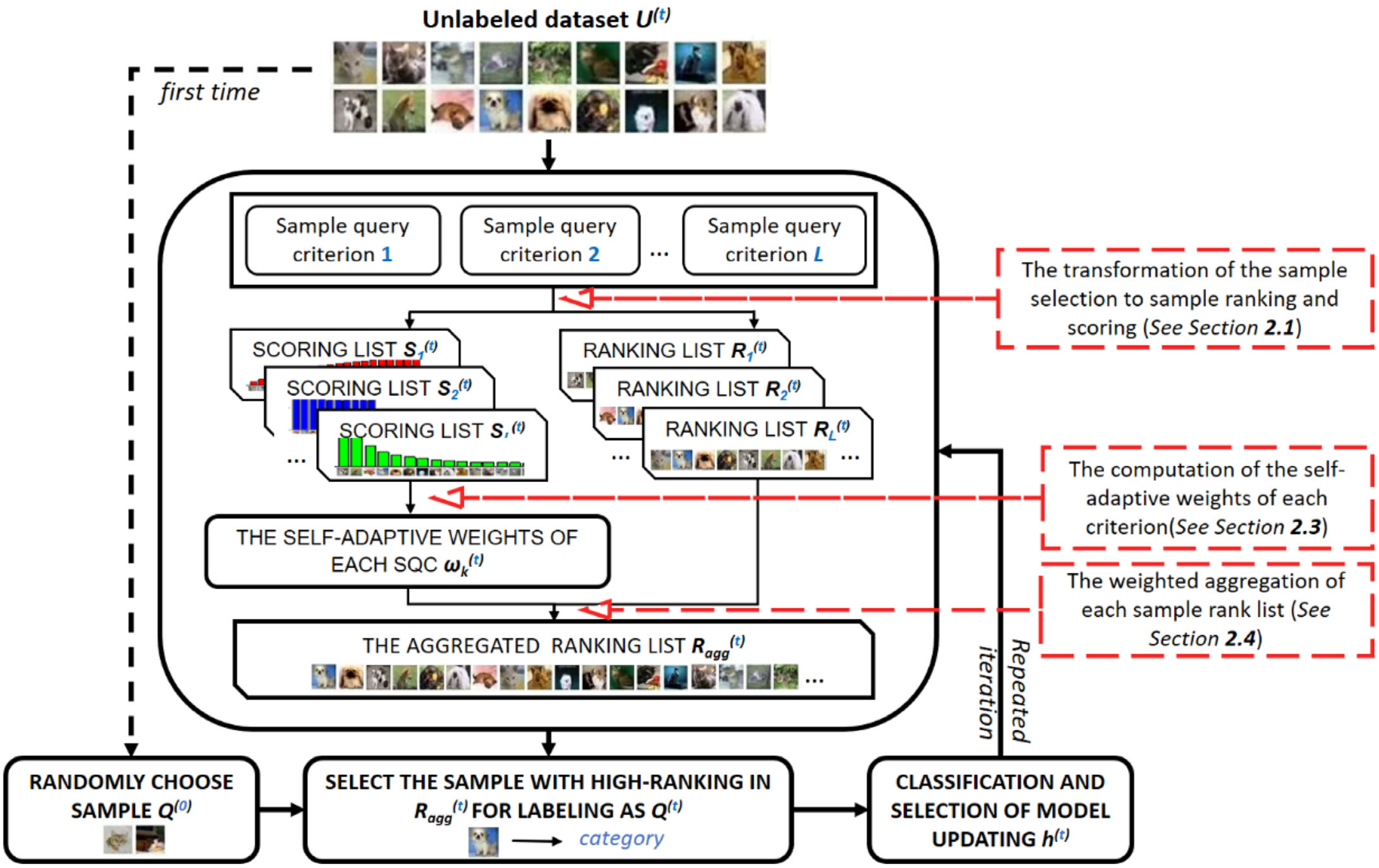

Unlabeled dataset $U^{(t)}$
first time
Sample query criterion 1
Sample query criterion 2
...
Sample query criterion L
The transformation of the sample selection to sample ranking and scoring (See Section 2.1)
SCORING LIST $S_1^{(t)}$
SCORING LIST $S_2^{(t)}$
SCORING LIST $S_l^{(t)}$
...
RANKING LIST $R_1^{(t)}$
RANKING LIST $R_2^{(t)}$
RANKING LIST $R_L^{(t)}$
...
The computation of the self-adaptive weights of each criterion(See Section 2.3)
THE SELF-ADAPTIVE WEIGHTS OF EACH SQC $\omega_k^{(t)}$
The weighted aggregation of each sample rank list (See Section 2.4)
THE AGGREGATED RANKING LIST $R_{agg}^{(t)}$
...
Repeated iteration
RANDOMLY CHOOSE SAMPLE $Q^{(0)}$
SELECT THE SAMPLE WITH HIGH-RANKING IN $R_{agg}^{(t)}$ FOR LABELING AS $Q^{(t)}$
category
CLASSIFICATION AND SELECTION OF MODEL UPDATING $h^{(t)}$

Fig. 4. The process of the proposed MQCAL based on rank aggregation (red boxes indicate the major steps of the proposed method). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

parameters, a certain number of labeled samples are unavoidably required to establish a validation set. Because the proposed method requires no empirical parameters and has high scalability, the users merely have to employ all candidate AL methods for selection as the multiple criteria. RMQCAL can satisfactorily combine them as an optimal ensemble AL method with self-adaptive adjustment of weights. The validation set is no longer needed, which can better serve the needs presented by practical problems, especially when the labeling cost of each sample is very expensive.

The highlights of this work include the following. (1) To the best of our knowledge, this is the first study to analyze and induct the existing MQCAL method with different integration criteria strategies. (2) This is also the first work to implement the MQCAL method by introducing weighted rank aggregation approaches, and the proposed framework may inspire future AL. (3) We present a mechanism that allows for a dynamic and self-adaptive tradeoff between any number and kind of involved SQC in a unified system by introducing the BVSB strategy. (4) We summarize basic rules for the use of our RMQCAL. The potentially best combination of involved SQCs and rank aggregation approaches is also found from experimental comparative results. (5) Several comparative experiments are conducted to prove the effectiveness of the proposed RMQCAL method in many public datasets.

The remainder of this paper is organized as follows. In Section 2, the framework of our RMQCAL is presented, and the three main steps of this framework are discussed in detail. Section 3 describes the experiments that were conducted to evaluate the performance of our RMQCAL and to define optimal operation parameters. In turn, the optimum combination of SQCs and best methods for rank aggregation can be obtained. Finally, our conclusions are presented in Section 4.

2. Approach

2.1. Problem definition

Assume that there is an initial dataset $\boldsymbol{D}$ that is used to train a binary classification model with a lower labeling cost. In any iteration of AL process t, the entire dataset $\boldsymbol{D}$ is always divided into two subsets: the subset $\boldsymbol{A}^{(t-1)}$ and $\boldsymbol{U}^{(t)}$. $\boldsymbol{U}^{(t)}$ is the currently unlabeled data-set, which stores $|\boldsymbol{U}^{(t)}|$ unlabeled samples $u_n^{(t)}$ in the form of feature vectors, where $n \in [1, \cdots, |\boldsymbol{U}^{(t)}|]$, and $|.|$ is a function that is used to calculate the length of an array. In addition, the existing labeled dataset is defined as $\boldsymbol{A}^{(t-1)}$ and is obtained from the previous iteration, which also stores the feature vector of labeled samples as $[a_m^{(t)},\ y_m]$, where $m \in [1, \cdots, |\boldsymbol{A}^{(t-1)}|]$ and $y_m = \{1,\ -1\}$. Through one specialized SQC $\mathrm{F}^{(t)}(.)$ from the old learning model $h^{(t-1)}$ that was previously trained, a conventional single criterion-based AL method selects several of the most important samples $\boldsymbol{Q}^{(t)}$ with the highest value from $\boldsymbol{U}^{(t)}$ in each iteration. Then, the labeled dataset can be reconstituted and used to train a new $h^{(t)}$ and update the SQC as $\mathrm{F}^{(t+1)}(.)$ for the next iteration $t+1(\boldsymbol{A}^{(t)} = \boldsymbol{A}^{(t-1)} \square\ \boldsymbol{Q}^{(t)})$.

Most of the SQC $\mathrm{F}^{(t)}(.)$ in the conventional AL process can be described as in formula (1):

$$\boldsymbol{Q}_{select}^{(t)} = \mathrm{F}^{(t)}\left(\boldsymbol{U}^{(t)}, N\right) = \underset{\boldsymbol{V} \subset \boldsymbol{U}^{(t)}}{argmin} \sum_{n=1}^{N} f^{(t)}(v_n) \tag{1}$$

where $u_n^{(t)}$ are the elements in $\boldsymbol{V}$ ($v_n \in \boldsymbol{V}$); $f^{(t)}(.)$, which is the kernel function in this SQC $\mathrm{F}^{(t)}(.)$ that is used to calculate the score of every unlabeled sample for sample selection, according to the existing labeled samples $\boldsymbol{A}^{(t-1)}$; N is the number of selections in each iteration of the AL process; $|\boldsymbol{V}| = N$, which is usually set as 1.

Unlike the traditional AL process, the intermediate process of MQCAL involves the use of a combination of L SQC, namely, $\mathrm{F}_k^{(t)}(.)$ $n \in [1, \cdots, L]$. Different $\mathrm{F}_k^{(t)}(.)$ have different kernel functions $f_k^{(t)}(.)$. Only the most comprehensively valuable samples that meet all these SQC are selected for labeling in each loop of iterations. The kernel of MQCAL is used to establish an integration criteria strategy that can combine most of the complementary information of each SQC as $\wedge_L^{k=1}(.)$. With regard to the existing MQCAL, including those that are criteria selection-based, MCDM system-based, parallel-form and serial-form, each of their integration criteria strategy can be calculated as formula (2), formula (3), formula (4), and formula (5), respectively:

$$\mathrm{Q}_{cs}^{(t)} = \wedge_{k=1}^{L}\left(\mathrm{F}_k^{(t)}\left(\boldsymbol{U}^{(t)}, N\right)\right) = argmin_{\boldsymbol{V} \subset \boldsymbol{U}^{(t)}} \sum_{n=1}^{N} f_{k^*}^{(t)}(v_n) \tag{2}$$

where $k^* = argmax(CSP_k^{(t)})_{k \in [1,\ \dots,\ L]}$, $CSP_k^{(t)}$ is the criteria selection parameter of $\mathrm{F}_k^{(t)}(.)$ in tth iteration.

$$\mathrm{Q}_{MCDM}^{(t)} = \wedge_{k=1}^{L}\left(\mathrm{F}_k^{(t)}\left(\boldsymbol{U}^{(t)}, N\right)\right) = argmin_{\boldsymbol{V} \subset \boldsymbol{U}^{(t)}} \sum_{n=1}^{N} info^{(t)}(v_n) \tag{3}$$

where $info^{(t)}$ represents the difference between the dominated index and the dominating index of each sample calculated by the MCDM system and $\mathrm{F}_k^{(t)}(.)$; where w_k is the weight parameter of $\mathrm{F}^{(t)}{}_k$. Weight parameters are always fixed empirically or through cross-validation.

$$Q_{parallel}^{(t)} = \wedge_{k=1}^{L}\left(\mathrm{F}_k^{(t)}\left(\boldsymbol{U}^{(t)}, N\right)\right) = argmin_{\boldsymbol{V} \subset \boldsymbol{U}^{(t)}} \sum_{n=1}^{N} \sum_{k=1}^{L} w_k f_k^{(t)}(v_n) \tag{4}$$

$$\begin{aligned} Q_{serial}^{(t)} &= \wedge_{k=1}^{L}\left(\mathrm{F}_k^{(t)}\left(\boldsymbol{U}^{(t)}, N\right)\right) \\ &= \mathrm{F}_L^{(t)}\left(\left(\cdots\left(\mathrm{F}_2^{(t)}\left(\mathrm{F}_1^{(t)}\left(\boldsymbol{U}^{(t)}, N_1\right)\right), N_2\ \right)\cdots\right), N_L\right) \end{aligned} \tag{5}$$

where N_k is the number of selections in layer k, and $N_L = N$.

Both their advantages and disadvantages are mentioned in the previous section.

We note that for each SQC $\mathrm{F}_k^{(t)}(.)$, the corresponding scoring list $\boldsymbol{S}_k^{(t)}$ of the currently unlabeled dataset $\boldsymbol{U}^{(t)}$ can be calculated by the corresponding kernel function $f_k^{(t)}(.)$, as given by formula (6):

$$\boldsymbol{S}_k^{(t)} = \left[f_k^{(t)}\left(u_1^{(t)}\right), \dots, f_k^{(t)}\left(u_n^{(t)}\right), \dots, f_k^{(t)}\left(u_{|U^{(t)}|}^{(t)}\right)\right] \tag{6}$$

Meanwhile, the ranking list $\boldsymbol{R}_k^{(t)}$ of each sample in $\boldsymbol{U}^{(t)}$ can also be easily obtained by sorting $\boldsymbol{S}_k^{(t)}$ in ascending or descending order. Then, we suggest that the integration criteria strategy of our RMQCAL can be designed as formula (7),

$$\mathrm{Q}_{RMQCAL}^{(t)} == \mathrm{F}^{(t)}\left(\boldsymbol{U}^{(t)}, N\right) = argmin_{\boldsymbol{V} \subset \boldsymbol{U}^{(t)}} \sum_{n=1}^{N} \boldsymbol{R}_{agg}^{(t)}\left(v_n^{(t)}\right) \tag{7}$$

where $\boldsymbol{R}_{agg}^{(t)}$ is the aggregated ranking list that satisfies formula (8); K is the calculation of Kendall's tau or Spearman's footrule distance [33]; and w_k is the self-adaptive tradeoff of $\boldsymbol{R}_k^{(t)}$, which is calculated by $\boldsymbol{S}_k^{(t)}0$

$$\boldsymbol{R}_{agg}^{(t)} = argmin_{\boldsymbol{R}} \frac{1}{L} \sum_{k=1}^{L} w_k K\left(\boldsymbol{R}, \boldsymbol{R}_k^{(t)}\right) \tag{8}$$

Then, the problem of our RMQCAL can be transformed as a weighted rank aggregation problem. In other words, three core contents of RMQCAL include the acquisition of $\boldsymbol{S}_k^{(t)}$ and $\boldsymbol{R}_k^{(t)}$, the weighted computation of each criterion $\boldsymbol{\omega_k}$ and the mechanism to effectively combine each SQC. These will be individually discussed in the following three parts.

2.2. The transformation of the sample selection to sample ranking and scoring

The purpose of this step is to obtain L different pairs of rank and score lists $\boldsymbol{S}_k^{(t)}$ and $\boldsymbol{R}_k^{(t)}$ of all remaining unlabeled samples in $\boldsymbol{U}^{(t)}$ from the L different SQCs. Because all SQCs can be described as shown in formula (1), the scoring lists of the currently unlabeled dataset $\boldsymbol{S}_k^{(t)}$ can be further denoted as formula (6), and the ranking lists $\boldsymbol{R}_k^{(t)}$ are the ranking of their corresponding $\boldsymbol{S}_k^{(t)}$from small to large.

As motivating examples to convey the important components of our RMQCAL and the control methods used in this paper, the separate scoring kernel functions $f_k^{(t)}(.)$ of SQCs of some typical AL methods are written as the following: formula (9), formula (10), formula (11) and formula (12).

Margin-based SQC: [16]

$$f^{(t)}_{margin}(x) = P\left(y^*_{\max}|x, h^{(t-1)}\right) \tag{9}$$

Diversity-based SQC: [11]

$$f^{(t)}_{diversity}(x) = \max\left(\cos^{-1}\left(\frac{K\left(x, a_m^{(t-1)}\right)}{\sqrt{K(x,x)K\left(a_m^{(t-1)}, a_m^{(t-1)}\right)}}\right)\right) \tag{10}$$

QBC-based SQC: [19]

$$f^{(t)}_{QBC}(x) = -1\times\sigma\left(h_1^{(t-1)}(x),\ \ldots, h_g^{(t-1)}(x)\right) \tag{11}$$

TED-based SQC: [13]

$$f^{(t)}_{TED}(x) = -1\times\sum_{i=1}^{|\boldsymbol{U}(t)|}\boldsymbol{Z}(:,i),\ i \text{ is the position of } x \text{ in } \boldsymbol{U}^{(t)} \tag{12}$$

where y_{max} is the most likely label of x, κ is the kernel distance, g is the number of committees in QBC, $Z = \min_{\boldsymbol{Z}}\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t)}\boldsymbol{Z}_{2,1} + \lambda\boldsymbol{Z}^T{}_{2,1}$, $s.t.\boldsymbol{Z} = [z_1, \ldots, z_{|\boldsymbol{U}^{(t)}|}] \in R^{|\boldsymbol{U}^{(t)}|\times|\boldsymbol{U}^{(t)}|}$ and σ is the function used to calculate the standard deviation.

Because each SQC has a different AL concept, each $\boldsymbol{S}_k^{(t)}$ must be normalized from −1.0 to 1.0 and sorted from smallest to largest as $\boldsymbol{S}_k^{*(t)}$. The following three points are worth mentioning. (1): In our RMQCAL, we force the definition that the score of the sample with the highest value is the lowest in the scoring list. An SQC that does not conform to the above definition should be revised by multiplying it by −1. (2) When experimental design-based criteria are included in the involved SQCs, their corresponding rank and score lists need to be calculated only once before the first iteration as $\boldsymbol{R}_k^{(0)}$ and $\boldsymbol{S}_k^{*(0)}$ because the experimental design-based SQC does not involve model updating, and its score list is constantly changing in subsequent iterations. (3) $\boldsymbol{R}_k^{(t)}$ from the committee-based SQC includes numerous duplicate values; thus, tie conditions applicable for obtaining $\boldsymbol{R}_k^{(t)}$ should be reflected rather than assigned random rankings.

2.3. The computation of the self-adaptive weights of each criterion

Based on the timeliness of the AL noted above, the contributions of each criterion change in response to different stages of the AL process. The subjective definition of each kind of SQC as its weight, which follows from the old pattern of the serial-designed MQCAL, is not used. In our method, a dynamic weighting system is established for calculating the self-adaptive weights of each SQC for every iteration.

We suggest that the rank list $\boldsymbol{R}_k^{(t)}$ from each involved SQC has its reliability and that the reliability of $\boldsymbol{R}_k^{(t)}$relates not only the design of its corresponding SQC but also the current AL iteration number. The purpose of the weighting system is to ensure the algorithm pays more attention to the SQC whose rank lists $\boldsymbol{R}_k^{(t)}$ are more reliable for the following rank aggregation by assigning them greater weights.

Motivated by the work in [34] for an image retrieval task, we believe that the reliability of $\boldsymbol{R}_k^{(t)}$ from each involved SQC can also be calculated from the distribution of its corresponding score lists $\boldsymbol{S}_k^{*(t)}$. Considering the ultimate goal of every SQC is for sample selection, the score lists $\boldsymbol{S}_k^{*(t)}$ from the SQC with the most reliable $\boldsymbol{R}_k^{(t)}$ should satisfy formula (13) and appear as the red bars shown in Fig. 5(a), since the top-N sample selection based on such $\boldsymbol{S}_k^{*(t)}$ is unique and has strong anti-interference (here we assume that the number of sample selections is $N = 3$).

$$\boldsymbol{S}^{*(t)}_{best}(index) = \begin{cases}\min\left(\boldsymbol{S}^{*(t)}_{best}\right), & if\ index \le N\\ \max\left(\boldsymbol{S}^{*(t)}_{best}\right), & otherwise\end{cases},\ index = 1, 2, \ldots, |\boldsymbol{U}^{(t)}| \tag{13}$$

where $\boldsymbol{S}_k^{*(t)}(index)$ is the score of the sample corresponding to *index* in $\boldsymbol{S}_k^{*(t)}$.

For the opposite case, the worst $\boldsymbol{S}_k^{*(t)}$ is a straight line represented by the purple bar shown in Fig. 5(a), wherein all values

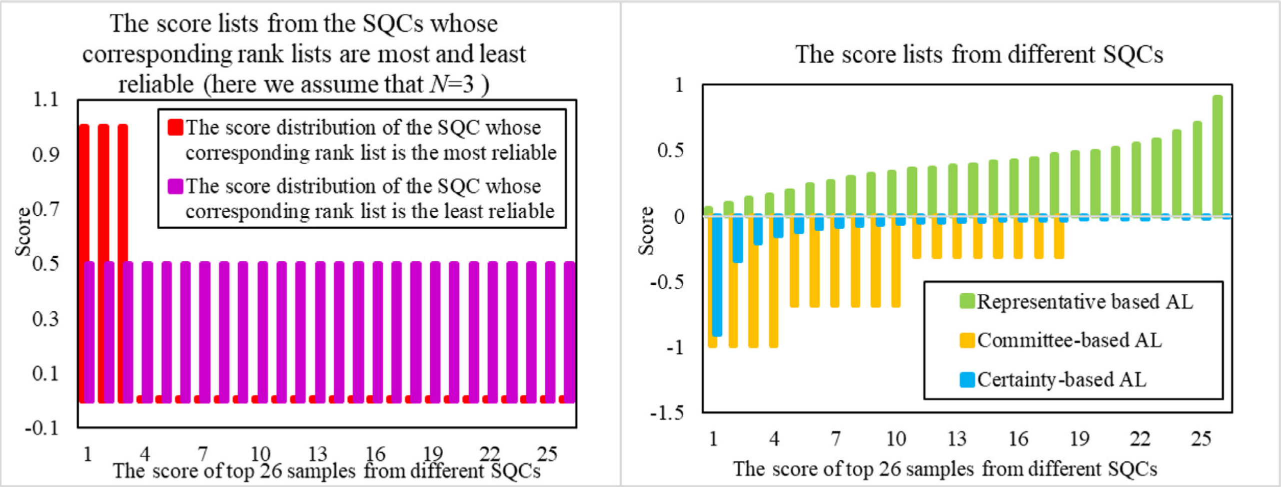


Fig. 5. The score lists from several SQCs (intercepting only the scores of the top 26 most valuable samples defined by three AL methods). The figure on the left is the score lists from the SQCs whose rank lists are most and least reliable and the figure on the right is the score list from the real SQCs. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

of $\boldsymbol{S}_k^{*(t)}$ are the same, and the top-N sample selection is completely uncertain. For practical issues, the real curves of $\boldsymbol{S}_k^{*(t)}$ are typically displayed as shown in Fig. 5(b), and the $\boldsymbol{S}_k^{*(t)}$ values of certainty and representativeness measure-based AL are curvilinear and contain fewer duplicate score values. In contrast, the $\boldsymbol{S}_k^{*(t)}$ of a committee-based AL is ladder-like and has several duplicate score values.

Exploiting a similar trick for the BVSB algorithm and the three curves of the different score lists described above, we suggest that the more obvious the difference in one of the $\boldsymbol{S}_k^{*(t)}$ between the selected samples and the almost-selected samples is, the greater the contribution of its corresponding SQC. For this, we present a weight-assignment method, as illustrated in formula (14), if none of the involved SQCs is committee based.

$$w1_k^{(t)} = (\boldsymbol{S}_k^{*(t)}(N) - \boldsymbol{S}_k^{*(t)}(N+1))/(\boldsymbol{S}_k^{*(t)}(1) - \boldsymbol{S}_k^{*(t)}(|\boldsymbol{U}^{(t)}|)),\quad k \in \text{not committee} \tag{14}$$

For the score list of a committee-based SQC, in which the shapes of score lists are entirely different from those in the two other cases, the above formula (14) cannot be applied here because its unique distribution ($\boldsymbol{S}_k^{*(t)}(N)$ is likely to be equal to $\boldsymbol{S}_k^{*(t)}(N+1)$). Then, we believe that the weights between each committee-based SQC can be calculated using formula (15), as shown below, instead. This means that the SQCs with lower likelihoods of the selected and almost-selected samples sharing the same score should be assigned a higher weight. I(.) is an indicator function that is equal to one if conditions within the parentheses are satisfied; otherwise, it is equal to zero.

$$w2_k^{(t)} = \sum_{index=N+1}^{|\boldsymbol{U}^{(t)}|} \frac{I\left(\boldsymbol{S}_k^{*(t)}(index) \neq \boldsymbol{S}_k^{*(t)}(N)\right)}{|\boldsymbol{U}^{(t)}|},\quad k \in \text{committee} \tag{15}$$

However, we have not developed a more generalized weight-assignment method that applies to representativeness-, certainty- and committee-based SQCs. When the involved SQCs in MQCAL include all three of the types listed above, the only workable revised weight assignment scheme is written as formula (16): where $c2$ is the number of committee-based SQC, and $c1 = L - c2$.

$$w_k^{(t)} = \begin{cases} \frac{c1}{L} * w1_k^{(t)} / \sum_j w1_j^{(t)}, & k,\ j \in \ not\ committee \\ \frac{c2}{L} * w2_k^{(t)} / \sum_j w2_j^{(t)}, & k,\ j \in \ committee \end{cases} \tag{16}$$

Algorithm 1 Weight calculations of the RMQCAL process.

```
Input: The $L$ score lists $\boldsymbol{S}_k^{*(t)}$ from the $n_1$ certainty-based SQCs, $n_2$
    committee-based SQCs and $n_3$ representativeness-measure SQCs, where
    $k = 1, 2, \cdots, L (L = n_1 + n_2 + n_3)$, and the number of samples is selected
    from each iteration $N$.
1: Normalize each score list to $-1.0$ to 0 or 0 to 1.0; then, $\boldsymbol{S}_k^{*(t)}$ can be
    obtained by sorting the scores in ascending order.
2: Calculate two correction parameters: $c2 = n_2$, and $c1 = n_1 + n_3$.
3: Calculate $w_k^{(t)}$ from formulas (14), (15) and (16).
Output: A vector $\boldsymbol{w}^{(t)} = [w_1^{(t)}, \cdots, w_L^{(t)}]$ that represents the weight of each $L$
    SQC in the $t$ th iteration.
```

2.4. *The weighted aggregation of each sample rank list*

After obtaining $\boldsymbol{R}_k^{(t)}$ from step 1 and $\boldsymbol{w}^{(t)}$ from step 2, the following problem is similar to a rank aggregation problem that can be elegantly solved by using improved rank aggregation methods.

Here, it is useful to review rank aggregation methods. Lin S summarized existing rank aggregation methods that had been developed up until 2010 [33] and are used to address problems related to recommendation systems. In recent years, many methods of rank aggregation have been designed, including the following: Borda's method, Bucklin voting [35], the Markov chain [36], Thurstone's model, the cross-entropy Monte Carlo model [33], the Condorcet method [37] and other stochastic methods [38]. Rank aggregation is now widely employed to address information retrieval problems. To our knowledge, this is the first study to apply rank aggregation methods to the AL problem.

However, some differences remain between common rank aggregation problems and our MQCAL problem, and some of the rank aggregation methods may not properly address MQCAL problems. Therefore, before introducing these methods into our algorithm, they must still be selected and improved. Put simply, specific differences include the following: (1) the number of rank lists (L) is not sufficiently large to establish a statistical model; (2) the number of elements in $\boldsymbol{R}_k^{(t)}$ is large, particularly for the first iterations (when t is small), which can result in inefficiencies; (3) traditional rank aggregation problems seldom involve weighting; and (4) in most cases, $\boldsymbol{R}_k^{(t)}$ is not a shuffled list from one to $|\boldsymbol{U}^{(t)}|$ as the same rankings may be involved.

Point (1) implies that statistical model-based rank aggregation methods, e.g., Thurstone's model, cannot work. With regard to point (2), we apply rank aggregation methods for lower computing complexity, e.g., Borda's and Bucklin's methods and the Markov chain. In addition, a sample truncation method is proposed as a means to further reduce the number of samples involved in rank aggregation. Regarding points (3) and (4), some improvements are made to existing rank aggregation methods (e.g., adding weights to each list).

For the above problems, we present three feasible means of rank aggregation of varying computing complexity and performance that are based on enhanced versions of the Borda, Bucklin voting and Markov chain approaches.

2.4.1. *Borda's methods*

Borda's methods are the most popular and intuitive rank aggregation methods [33], and they are still widely used to study elections. There are two main phases of Borda methods.

1. The first phase involves the construction of a mapping function MAP(.) between the ranking $\boldsymbol{R}_k^{(t)}$ and its corresponding Borda score $\boldsymbol{B}_k^{(t)}$. When addressing practical issues, MAP(.) is typically designed to score as 1 when ranked first, as 2 when ranked second, and so on. In the other words, MAP(.) is expressed as formula (17):

$$\boldsymbol{B}_k^{(t)}(index) = MAP\left(\boldsymbol{R}_k^{(t)}(index)\right) \approx \boldsymbol{R}_k^{(t)}(index),\ \ index = 1, 2, \ldots, \left|\boldsymbol{U}^{(t)}\right| \tag{17}$$

where $\boldsymbol{R}_k^{(t)}(index)$ is the score of indexed samples in $\boldsymbol{R}_k^{(t)}$.

For the application of such methods to our RMQCAL, due to the processing that is involved in weighting, formula (17) should be reformulated as formula (18):

$$\boldsymbol{B}_k^{(t)}(index) = MAP\left(\boldsymbol{R}_k^{(t)}(index)\right) \cdot w_k^{(t)} \approx \boldsymbol{R}_k^{(t)}(index) \cdot w_k^{(t)},\ \ index = 1, 2, \ldots, \left|\boldsymbol{U}^{(t)}\right| \tag{18}$$

2. The second phase involves the use of f_{borda}(.), which we refer to as the Borda score fusion algorithm. This algorithm is used to obtain the overall Borda score $\boldsymbol{B}_{borda}^{(t)}$ by combining all Borda scores in formula (19):

$$\boldsymbol{B}_{borda}^{(t)} = f_{borda}\left(\boldsymbol{B}_1^{(t)}, \boldsymbol{B}_2^{(t)}, \ldots,\ \boldsymbol{B}_L^{(t)}\right) = \begin{cases} min\left(\boldsymbol{B}_1^{(t)}, \boldsymbol{B}_2^{(t)}, \ldots,\ \boldsymbol{B}_L^{(t)}\right) & (minimum) \\ median\left(\boldsymbol{B}_1^{(t)}, \boldsymbol{B}_2^{(t)}, \ldots,\ \boldsymbol{B}_L^{(t)}\right) & (median) \\ \left(\prod_{k=1}^{L} \boldsymbol{B}_k^{(t)}\right)^{1/L} & (geometric\ mean) \\ \sum_{k=1}^{L} \left(\boldsymbol{B}_k^{(t)}\right)^p & (p - norm) \end{cases} \tag{19}$$

where p is typically set equal to 1 (Here, the p-norm algorithm is the arithmetic mean).

The Borda-based rank aggregation method of our RMQCAL involves the following four steps:

Algorithm 2 Borda-based rank aggregation of the RMQCAL process.

```
Input: The L rank lists $\boldsymbol{R}_k^{(t)}$ from corresponding SQC values, where $k = 1, 2, \ldots,$
   L, and the number of samples is selected in each iteration $N$ and $\boldsymbol{w}^{(t)}$.
1: Determine the mapping function MAP(.), the core fusion algorithm $\mathrm{f}_{borda}(.)$,
   and parameter $N$
2: An $L \times |\boldsymbol{U}^{(t)}|$ Borda score list can be established according to the above
   formula (18), in which every row is the Borda score of one rank list, and
   every column includes the $L$ Borda score of one sample from different SQC
   values.
3: The overall Borda score can be obtained from formula (19). For each column
   of the above list, $\boldsymbol{R}_{agg}^{(t)}$ is the rank list of this Borda score from small to large.
4: $N$ samples with the lowest overall Borda score $\boldsymbol{B}_{borda}^{(t)}$ can be selected as $\boldsymbol{Q}^{(t)}$.
Output: $\boldsymbol{Q}^{(t)}$, denoting the most $N$ valuable samples, is selected from $\boldsymbol{U}^{(t)}$ in the
   $t$-th iteration.
```

2.4.2. Bucklin voting method

According to the method described in [35], the earliest iteration of the Bucklin voting method was also used in voting systems for candidate selection. According to the kernel principle of the Bucklin voting method, when one candidate has a majority, that candidate wins. Otherwise, the second choice is added to the first choice. Whether one candidate has a majority is re-estimated; if so, that candidate wins. If not, the previous tasks are repeated.

Due to the importance of weighting factors, the algorithm is similar to the Electoral College system. Election candidates are used as samples. The first and second choices correspond, respectively, to the first and second ranked $\boldsymbol{R}_k^{(t)}$ values. L SQCs are no longer L voters but are L states, and $\boldsymbol{R}_k^{(t)}$ can be regarded as the number of electoral votes cast in each state. Based on a previous publication [35], the overall process is described as follows.

Algorithm 3 Bucklin voting-based rank aggregation of the RMQCAL process.

```
Input: $L$ rank lists $\boldsymbol{R}_k^{(t)}$ from corresponding SQCs, where $k = 1, 2, \ldots, L$, and the
   number of samples is selected in each iteration $N$ and $\boldsymbol{w}^{(t)}$.
1: Establish a $1 \times |\boldsymbol{U}^{(t)}|$ sparse list $\boldsymbol{SL}$, where each column records the electoral
   votes of each sample from $L$ rank lists initialized to zeros.
2: Set $ch = 1$ and $ii = 1$ and construct an empty $\boldsymbol{Q}^{(t)}$.
      3: Start searching the sample with the $ii$th value in the aggregate rank list;
   the positioning of this sample is saved in $\boldsymbol{Q}^{(t)}$.
While $(ii < 1 + N)$
      4: For $j = 1: L$
         If the $index$th sample is the $ch$th of $\boldsymbol{R}_k^{(t)}$, $\boldsymbol{R}_k^{(t)}(index) = ch$
            Update the list $\boldsymbol{SL}$, $\boldsymbol{SL}\ (ch,\ index) = \boldsymbol{SL}\ (ch,\ index) + w_k^{(t)}$.
         End
      End
      If there is a column $index^*$ of list $\boldsymbol{SL}$ whose summation is greater than 0.5
         5: Record $index^*$ in $\boldsymbol{Q}^{(t)}$, $ii = ii + 1$; clear this column to zero to prevent it
   from being selected a second time.
      Else
         5: $ch = ch + 1$; insert a new line below $\boldsymbol{SL}$.
      End
End
Output: $\boldsymbol{Q}^{(t)}$, which are the $N$ samples with the highest value selected from
   $\boldsymbol{Q}^{(t)}$, of the $t$th iteration.
```

Theoretically, the Bucklin voting method is an ideal method to use in our RMQCAL because it makes no attempt to initially aggregate a complete rank list $\boldsymbol{R}_{agg}^{(t)}$ for all involved samples. The samples of each place in the rank list after aggregation are confirmed individually and are exactly what our RMQCAL requires (only the most N valuable samples need be selected for each iteration, where N is always small).

2.4.3. Markov chain method

The Markov chain method was first introduced into the PageRank algorithm (a practical rank aggregation problem) by Dwork in 2001. As noted in the literature [36], the Markov chain serves as an elegant, rational and high-performance solution to rank aggregation problems. The central ideas of the conventional Markov chain that are used for rank aggregation typically involve two steps. Step 1: Convert aggregated targets by incorporating several input ranking lists into a specific transition matrix using one form of probability assignment P(.). Step 2: According to [36], regardless of the initial state, the Markov chain system based on one specific transition matrix will always eventually reach a unique fixed point at which the state distribution does not change. We define this point as the stationary distribution of the corresponding transition matrix, which is also the basis for ranking lists after aggregation. According to the different transition matrices, the Markov chain method can be subdivided into the following: MC1, MC2 and MC3 [33].

However, traditional Markov chain methods are not completely suited to address our RMQCAL problem because the original methods do not apply weights. In addition, its computation complexity is high, particularly when t is small. To solve these two problems, two changes are made to the Markov chain method in the proposed RMQCAL approach, as follows.

First, before building a transition matrix, an extra ‘sample truncation’ step is added to significantly reduce computation complexity levels using only some of the samples. Only the samples that are among the top N^* in each $\boldsymbol{R}_k^{(t)}$ at least once will join the next phase of transition matrix establishment, and the remainder are ignored. The sample truncation process is expressed as formula (20):

$$\boldsymbol{R}_k^{*(t)} = \boldsymbol{R}_k^{(t)}(n),\ \ n \in \left\{ i| \sum_{k=1}^{L} I\left(\boldsymbol{R}_k^{(t)}(i) \le N^*\right) \ge 1, i = 1, \ldots, \left|\boldsymbol{U}^{(t)}\right| \right\} \tag{20}$$

where $\boldsymbol{R}_k^{*(t)}$ is a modified version of $\boldsymbol{R}_k^{(t)}$ after sample truncation, N* is typically set as $N^* = N + tun_2$. (tun_2 can be set equal to 5), and $N\langle\ N^* \le| \boldsymbol{R}_k^{*(t)}| \ll | \boldsymbol{R}_k^{(t)}|=\ |\boldsymbol{U}^{(t)}|$.

This improvement is applicable to our RMQCAL problem. For our RMQCAL, in each iteration, the samples that are used are ranked at the top of the N list after rank aggregation. Furthermore, the higher a sample ranks in any $\boldsymbol{R}_k^{(t)}$, the more likely it is to occupy the top N place in the aggregated list of all $\boldsymbol{R}_k^{(t)}$ values. Therefore, the rank aggregated results of each $\boldsymbol{R}_k^{(t)}$ and the front section of each $\boldsymbol{R}_k^{(t)}$ are likely to be the same, particularly when N is not large and when most $\boldsymbol{R}_k^{(t)}$ values are relatively similar.

Second, for each pair of samples $u_i^{(t)}$, $u_j^{(t)}$ ($i \ne j$ and i, $j\ \in [1,\ 2,\ \cdots,\ |\ \boldsymbol{R}_k^{*(t)}|]$), the improved weighted transition probability $\boldsymbol{Tran}^{(t)}(i, j)$ in RMQCAL can be described as formula (21):

$$\begin{aligned}\boldsymbol{Tran}^{(t)}(i,\ \ j) \\ = \mathrm{P}\left(u_i^{(t)} \to u_j^{(t)}\right) \\ = \begin{cases} \frac{1}{\left|\boldsymbol{U}^{(t)}\right|} \cdot I\left(\sum_{k=1}^{L} w_k^{(t)} \cdot \left(I\left(\boldsymbol{R}_k^{(t)}(i) > \boldsymbol{R}_k^{(t)}(j)\ \right)\right) > 0\right) :\ MC1 \\ \frac{1}{\left|\boldsymbol{U}^{(t)}\right|} \cdot I\left(\sum_{k=1}^{L} w_k^{(t)} \cdot \left(I\left(\boldsymbol{R}_k^{(t)}(i) > \boldsymbol{R}_k^{(t)}(j)\ \right)\right) > \frac{1}{2}\right) :\ MC2 \\ \frac{1}{\left|\boldsymbol{U}^{(t)}\right|} \cdot \sum_{k=1}^{L} w_k^{(t)} \cdot \left(I\left(\boldsymbol{R}_k^{(t)}(i) > \boldsymbol{R}_k^{(t)}(j)\ \right)\right)\ \ :\ MC3 \end{cases}\end{aligned} \tag{21}$$

After all $\mathrm{P}(u_i^{(t)} \to u_j^{(t)})$ values have been calculated, $\mathrm{P}(u_i^{(t)} \to u_i^{(t)})$ can be obtained from formula (22):

$$\boldsymbol{Tran}^{(t)}(i, i) = \left(u_i^{(t)} \to u_i^{(t)}\right) = 1 - \sum_{i \ne j} \boldsymbol{Tran}^{(t)}(i, j) \tag{22}$$

Because L is typically not large in our RMQCAL problem, the above $\boldsymbol{Tran}^{(t)}$ is often a large sparse matrix with several 0 elements. To ensure ergodic results for the transition matrix, a tuning parameter t is introduced and treated as follows in formula (23):

$$\boldsymbol{Tran}^{*(t)}(i,\ j) = \boldsymbol{Tran}^{(t)}(i,\ j) \times (1 - tun_1) + \frac{tun_1}{|\boldsymbol{U}^{(t)}|} \tag{23}$$

where tun_1 is typically set to range from 0.01 to 0.15, as specified by the above reference.

Finally, from the perspective of matrix theory, the stationary distribution of one transition matrix is its principal left eigenvector, which can be computed from a regular power-iteration algorithm after transposing the above matrix. The improved MC method used for the RMQCAL problem is as follows:

Algorithm 4 Markov chain-based rank aggregation for the RMQCAL process.

```
Input: $L$ rank lists $\boldsymbol{R}_k^{(t)}$ and corresponding $\boldsymbol{w}^{(t)}$ values from the corresponding
    SQCs, where $k = 1, 2, ..., L$, the number of samples selected in each iteration
    is designated $N$, the tuning parameter is designated $tun_1$, and the number of
    other samples of interest is designated $tun_2$.
1: For each $\boldsymbol{R}_k^{(t)}$ except for the committee-based one, use formula (20) and
    $tun_2$ to truncate, and reconstruct the term as $\boldsymbol{R}_k^{(t)}$.
2: Preferences among pairs of samples for each $\boldsymbol{R}_k^{(t)}$ are calculated through one
    mode of probability assignment P(.) using formulas (21) and (22).
3: A transition matrix is established and adapted using formula (23) and the
    tuning parameter $tun_1$.
4: The obtained transition matrix must be transposed and then used in a
    regular power iteration algorithm to calculate its left eigenvector (the
    stationary distribution).
5: The value of each element in a stationary distribution can be regarded as a
    Markov chain score of its corresponding samples. The $\boldsymbol{R}_{agg}^{(t)}$ is the rank list of
    the Markov chain score from large to small, and the top $N$ samples with
    high Markov chain scores are collected as $\boldsymbol{Q}^{(t)}$ values to query for labels.
Output: $\boldsymbol{Q}^{(t)}$, which denotes the most $N$ valuable samples, is selected from $\boldsymbol{U}^{(t)}$
    in the $t$th iteration.
```

2.4.4. Comparison of methods

Note that, regarding levels of computational complexity, in the above weighted rank aggregation methods, the algorithm for solving the top-k problem in an unsorted array with n elements is unified, and the computational complexity of these methods is equal to O(n). Then, on average, the computational complexity of Borda's and Bucklin voting are all O($|\boldsymbol{U}^{(t)}| \times L$). Because we employ 'sample truncation', the computational complexity of the Markov Chain can be diminished to O(N^{*3}) from O($|\boldsymbol{U}^{(t)}|^3$). Because the value of N^* is far lower than that of $|\boldsymbol{U}^{(t)}|$, the computational complexity of our improved Markov Chain method for MQCAL will be acceptable.

Borda's method is inferior to the others because in certain special cases, particularly when L is small or when one rank list $\boldsymbol{R}_k^{(t)}$ is committee-based, using Borda's method can cause most unlabeled samples to have the same overall Borda score. In turn, the most valuable samples with the lowest overall Borda scores often cannot be selected. This situation does not occur when the Markov chain and Bucklin voting methods are applied. Considering the corresponding performance, relevant documents indicate that the Markov chain works better than the Borda and Bucklin voting methods when applied to traditional rank aggregation problems. For the RMQCAL problem, the rationality and validity of the above ranking aggregation method still need to be confirmed by multiple experiments, as described in Section 4.

Above all, the improved process of our RMQCAL can be defined as follows:

Algorithm 5 Our RMQCAL process.

```
Input: The $L$ SQC, namely, $\mathrm{F}_k^{(t)}$ ($k = 1$: $L$), the unlabeled dataset $\boldsymbol{U}^{(0)}$, and the
    number of samples selected in each iteration $N$.
Repeat
    If the number of iterations $t = 0$
        Step 0: randomly use one kind of experimental design-based SQC to
    select the first batch of unlabeled samples for labeling as $\boldsymbol{Q}^{(0)}$,
    $\boldsymbol{A}^{(0)} = \boldsymbol{Q}^{(0)}$ and $\boldsymbol{U}^{(1)} = \boldsymbol{U}^{(0)} \backslash \boldsymbol{Q}^{(0)}$
    Else
        Step 1: Obtain each pair of $\boldsymbol{R}_k^{(t)}$ and $\boldsymbol{S}_k^{(t)}$ using $\mathrm{F}_k^{(t)}$ in $\boldsymbol{U}^{(t)}$.
        Step 2: Using Algorithm 1 for $\boldsymbol{S}_k^{(t)}$, obtain the weights of each $\mathrm{F}_k^{(t)}$ in
    the $t$ th iteration $\boldsymbol{S}_k^{(t)} \rightarrow \boldsymbol{S}_k^{*(t)} \rightarrow \boldsymbol{w}_k^{(t)}$
        Step 3: Choose one rank aggregation method (Algorithm 2, Algorithm
    3, or Algorithm 4) to obtain the weighted aggregated rank list and select the
    sample with the top $N$ value as $\boldsymbol{Q}^{(t)}$. Then, $N$, $\boldsymbol{R}_k^{(t)}$, $\boldsymbol{w}_k^{(t)} \rightarrow \boldsymbol{Q}^{(t)}$
        Step 4: Request a label of $\boldsymbol{Q}^{(t)}$ from the Oracle. Then,
    $\boldsymbol{A}^{(t)} = \boldsymbol{A}^{(t-1)} \cup \boldsymbol{Q}^{(t)}$ and $\boldsymbol{U}^{(t+1)} = \boldsymbol{U}^{(t)} \backslash \boldsymbol{Q}^{(t)}$.
    End
Until: a stopping criterion is applied or $|\boldsymbol{U}^{(t)}| = 0$.
```

3. Experiments

3.1. Dataset description

To evaluate the overall performance of our RMQCAL, comparative experiments are conducted on 14 different binary classification problems from the UCI Repository downloaded from the public website http://archive.ics.uci.edu/ml/. Each problem corresponds to one dataset, as shown in Table 1. *Vehicle**, *Isolet**, *Titato**, *Austra**, *LetterEF**, *LetterIJ**, *LetterMN**, *LetterDP**, *LetterUV** and *Wdbc** are 10 common datasets, which are consistent with the datasets provided in [8]. Moreover, the remaining four datasets, *Mushroom*+, *EEG*+, *Mocap*+, and *Epilepsy*+, which contain more than 4000 samples, are used to validate the performance of the proposed method on a large-scale dataset. Notably, the dataset *Wdbc** is also employed to search for the best combination of rank aggregation methods and SQCs in RMQCAL.

Before an experiment is conducted, each dataset is normalized and randomly divided into two parts of equal size. One part is used as a test set, and the other is used as the unlabeled sample for AL methods. To ensure the reliability of the experimental results, most of the experiments listed below are run 10 times, and the average for each period is shown as the final performance result.

3.2. Experimental setting

All operations are executed using MATLAB R2014a software (Mathworks, Inc., Natick, MA, USA) installed on a PC with an Intel Core i3-2100 CPU (3.10 GHz) and 3 GB memory. Because the

Table 1
The experimental datasets used.

	Pos: Neg	Feature number	Sample size
Vehicle*	218:217	18	438
Isolet*	300:300	617	600
Wdbc*	212:357	30	569
Titato*	626:332	9	958
Austra*	307:383	14	690
LetterEF*	768:775	16	1543
LetterIJ*	755:747	16	1502
LetterMN*	792:783	16	1575
LetterDP*	805:803	16	1608
LetterUV*	813:764	16	1577
Mushroom+	4208:3916	22	8124
EEG+	6723:8257	14	14,980
Mocap+	16,265:15,733	15	31,998
Epilepsy+	2300:2300	178	4600

main purpose of AL is to effectively and efficiently establish a good learning model regardless of whether it improves its performance, this paper only applies to an SVM classifier with an RBF kernel—the same as was used in [8]—as the baseline against which comparisons to all approaches can be drawn. The SVM classifier is supported by LibSVM at http://www.csie.ntu.edu.tw/~cjlin/libsvm/. The source code of our RMQCAL and the control method have also been uploaded to GitHub. The interested reader can download the code from https://github.com/wangtaoz/RMQCAL.git.

3.3. Performance metrics

For Experiments A and B**,** two metrics (namely, accuracy and F1-measure) are used to evaluate the performance of approaches relative to those described in [8]. The F1-measure is a common metric described in formula (24):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{24}$$

For Experiment C, in addition to accuracy, the area under the ROC curve (AUC) is added as an additional evaluation metric. In addition, paired t-tests conducted at the 95 percent significance level are introduced to reflect the difference between the two methods. For Experiment A, Kendal's tau [39] and Spearman's footrule distance [40] are also introduced to evaluate rank aggregation effects as described in formula (25):

$$\begin{cases} dist_{Spearman} = \sum_{k=1}^{L} Spear\left(\boldsymbol{R}_{agg}^{(t)}, \boldsymbol{R}_{k}^{(t)}\right) \\ \quad = \sum_{k=1}^{L} \sum_{i=1}^{|\boldsymbol{U}^{(t)}|} \left|\boldsymbol{R}_{agg}^{(t)}(i) - \boldsymbol{R}_{k}^{(t)}(i)\right| \\ dist_{Kendal} = \sum_{k=1}^{L} Kendal\left(\boldsymbol{R}_{agg}^{(t)}, \boldsymbol{R}_{k}^{(t)}\right) \\ \quad = \sum_{k=1}^{L} \sum_{i,j=1}^{|\boldsymbol{U}^{(t)}|} I\left(\left(\boldsymbol{R}_{agg}^{(t)}(i) - \boldsymbol{R}_{agg}^{(t)}(j)\right)\left(\boldsymbol{R}_{k}^{(t)}(i) - \boldsymbol{R}_{k}^{(t)}(j)\right) < 0\right) \end{cases} \tag{25}$$

where $i \neq j \in [1, 2, \ldots, |U^t|]$, and $\boldsymbol{R}_{agg}^{(t)}$ is the rank list after aggregation.

As to the Experiment D, the CPU time is used to measure the efficiency of each contrast algorithms and the proposed RMQCAL.

3.4. Experimental goals

The whole experiment section has two goals: determining which implementation of the method is used in each step of RMQCAL and verifying the performance of RMQCAL.

For the first goal, Experiment A is designed to choose the best rank aggregation method for RMQCAL from the involved candidate methods. In a related aspect, Experiment B needs to find possible RMQCAL rules, including the results within various combinations of involved SQCs and the results for various numbers of involved SQCs. Then, a credible and complete RMQCAL process can be determined and will be used in the following experiments.

The goal of Experiment C is to confirm the effectiveness and superiority of RMQCAL through a series of comparative experiments on several different kinds of control methods. In addition, Experiment D is used to analyze and evaluate the algorithm efficiency of the proposed RMQCAL.

3.5. Experimental process

Experiment A. The selection of the most appropriate rank aggregation methods for RMQCAL

Description of Experiment A:

To preliminarily narrow the selection of rank aggregation methods, in Experiment A, a toy example is presented as an input list with seven ranking lists to first illustrate the performance differences of the candidate rank aggregation methods, which are confirmed as available in the above article including the Borda methods within different Borda score expressions (i.e., median, p-norm, minimum and geometric mean, as described in formula (19)), Bucklin voting, and Markov chain methods within several weighted transition probability expressions (i.e., MC1, MC2, MC3, as described in formula (21)). Moreover, in this and the following experiments, the tuning parameter of the Markov chain method and the Borda parameter are set as 0.05 and 1, respectively.

The candidate rank aggregation methods with the better performance on the above toy example will be used as RMQCAL integration criteria strategies. Then, a complete RMQCAL process is implemented for dataset *Wdbc**. In this phase, the controlled experiment is designed to further evaluate the effects of various rank aggregation methods on our MQCAL process. The SQCs are fixed, and their accuracy levels and F1-measures (X-axis) with different labeling costs (*Y*-axis), which are indicated as the percentage of the sample designated for label selection, are compared in Fig. 6.

Result of Experiment A:

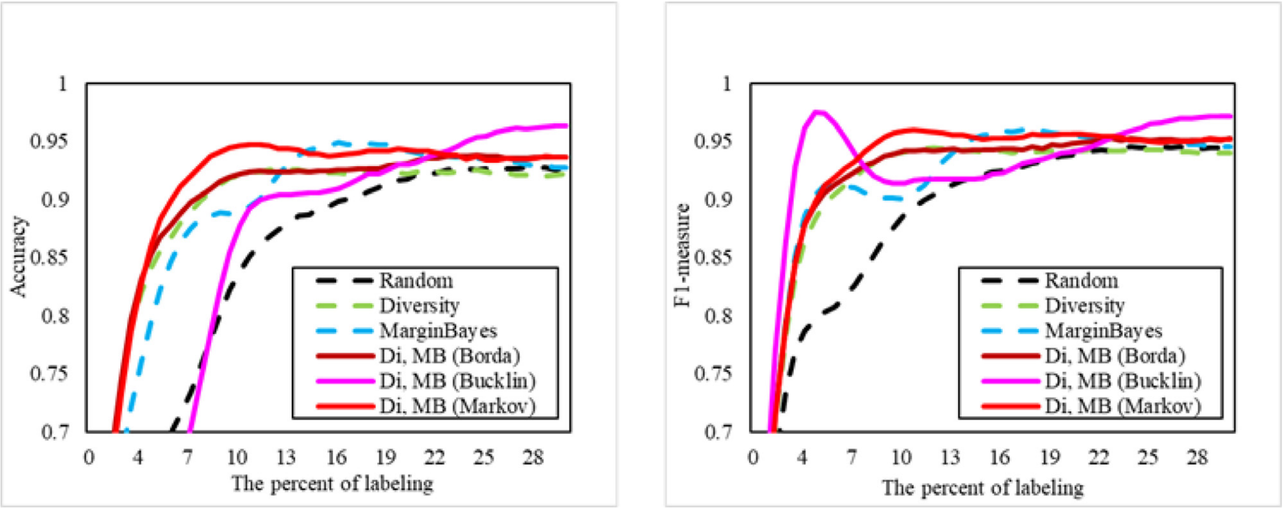


Fig. 6. Performance of RMQCAL realized using the Borda, Bucklin voting and Markov chain methods.

Table 2
Comparison of ranking aggregation methods used in a toy example.

Sample	Input lists							Borda method				Markov chain			
	L1	L2	L3	L4	L5	L6	L7	Min	Med	P-norm	Geo	Buc	MC1	MC2	MC3
Sample1	8	6	3	3	7	4	1	1	2	3	3	4	8	3	3
Sample2	10	8	2	2	9	5	1	6	4	5	4	7	7	4	7
Sample3	2	2	1	1	5	3	1	2	1	1	1	1	1	1	1
Sample4	9	1	4	4	3	2	1	3	3	2	2	3	3	2	2
Sample5	5	7	8	10	10	7	5	9	7	9	9	9	9	8	10
Sample6	7	10	8	8	1	1	5	4	8	7	5	5	5	9	5
Sample7	6	5	8	9	8	10	5	10	10	10	10	10	10	10	9
Sample8	1	9	6	7	2	8	5	5	9	6	6	4	4	6	5
Sample9	4	4	5	6	6	9	5	8	6	8	8	6	6	7	8
Sample10	3	3	7	5	4	6	5	7	5	4	7	2	2	5	4
Kendall's tau distance								93	83	**81**	84	**85**	99	**79**	84
Spearman's footrule distance								178	156	**156**	166	**152**	172	**154**	158

Analysis of Experiment A:

From the results in Table 2, it can be observed that the MC2 method, the Bucklin method and the Borda method with the p-norm (***p*** equal to 1) perform better with Kendall's tau distance and Spearman's distance values of (79,154), (85,152) and (81,156) on above toy example, respectively. Additionally, only these three methods will be introduced in the next step as the three rank aggregation methods used for integration. As the results show in Fig. 6, we find that the enhanced effects of our RMQCAL method based on the Markov chain approach with an MC2 weighted transition probability expression are the most easily detectable; the Borda approach ranks second, and the Bucklin voting method is not satisfactory. This final result is likely attributable to a property of the Bucklin voting method, which often positions samples with the highest median ratings at the top of a rank list after aggregation, an unsuitable property for AL problems. Thus, subsequent experiments use the Markov chain approach with an MC2 weighted transition probability expression.

Experiment B. RMQCAL experiments with various SQC combinations

Description of Experiment B:

After determining the best ranking aggregation method for the proposed RMQCAL method in Experiment A, Experiment B involves a series of comparative experiments that are used to reflect the performance of RMQCAL in dataset *Wdbc** when applying various combinations of SQCs. The candidate SQCs used include the following: Diversity (DI), Margin in RBF-SVM (MR), Margin in Bayes (MB), QBC and TED. The performance curves of each case, including their accuracies and F1-measures (X-axis) with different labeling costs (Y-axis), are presented in Figs. 7–15.

Result of Experiment B:

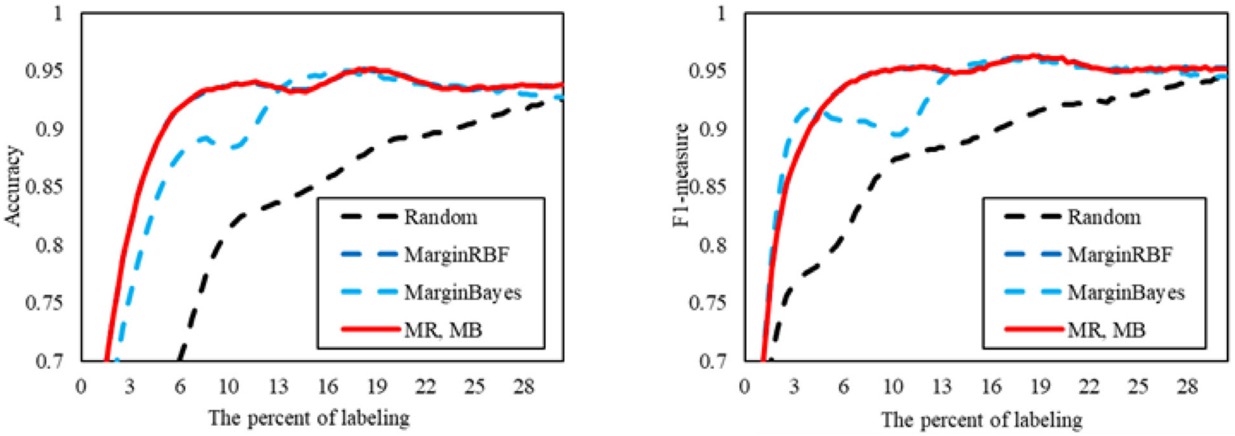


Fig. 7. Performance of RMQCAL with two certainty-based SQCs.

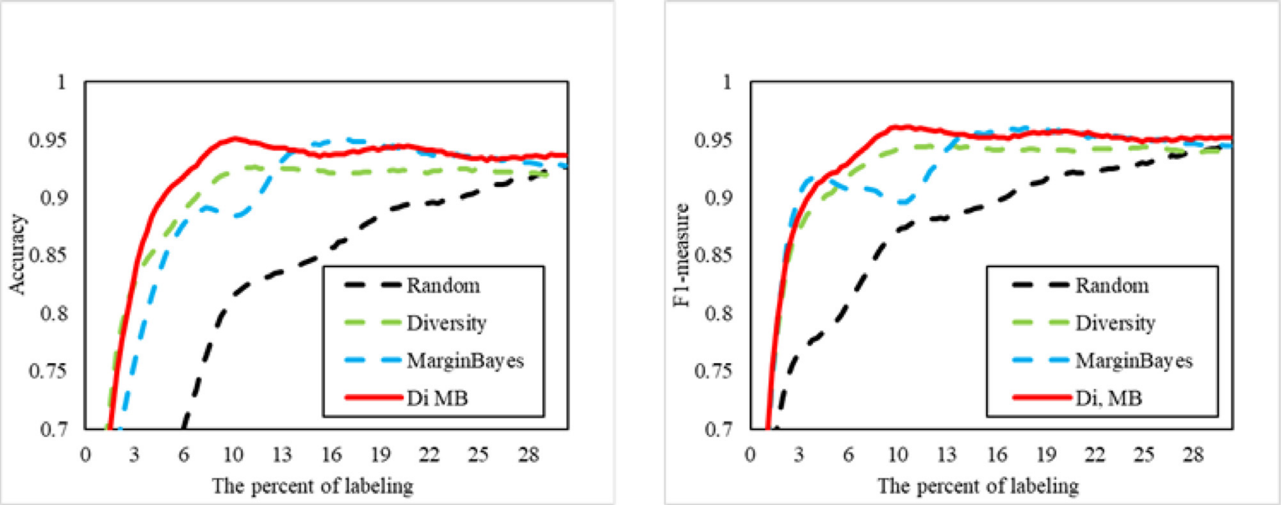


Fig. 8. Performance of RMQCAL with certainty-based and connection-based SQCs.

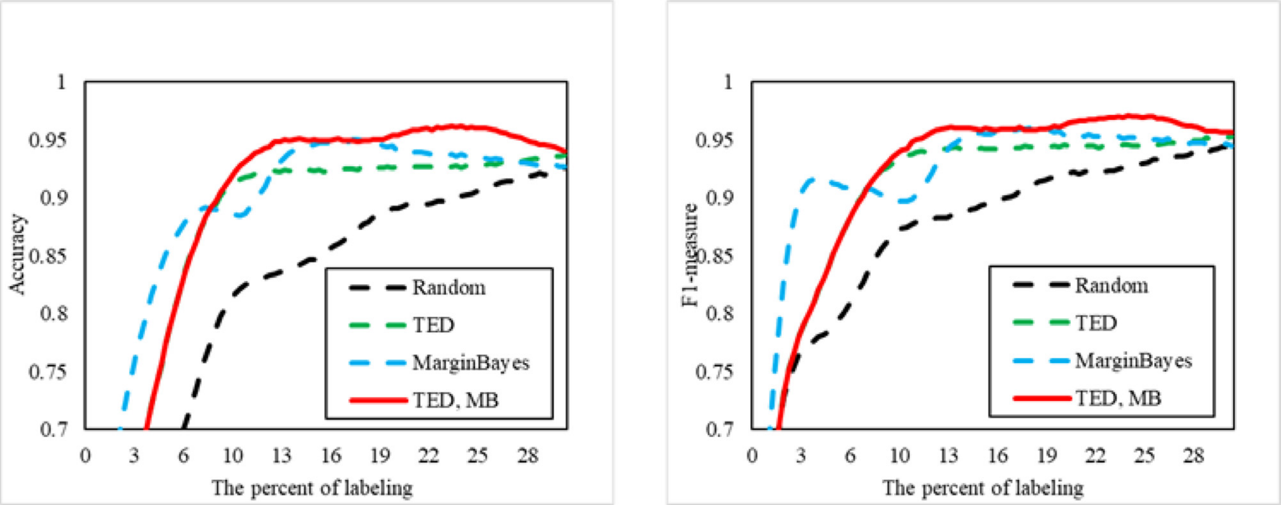


Fig. 9. Performance of RMQCAL with SQCs based on certainty and experimental design.

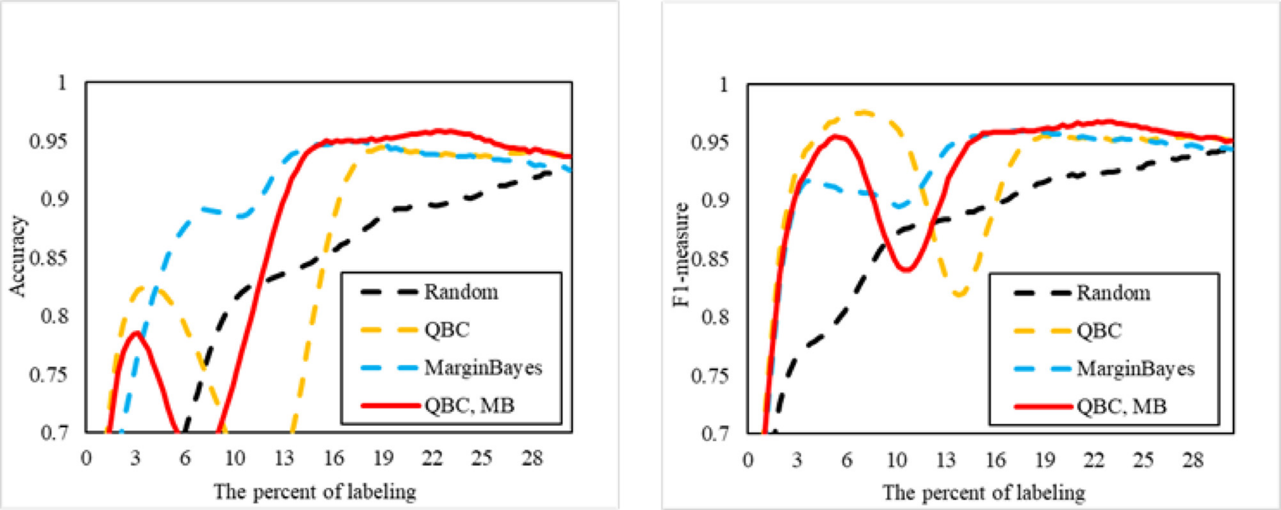


Fig. 10. Performance of RMQCAL with certainty-based and committee-based SQCs.

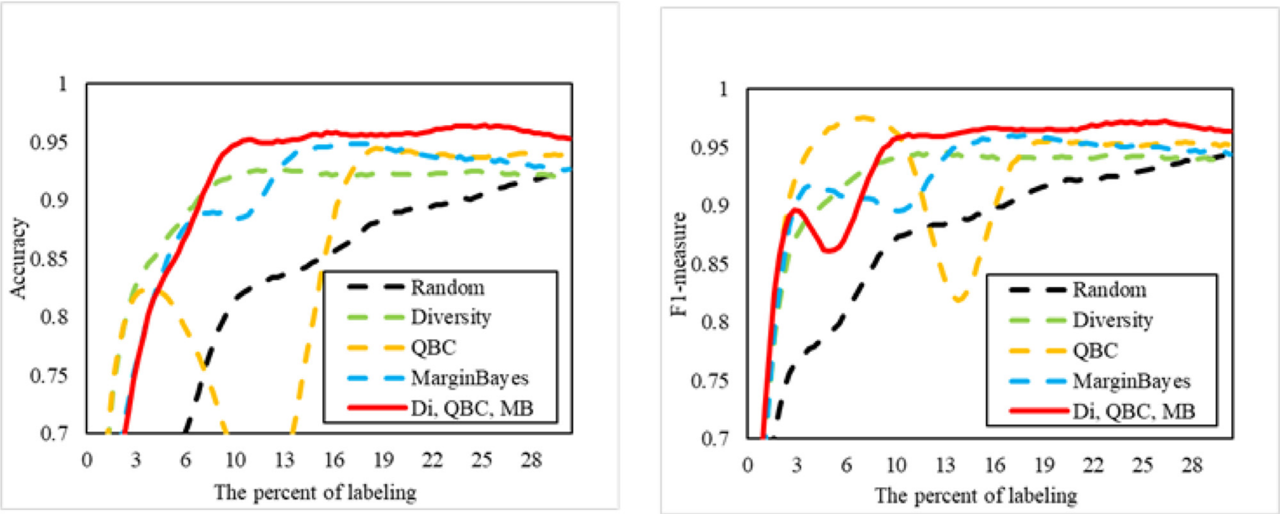


Fig. 11. Performance of RMQCAL with SQCs based on certainty, committee, connection.

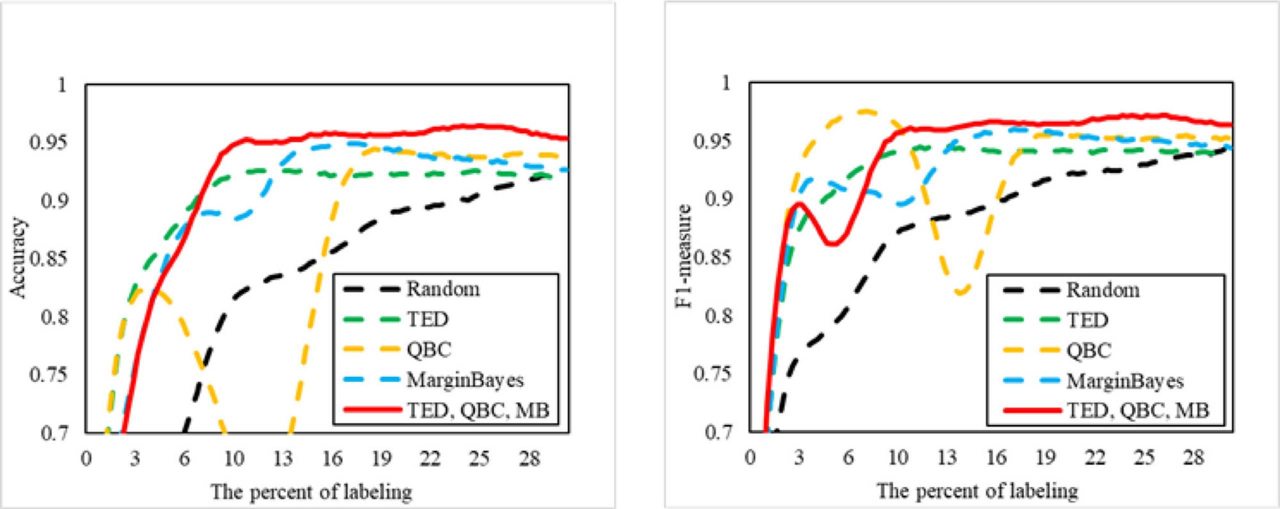


Fig. 12. Performance of RMQCAL with SQCs based on certainty, committee, experimental design.

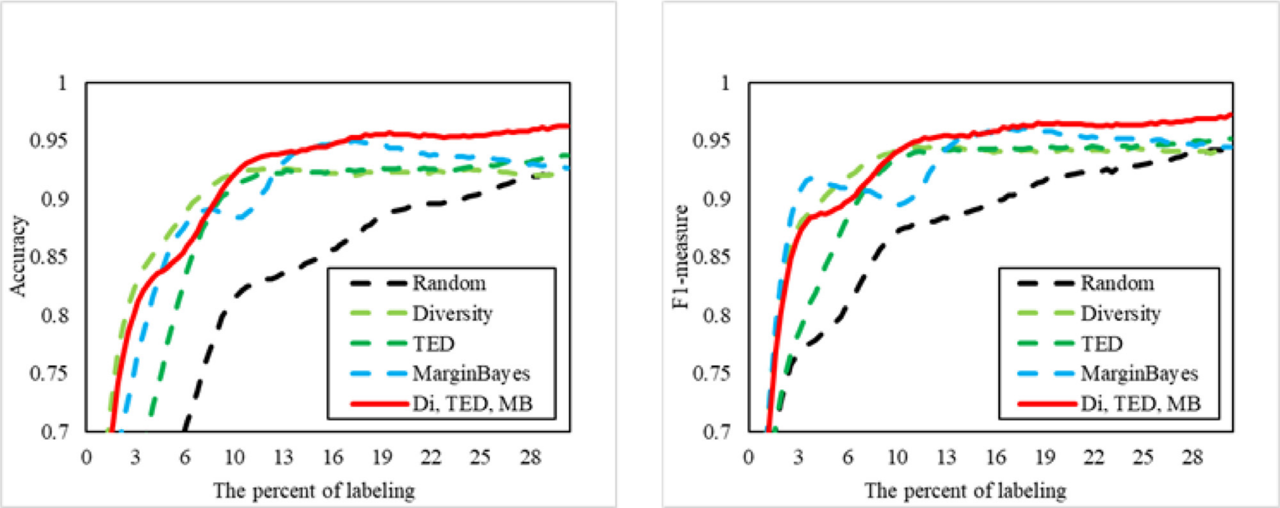


Fig. 13. Performance of RMQCAL with SQCs based on certainty, connection and experimental design.

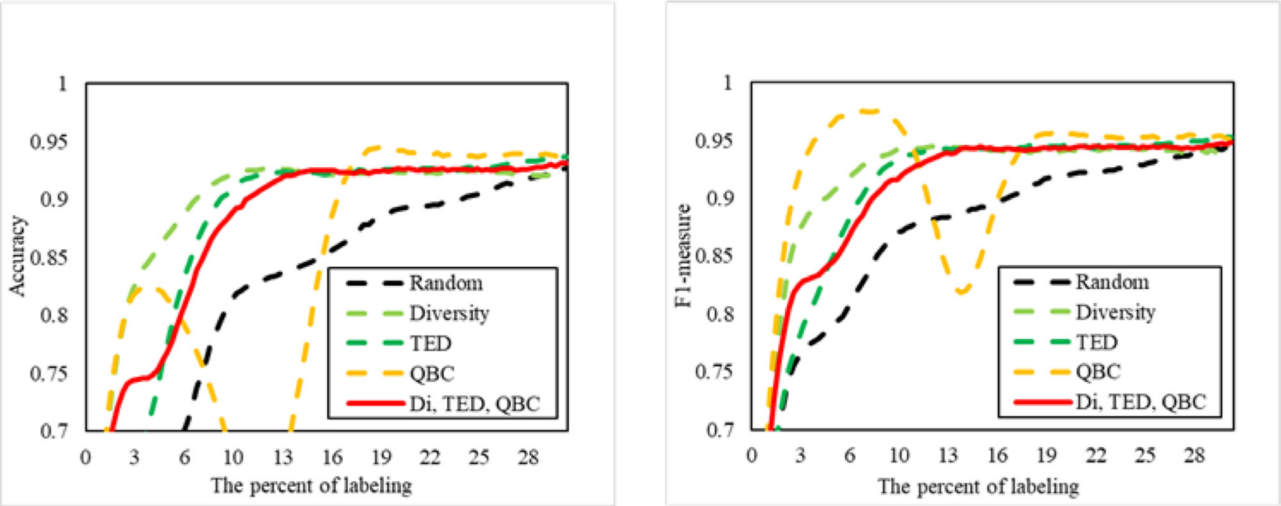


Fig. 14. Performance of RMQCAL with SQCs based on committee, connection, experimental design.

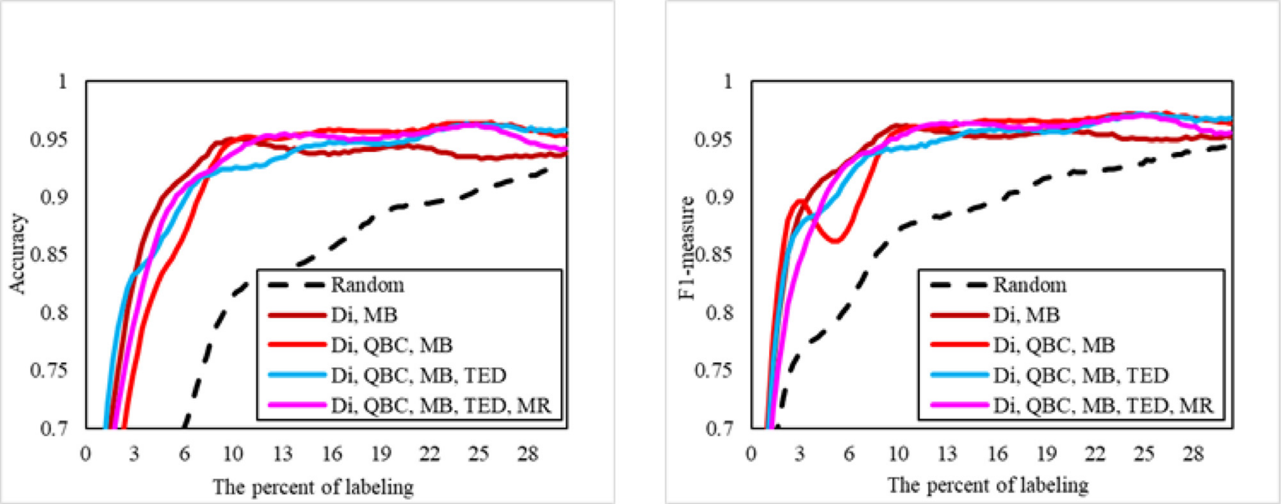


Fig. 15. Performance of RMQCAL with several SQCs.

Analysis of Experiment B:

Experiment B helps explain several of the problems. First, the proposed MQCAL does have high scalability, which would enable it to offer a variety of criteria combinations with less algorithm modification. Second, the RMQCAL with the combination of multiple SQCs usually (not always) performs better than its components,

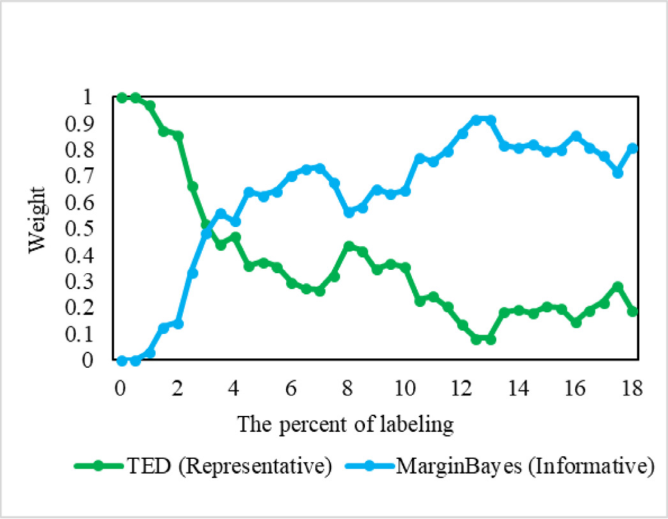


Fig. 16. The weight variations in TED and MB.

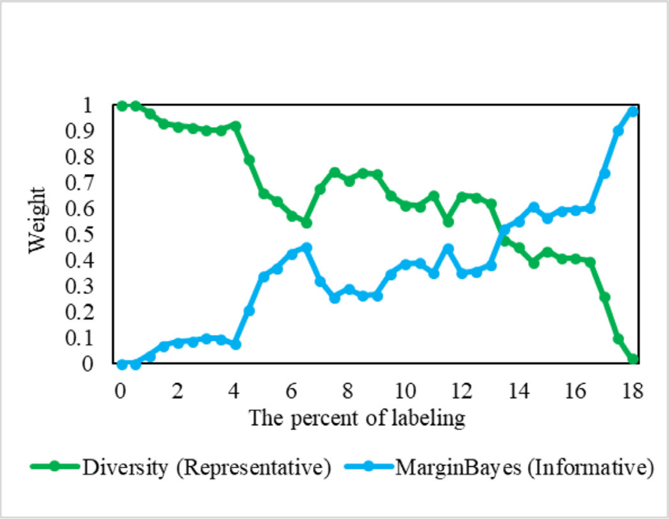


Fig. 17. The weight variations in DI and MB.

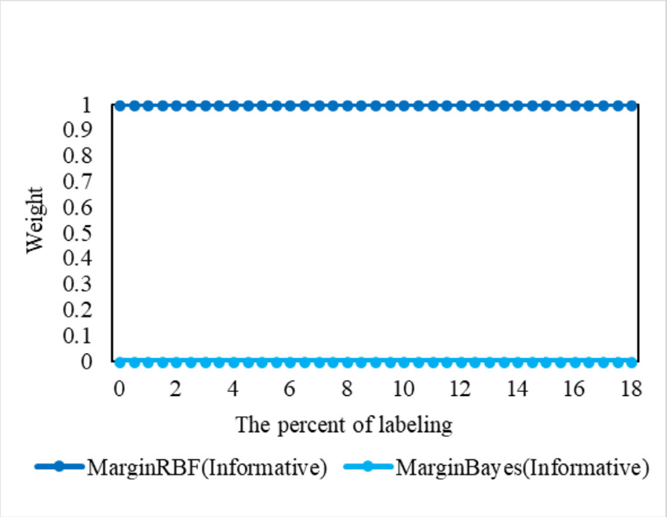


Fig. 18. The weight variations in MR and MB.

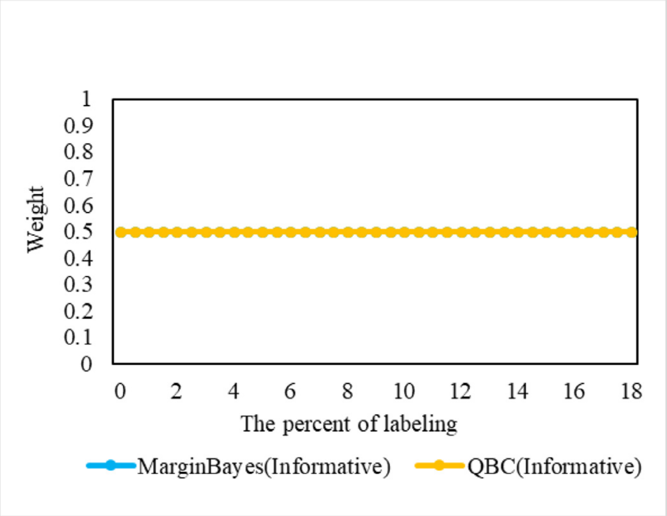


Fig. 19. The weight variations in QBC and MB.

i.e., an AL with a single criterion. However, an inappropriate combination may lead to no definable benefit, as shown, for example, in Figs. 7, 10 and 14. This article considers that the failure of the dynamic weighting process is the primary explanation for these exceptions.

To prove above view, we specifically record weight changes in each involved single criterion under the experimental conditions used for Figs. 7–10 as well as Figs. 16–19, where coordinate-axis X indicates the labeling cost and coordinate-axis Y indicates the value of self-adaptive weights. Under ideal conditions, the weight changes of each single criterion involved in the AL process should be dynamic and should satisfactorily reflect the contribution of each criterion. In the successful cases shown in Figs. 8 and 9, both of their weight changes in Figs. 16 and 17 show a gradual decline in weight for the representativeness measure-based SQC, whereas the weight of the informativeness measure-based SQC rises continuously. Combined with 'the timeliness of AL' described above, such weight changes are just what we need. Conversely, regarding the ineffective cases, the involved two SQCs in Fig. 7 belong to the same category (Certainty), and one (MR) is always better than the others (MB), causing our RMQCAL to assign a higher weight to the MR from beginning to end (as shown in Fig. 18). Additionally, the combination that involved the committee-based SQC in Fig. 10 does not seem to perform well. We attribute this performance to our designed weighting calculation step, in which the involved SQC will be equally weighted only if there are only two SQCs, and one of them is QBC (as shown in Fig. 19). Similarly, it is not surprising that the case shown in Fig. 14 does not perform well not only because it involves the committee-based SQC but also because the other two involved SQCs belong to same type of AL (i.e., representativeness measure based).

Moreover, Fig. 15 indicates that with the increasing number of SQCs from more AL methods, the performance of RMQCAL improves and becomes more stable, although the room for continued improvement diminishes. On the other hand, as shown in Fig. 15, the potential predominance of RMQCAL in practical applications can also be partially supported if we regard these involved AL methods as the candidate methods.

In sum, several rules exist by which SQCs are selected for combination. (1) The SQCs do not have to be numerous. In general, three SQCs are sufficient. (2) The involved SQCs preferably belong to different types of AL. At least one of them is certainty based, and another one is representativeness measure based. Committee-based SQCs should be used in conjunction with two or more different types of SQCs. (3) Certainty-based SQCs can promote the performance of our RMQCAL in the middle stages of the process. Representativeness measure-based SQCs can enhance AL performance in the early stages. QBC can smooth the performance curve.

According to the above rules and the results shown in Fig. 15, we recommend employing Diversity, Margin and QBC as the three involved SQCs in the proposed RMQCAL. These will also be used in all tests shown below. In our view, this combination is both typical and adequate.

Experiment C. Comparisons between RMQCAL and other AL methods

Description of Experiment C:

Experiment C is the focus of our experiments, which compare our research results with various kinds of existing AL methods as the controlled methods, which include RANDOM, MARGIN [41], DIVERSITY [11], QBC [19], CSAL [21] (based on criteria selection), SMQCAL [23] (serial-form), PMQCAL [23] (parallel-form), MCDMAL [9] (based on multicriteria decision-making), DUAL [7] and QUIRE [8].

In these methods, MARGIN, Diversity and QBC are three classic single criterion-based AL methods (*dotted line in the diagram*), whose internal SQCs are also the individual components of the proposed RMQCAL. RMQCAL shall not be determined effective, unless RMQCAL achieve a higher performance than them; CSAL, SMQCAL, PMQCAL and MCDMAL are four well-developed approaches of other forms of MQCAL with different integration criteria strategies (*solid line in the diagram*). Similarly, the performance of RMQCAL must exceed that of these four methods to be considered superior. In addition, DUAL and QUIRE are two existing state-of-the-art AL methods with their own specially designed SQCs and accessional integration criteria strategies (*dashed line in the diagram*), which are used to further illustrate the pros and cons of the proposed RMQCAL.

The SQCs involved in our RMQCAL method include DIVERSITY and MARGIN in the RBF-SVM and QBC, which may be not optimal but nevertheless represent the most typical combination involving three kinds of single-criterion AL methods. With respect to the integration strategy, SQCs employ an improved rank aggregation method based on a Markov chain.

To ensure the validity of the comparative experiments and avoid the effects of other factors, we reproduce an experimental environment that is exactly the same as that described in another paper [8]. The same 10 small-scale datasets (i.e., *Wdbc**, *Vehicle**, *Isolet**, *Titato**, *Austra**, *LetterDP**, *LetterEF**, *LetterIJ**, *LetterMN** and *LetterUV**) are used with the same experimental parameters provided in the corresponding literature to determine whether the proposed RMQCAL is competitive with existing MQCAL methods. Moreover, four additional large-scale datasets (i.e., Mushroom+, EEG+, Mocap+, Epilepsy+) are also introduced to validate the performance of the proposed method on big data issues.

For each dataset, a corresponding experiment is repeated 10 times, and performance curves measuring accuracy (*X*-axis) against labeling costs (*Y*-axis) for the methods are shown in Fig. 20. In addition to the average AUC (MEAN) of each method, as calculated in 5 different stages of the AL process (when labeled samples account for 5%, 10%, 20%, 30% and 40% of the total dataset for the small-scale dataset task, and the number of labeled samples is 5, 65, 125, 185 and 245 for the large-scale dataset task), we also record their standard deviations (SDs) in Tables 3–6. The best result and its performance are recorded in bold based on paired t-tests conducted at the 95 percent significance level. A more detailed comparison (Win/Tie/Loss Counts) between RMQCAL and another MQCAL method is shown in Table 7. It is worth mentioning that partial contrast methods are invalid on large-scale dataset tasks because of their high algorithm complexity. This situation is recorded as ‘Null’ in Table 6. Similarly, in Table 7, the data outside of the parentheses represent the Win counts if we account for the above invalid cases and inside of the parentheses if we do not.

Result of Experiment C:

Table 3
Comparison of the AUC values of the 14 datasets (1).

The labeled samples		5%	10%	20%	30%	40%
Database	Algorithms	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
Austra	RANDOM	0.868 ± 0.027	0.894 ± 0.022	0.897 ± 0.023	0.901 ± 0.022	0.909 ± 0.015
	MARGIN	0.751 ± 0.137	0.838 ± 0.119	0.885 ± 0.043	0.909 ± 0.010	0.911 ± 0.012
	DIVERSITY	0.857 ± 0.038	0.870 ± 0.036	0.892 ± 0.021	0.895 ± 0.019	0.897 ± 0.016
	QBC	0.733 ± 0.073	0.796 ± 0.024	0.820 ± 0.018	0.847 ± 0.028	0.865 ± 0.022
	CSAL	0.820 ± 0.055	0.841 ± 0.055	0.859 ± 0.039	0.864 ± 0.022	0.854 ± 0.020
	SMCDMAL	0.822 ± 0.061	0.850 ± 0.014	0.861 ± 0.023	0.878 ± 0.016	0.885 ± 0.013
	PMCQAL	0.845 ± 0.061	0.834 ± 0.042	0.858 ± 0.025	0.879 ± 0.023	0.885 ± 0.022
	MCDMAL	0.843 ± 0.045	0.838 ± 0.035	0.840 ± 0.027	0.856 ± 0.022	0.864 ± 0.018
	DUAL	0.866 ± 0.037	0.878 ± 0.036	0.875 ± 0.018	0.876 ± 0.016	0.879 ± 0.013
	QUIRE	0.887 ± 0.014	0.901 ± 0.010	0.906 ± 0.016	0.912 ± 0.009	0.914 ± 0.009
	RMQCAL	**0.916 ± 0.010**	**0.922 ± 0.009**	**0.926 ± 0.007**	**0.926 ± 0.005**	**0.929 ± 0.007**
Isolet	RANDOM	0.995 ± 0.006	0.998 ± 0.002	0.999 ± 0.001	**1.000 ± 0.000**	1.000 ± 0.000
	MARGIN	0.965 ± 0.052	0.999 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.000**	1.000 ± 0.000
	DIVERSITY	**0.978 ± 0.056**	0.931 ± 0.093	0.983 ± 0.021	0.993 ± 0.003	0.972 ± 0.037
	QBC	0.950 ± 0.042	0.931 ± 0.067	0.854 ± 0.052	0.848 ± 0.062	0.838 ± 0.075
	CSAL	0.948 ± 0.047	0.934 ± 0.039	0.894 ± 0.051	0.938 ± 0.074	**0.960 ± 0.107**
	SMCDMAL	**1.000 ± 0.001**	**1.000 ± 0.001**	**1.000 ± 0.000**	**1.000 ± 0.000**	1.000 ± 0.000
	PMCQAL	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	MCDMAL	**0.998 ± 0.003**	0.990 ± 0.020	0.939 ± 0.085	0.924 ± 0.099	0.866 ± 0.174
	DUAL	0.993 ± 0.008	0.999 ± 0.001	0.999 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.001**
	QUIRE	0.997 ± 0.002	0.999 ± 0.001	0.999 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.001**
	RMQCAL	**1.000 ± 0.000**	**1.000 ± 0.001**	**1.000 ± 0.000**	**1.000 ± 0.001**	**1.000 ± 0.000**

Table 4
Comparison of the AUC values of the 14 datasets (2).

The labeled samples		5%	10%	20%	30%	40%
Database	Algorithms	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
Titato	RANDOM	**0.762 ± 0.033**	**0.861 ± 0.031**	0.954 ± 0.023	0.979 ± 0.011	0.991 ± 0.007
	MARGIN	0.645 ± 0.096	0.753 ± 0.078	0.946 ± 0.043	0.998 ± 0.001	1.000 ± 0.000
	DIVERSITY	**0.723 ± 0.031**	**0.759 ± 0.034**	0.849 ± 0.015	0.881 ± 0.014	0.901 ± 0.011
	QBC	**0.720 ± 0.035**	**0.779 ± 0.019**	0.829 ± 0.015	0.834 ± 0.018	0.857 ± 0.015
	CSAL	0.645 ± 0.032	0.676 ± 0.045	0.760 ± 0.031	0.836 ± 0.035	0.862 ± 0.022
	SMCDMAL	0.658 ± 0.044	0.711 ± 0.041	0.777 ± 0.030	0.822 ± 0.019	0.843 ± 0.014
	PMCQAL	0.644 ± 0.031	0.692 ± 0.039	0.770 ± 0.042	0.798 ± 0.037	0.839 ± 0.027
	MCDMAL	**0.723 ± 0.029**	0.798 ± 0.027	0.826 ± 0.010	0.848 ± 0.016	0.871 ± 0.023
	DUAL	**0.708 ± 0.069**	0.782 ± 0.064	0.900 ± 0.027	0.981 ± 0.012	0.995 ± 0.006
	QUIRE	**0.736 ± 0.037**	**0.861 ± 0.025**	**0.991 ± 0.004**	**0.999 ± 0.001**	**1.000 ± 0.000**
	RMQCAL	**0.729 ± 0.073**	0.813 ± 0.051	0.861 ± 0.059	0.932 ± 0.024	0.963 ± 0.019
Vehicle	RANDOM	0.818 ± 0.064	**0.864 ± 0.039**	0.925 ± 0.032	0.949 ± 0.026	0.968 ± 0.016
	MARGIN	0.693 ± 0.078	0.828 ± 0.077	0.883 ± 0.105	**0.981 ± 0.014**	**0.993 ± 0.005**
	DIVERSITY	0.807 ± 0.095	**0.843 ± 0.110**	**0.910 ± 0.091**	0.952 ± 0.025	0.956 ± 0.025
	QBC	0.737 ± 0.131	0.784 ± 0.099	**0.924 ± 0.054**	0.962 ± 0.025	0.976 ± 0.013
	CSAL	0.815 ± 0.049	0.852 ± 0.056	0.929 ± 0.025	0.947 ± 0.020	0.959 ± 0.022
	SMCDMAL	0.835 ± 0.075	**0.892 ± 0.060**	0.919 ± 0.048	0.949 ± 0.041	0.975 ± 0.019
	PMCQAL	0.834 ± 0.044	**0.884 ± 0.051**	0.926 ± 0.042	0.946 ± 0.038	0.969 ± 0.019
	MCDMAL	**0.884 ± 0.039**	**0.903 ± 0.040**	0.924 ± 0.026	0.959 ± 0.017	0.973 ± 0.011
	DUAL	0.680 ± 0.074	0.706 ± 0.114	0.817 ± 0.061	0.875 ± 0.035	0.908 ± 0.035
	QUIRE	0.750 ± 0.137	**0.912 ± 0.024**	**0.956 ± 0.025**	**0.985 ± 0.007**	0.989 ± 0.006
	RMQCAL	0.806 ± 0.078	**0.906 ± 0.053**	**0.959 ± 0.015**	**0.986 ± 0.015**	**0.996 ± 0.006**
	RANDOM	0.984 ± 0.006	0.986 ± 0.005	0.990 ± 0.004	0.991 ± 0.004	0.991 ± 0.004
	MARGIN	**0.967 ± 0.038**	0.990 ± 0.002	0.993 ± 0.003	0.993 ± 0.003	0.993 ± 0.003

(*continued on next page*)

Table 4 (*continued*)

The labeled samples		5%	10%	20%	30%	40%
Database	Algorithms	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
Wdbc	DIVERSITY	0.946 ± 0.036	0.964 ± 0.012	0.978 ± 0.008	0.981 ± 0.008	0.983 ± 0.008
	QBC	0.970 ± 0.023	0.974 ± 0.013	0.985 ± 0.007	0.985 ± 0.008	0.986 ± 0.008
	CSAL	0.967 ± 0.016	0.972 ± 0.013	0.974 ± 0.011	0.983 ± 0.006	0.986 ± 0.005
	SMCDMAL	0.954 ± 0.030	0.968 ± 0.015	0.982 ± 0.005	0.984 ± 0.007	0.985 ± 0.004
	PMCQAL	0.923 ± 0.063	0.954 ± 0.027	0.967 ± 0.012	0.984 ± 0.006	0.985 ± 0.006
	MCDMAL	0.934 ± 0.065	0.960 ± 0.032	0.981 ± 0.005	0.982 ± 0.006	0.983 ± 0.007
	DUAL	0.955 ± 0.025	0.964 ± 0.016	0.972 ± 0.015	0.988 ± 0.009	0.992 ± 0.003
	QUIRE	0.985 ± 0.006	0.990 ± 0.004	0.993 ± 0.003	0.993 ± 0.003	0.993 ± 0.003
	RMQCAL	**0.992 ± 0.006**	**0.995 ± 0.005**	**0.997 ± 0.002**	**0.998 ± 0.002**	**0.998 ± 0.002**
LetterDP	RANDOM	0.990 ± 0.004	0.995 ± 0.002	0.997 ± 0.002	0.998 ± 0.001	0.998 ± 0.001
	MARGIN	**0.994 ± 0.005**	**0.999 ± 0.001**	0.999 ± 0.000	0.999 ± 0.001	0.999 ± 0.001
	DIVERSITY	0.984 ± 0.007	0.990 ± 0.004	0.997 ± 0.002	0.999 ± 0.000	0.999 ± 0.000
	QBC	0.986 ± 0.010	0.995 ± 0.003	0.989 ± 0.006	0.986 ± 0.009	0.989 ± 0.006
	CSAL	0.987 ± 0.008	0.996 ± 0.004	**0.999 ± 0.001**	1.000 ± 0.000	**1.000 ± 0.000**
	SMCDMAL	0.992 ± 0.005	**0.998 ± 0.003**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	PMCQAL	0.989 ± 0.007	**0.988 ± 0.019**	**1.000 ± 0.001**	1.000 ± 0.000	**1.000 ± 0.000**
	MCDMAL	0.994 ± 0.003	**0.997 ± 0.003**	**0.999 ± 0.001**	1.000 ± 0.000	**1.000 ± 0.000**
	DUAL	0.978 ± 0.005	0.986 ± 0.001	0.988 ± 0.004	0.990 ± 0.004	0.996 ± 0.001
	QUIRE	**0.998 ± 0.001**	**0.999 ± 0.001**	**0.999 ± 0.001**	0.999 ± 0.001	0.999 ± 0.001
	RMQCAL	**0.997 ± 0.001**	**0.999 ± 0.001**	**0.999 ± 0.001**	1.000 ± 0.000	0.999 ± 0.000

Table 5
Comparison of the AUC values of the 14 datasets (3).

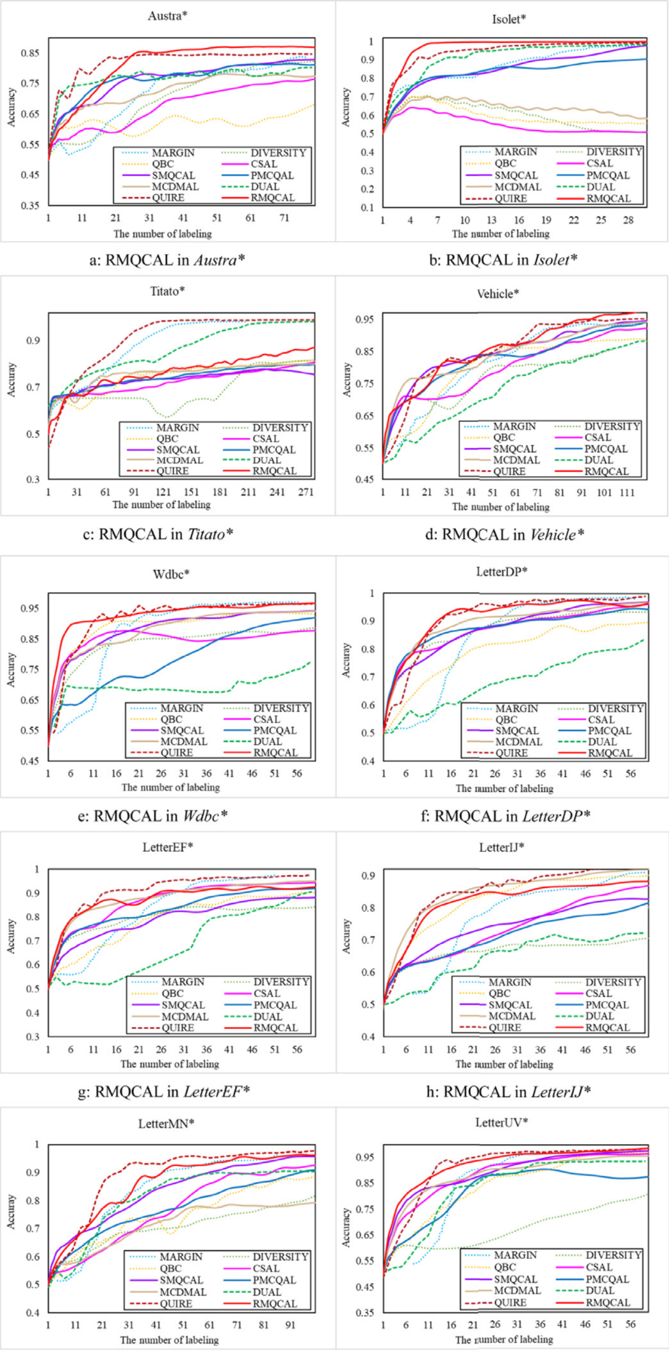
The labeled samples		5%	10%	20%	30%	40%
Database	Algorithms	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
LetterEF	RANDOM	**0.977 ± 0.020**	0.988 ± 0.009	0.994 ± 0.002	0.997 ± 0.002	0.998 ± 0.001
	MARGIN	**0.987 ± 0.008**	**0.999 ± 0.001**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	DIVERSITY	0.957 ± 0.014	0.977 ± 0.008	0.985 ± 0.008	0.992 ± 0.005	0.997 ± 0.003
	QBC	**0.978 ± 0.011**	0.979 ± 0.007	0.995 ± 0.002	0.994 ± 0.003	0.995 ± 0.002
	CSAL	**0.981 ± 0.008**	0.994 ± 0.003	0.999 ± 0.001	1.000 ± 0.000	**1.000 ± 0.000**
	SMCDMAL	0.964 ± 0.012	0.983 ± 0.009	0.995 ± 0.006	0.999 ± 0.001	0.999 ± 0.001
	PMCQAL	**0.970 ± 0.017**	0.984 ± 0.008	0.993 ± 0.006	0.999 ± 0.001	0.999 ± 0.001
	MCDMAL	**0.981 ± 0.013**	0.995 ± 0.004	0.998 ± 0.001	0.999 ± 0.001	**1.000 ± 0.000**
	DUAL	**0.976 ± 0.011**	0.993 ± 0.003	0.996 ± 0.002	0.996 ± 0.002	0.996 ± 0.002
	QUIRE	**0.988 ± 0.009**	0.999 ± 0.000	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	RMQCAL	**0.982 ± 0.007**	0.988 ± 0.004	0.999 ± 0.001	0.999 ± 0.000	0.999 ± 0.000
LetterIJ	RANDOM	**0.943 ± 0.025**	**0.966 ± 0.017**	**0.980 ± 0.004**	0.983 ± 0.005	0.985 ± 0.005
	MARGIN	0.882 ± 0.096	**0.960 ± 0.027**	**0.986 ± 0.005**	**0.989 ± 0.006**	0.991 ± 0.004
	DIVERSITY	0.894 ± 0.031	0.929 ± 0.025	0.959 ± 0.014	0.973 ± 0.008	0.980 ± 0.007
	QBC	0.914 ± 0.022	**0.951 ± 0.012**	**0.968 ± 0.012**	0.973 ± 0.012	0.977 ± 0.008
	CSAL	0.921 ± 0.022	**0.958 ± 0.008**	**0.978 ± 0.007**	**0.986 ± 0.005**	**0.991 ± 0.003**
	SMCDMAL	0.929 ± 0.021	**0.955 ± 0.013**	**0.977 ± 0.009**	0.982 ± 0.007	0.985 ± 0.006
	PMCQAL	0.880 ± 0.050	0.925 ± 0.050	**0.964 ± 0.014**	**0.983 ± 0.008**	0.988 ± 0.005
	MCDMAL	**0.954 ± 0.012**	**0.973 ± 0.008**	**0.982 ± 0.007**	**0.987 ± 0.005**	0.989 ± 0.004
	DUAL	0.819 ± 0.120	0.897 ± 0.058	0.934 ± 0.030	0.954 ± 0.017	0.959 ± 0.014
	QUIRE	**0.951 ± 0.023**	**0.963 ± 0.013**	**0.976 ± 0.011**	**0.989 ± 0.010**	0.991 ± 0.004
	RMQCAL	**0.955 ± 0.011**	**0.965 ± 0.023**	**0.977 ± 0.017**	**0.988 ± 0.003**	0.989 ± 0.003
LetterMN	RANDOM	0.977 ± 0.010	**0.992 ± 0.002**	0.994 ± 0.003	0.996 ± 0.002	0.997 ± 0.001
	MARGIN	0.964 ± 0.040	**0.991 ± 0.014**	**0.999 ± 0.000**	0.999 ± 0.000	0.999 ± 0.000
	DIVERSITY	0.834 ± 0.059	0.904 ± 0.047	0.978 ± 0.011	0.994 ± 0.004	0.998 ± 0.001
	QBC	0.978 ± 0.013	0.977 ± 0.016	0.989 ± 0.004	0.992 ± 0.003	0.995 ± 0.002
	CSAL	0.934 ± 0.039	0.979 ± 0.006	0.990 ± 0.004	0.994 ± 0.005	0.997 ± 0.002
	SMCDMAL	0.942 ± 0.027	**0.989 ± 0.005**	0.998 ± 0.003	**1.000 ± 0.001**	**1.000 ± 0.001**
	PMCQAL	0.872 ± 0.035	0.953 ± 0.024	0.988 ± 0.009	0.998 ± 0.001	0.999 ± 0.000
	MCDMAL	0.935 ± 0.022	0.960 ± 0.023	0.972 ± 0.013	0.991 ± 0.004	0.995 ± 0.001
	DUAL	0.950 ± 0.025	0.972 ± 0.011	0.974 ± 0.007	0.980 ± 0.008	0.983 ± 0.007
	QUIRE	**0.986 ± 0.007**	**0.996 ± 0.003**	0.998 ± 0.001	0.999 ± 0.000	0.999 ± 0.000
	RMQCAL	0.973 ± 0.010	**0.990 ± 0.008**	0.997 ± 0.001	0.998 ± 0.001	0.998 ± 0.001
LetterUV	RANDOM	0.992 ± 0.005	0.996 ± 0.004	0.998 ± 0.001	0.999 ± 0.000	1.000 ± 0.000
	MARGIN	0.998 ± 0.002	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	DIVERSITY	0.973 ± 0.020	0.986 ± 0.010	0.994 ± 0.006	**1.000 ± 0.001**	1.000 ± 0.001
	QBC	0.996 ± 0.004	0.997 ± 0.002	0.998 ± 0.001	0.998 ± 0.001	0.998 ± 0.001
	CSAL	0.992 ± 0.005	0.998 ± 0.002	1.000 ± 0.000	**1.000 ± 0.000**	1.000 ± 0.000
	SMCDMAL	0.992 ± 0.006	0.998 ± 0.002	1.000 ± 0.000	**1.000 ± 0.000**	1.000 ± 0.000
	PMCQAL	0.987 ± 0.008	0.997 ± 0.003	1.000 ± 0.000	**1.000 ± 0.000**	1.000 ± 0.000
	MCDMAL	0.993 ± 0.003	0.998 ± 0.001	0.999 ± 0.000	**1.000 ± 0.001**	1.000 ± 0.001
	DUAL	0.983 ± 0.014	0.986 ± 0.008	0.990 ± 0.008	0.991 ± 0.008	0.993 ± 0.007
	QUIRE	**0.999 ± 0.001**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	RMQCAL	0.993 ± 0.004	0.999 ± 0.001	1.000 ± 0.001	**1.000 ± 0.001**	0.999 ± 0.000

Table 6
Comparison of the AUC values of the 14 datasets (4).

The labeled samples		5	65	125	185	245
Database	Algorithms	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
EEG+	RANDOM	**0.513 ± 0.059**	**0.567 ± 0.040**	0.567 ± 0.054	0.581 ± 0.042	0.597 ± 0.037
	MARGIN	0.496 ± 0.032	**0.625 ± 0.039**	**0.655 ± 0.054**	**0.697 ± 0.054**	**0.726 ± 0.071**
	DIVERSITY	**0.532 ± 0.063**	**0.565 ± 0.035**	**0.608 ± 0.029**	0.616 ± 0.033	0.635 ± 0.044
	QBC	**0.509 ± 0.049**	0.531 ± 0.049	0.541 ± 0.041	0.545 ± 0.042	0.551 ± 0.045
	CSAL	**0.512 ± 0.055**	**0.589 ± 0.029**	**0.606 ± 0.036**	0.611 ± 0.027	0.618 ± 0.032
	SMCDMAL	**0.513 ± 0.046**	**0.581 ± 0.046**	**0.614 ± 0.063**	0.628 ± 0.079	0.650 ± 0.091
	PMCQAL	Null	Null	Null	Null	Null
	MCDMAL	**0.531 ± 0.042**	**0.621 ± 0.044**	**0.655 ± 0.049**	**0.677 ± 0.055**	**0.707 ± 0.063**
	DUAL	Null	Null	Null	Null	Null
	QUIRE	Null	Null	Null	Null	Null
	RMQCAL	**0.539 ± 0.049**	**0.602 ± 0.056**	**0.657 ± 0.072**	**0.715 ± 0.088**	**0.751 ± 0.071**
Epilepsy+	RANDOM	0.503 ± 0.023	0.544 ± 0.016	0.625 ± 0.028	0.714 ± 0.037	0.777 ± 0.039
	MARGIN	**0.984 ± 0.007**	0.985 ± 0.005	0.985 ± 0.007	0.986 ± 0.007	0.989 ± 0.005
	DIVERSITY	**0.984 ± 0.006**	0.984 ± 0.006	0.984 ± 0.006	0.986 ± 0.006	0.988 ± 0.006
	QBC	0.863 ± 0.043	0.963 ± 0.035	**0.969 ± 0.037**	**0.970 ± 0.037**	0.975 ± 0.023
	CSAL	**0.984 ± 0.007**	0.984 ± 0.007	0.984 ± 0.007	0.984 ± 0.007	0.984 ± 0.007
	SMCDMAL	0.850 ± 0.060	0.978 ± 0.005	0.984 ± 0.004	0.984 ± 0.005	0.986 ± 0.003
	PMCQAL	Null	Null	Null	Null	Null
	MCDMAL	**0.984 ± 0.008**	0.985 ± 0.004	0.989 ± 0.004	**0.992 ± 0.002**	**0.994 ± 0.002**
	DUAL	Null	Null	Null	Null	Null
	QUIRE	Null	Null	Null	Null	Null
	RMQCAL	**0.985 ± 0.008**	**0.990 ± 0.003**	**0.992 ± 0.001**	**0.994 ± 0.002**	**0.994 ± 0.001**
Mocap+	RANDOM	**0.831 ± 0.111**	0.916 ± 0.015	0.931 ± 0.012	0.941 ± 0.009	0.949 ± 0.009
	MARGIN	**0.832 ± 0.114**	**0.918 ± 0.094**	**0.939 ± 0.080**	**0.943 ± 0.079**	**0.973 ± 0.014**
	DIVERSITY	0.801 ± 0.099	0.897 ± 0.008	0.914 ± 0.008	0.922 ± 0.010	0.929 ± 0.007
	QBC	0.710 ± 0.145	0.918 ± 0.020	0.939 ± 0.011	0.946 ± 0.011	0.950 ± 0.011
	CSAL	0.743 ± 0.121	0.834 ± 0.076	0.912 ± 0.035	0.934 ± 0.008	0.943 ± 0.004
	SMCDMAL	0.856 ± 0.034	**0.937 ± 0.032**	**0.962 ± 0.008**	**0.967 ± 0.010**	**0.969 ± 0.010**
	PMCQAL	Null	Null	Null	Null	Null
	MCDMAL	**0.824 ± 0.143**	**0.937 ± 0.048**	**0.966 ± 0.011**	**0.969 ± 0.013**	**0.970 ± 0.012**
	DUAL	Null	Null	Null	Null	Null
	QUIRE	Null	Null	Null	Null	Null
	RMQCAL	**0.900 ± 0.020**	**0.954 ± 0.009**	**0.963 ± 0.011**	**0.965 ± 0.012**	**0.969 ± 0.012**
Mushroom+	RANDOM	0.806 ± 0.105	0.926 ± 0.026	0.964 ± 0.008	0.982 ± 0.007	0.987 ± 0.008
	MARGIN	**0.889 ± 0.038**	**0.992 ± 0.003**	**0.996 ± 0.002**	**0.997 ± 0.002**	**0.997 ± 0.002**
	DIVERSITY	**0.858 ± 0.082**	**0.991 ± 0.004**	**0.995 ± 0.002**	**0.997 ± 0.002**	**0.997 ± 0.002**
	QBC	**0.892 ± 0.077**	**0.976 ± 0.020**	0.986 ± 0.010	**0.992 ± 0.005**	**0.994 ± 0.003**
	CSAL	0.650 ± 0.159	**0.993 ± 0.004**	**0.998 ± 0.002**	**0.998 ± 0.001**	**0.999 ± 0.001**
	SMCDMAL	**0.754 ± 0.174**	**0.991 ± 0.009**	**0.994 ± 0.007**	**0.998 ± 0.002**	**0.998 ± 0.001**
	PMCQAL	Null	Null	Null	Null	Null
	MCDMAL	**0.861 ± 0.062**	**0.992 ± 0.003**	**0.995 ± 0.003**	**0.997 ± 0.001**	**0.998 ± 0.001**
	DUAL	Null	Null	Null	Null	Null
	QUIRE	Null	Null	Null	Null	Null
	RMQCAL	**0.813 ± 0.119**	**0.988 ± 0.008**	**0.996 ± 0.005**	**0.996 ± 0.005**	**0.996 ± 0.004**

Table 7
Win/Tie/Loss Counts of RMQCAL and Other Methods using Paired t-Tests for the 14 datasets.

	1th	2th	3th	4th	5th	IN ALL(*t*-test)
Algorithms	WIN/TIE/LOSS	WIN/TIE/LOSS	WIN/TIE/LOSS	WIN/TIE/LOSS	WIN/TIE/LOSS	WIN/TIE/LOSS
RANDOM	5/9/0	8/5/1	12/1/1	12/1/1	12/0/2	49/16/5
MARGIN	6/7/1	5/7/2	5/5/4	3/8/3	4/6/4	23/33/14
DIVERSITY	8/6/0	11/3/0	10/4/0	12/2/0	10/4/0	51/19/0
QBC	7/7/0	11/3/0	10/4/0	12/2/0	13/1/0	53/17/0
COMPARE1	21/20/1	27/13/2	25/13/4	27/12/3	27/11/4	127/69/14
CSAL	9/5/0	9/4/1	8/5/1	9/5/0	7/3/4	42/22/6
SMCDMAL	9/5/0	4/10/0	5/9/0	7/5/2	7/4/3	32/33/5
PMCQAL	11(7)/3/0	10(6)/4/0	10(6)/4/0	8(4)/6/0	8(4)/3/3	47(27)/20/3
MCDMAL	4/9/1	5/8/1	6/8/0	6/8/0	6/6/2	27/39/4
COMPARE2	33(29)/22/1	28(24)/26/2	29(25)/26/1	30(26)/24/2	28(24)/16/12	148(128)/114/18
DUAL	12(8)/2/0	12(8)/1/1	13(9)/1/0	12(8)/1/1	12(8)/1/1	61(41)/6/3
QUIRE	7(3)/5/2	7(3)/4/3	7(3)/4/3	6(2)/5/3	7(3)/3/4	34(14)/21/15
COMPARE3	19(11)/7/2	19(11)/5/4	20(12)/5/3	18(10)/6/4	19(11)/4/5	95(55)/27/18
IN ALL	**78(66)/58/4**	**82(70)/49/9**	**86(74)/45/9**	**87(75)/43/10**	**86(74)/31/23**	**419(359)/226/55**



a: RMQCAL in *Austra**

b: RMQCAL in *Isolet**

c: RMQCAL in *Titato**

d: RMQCAL in *Vehicle**

e: RMQCAL in *Wdbc**

f: RMQCAL in *LetterDP**

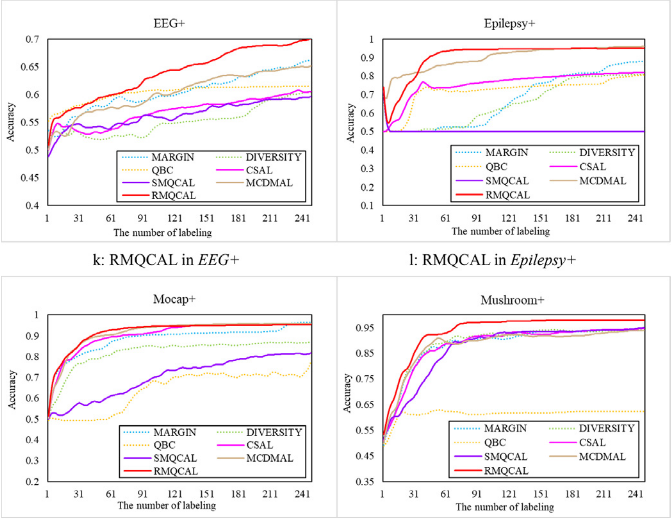
g: RMQCAL in *LetterEF**

h: RMQCAL in *LetterIJ**

i: RMQCAL in *LetterMN**

j: RMQCAL in *LetterUV*

Fig. 20. Comparisons of the accuracy of the 14 datasets.



k: RMQCAL in *EEG+*

l: RMQCAL in *Epilepsy+*

M: RMQCAL in *Mocap+*

N: RMQCAL in *Mushroom+*

Fig. 20. Continued

Analysis of Experiment C:

Experiment C is the focus of our experiments, through which the ideals mentioned before have been verified and following conclusions have been obtained.

According to Fig. 20 and Tables 3–6, the performance of the DIVERSITY method (one kind of representativeness measure-based AL) in most of the above cases is much better than that of the MARGIN method (one kind of informativeness measure-based AL) at the early stage of the AL process; however, the MARGIN method gradually improves with the increased number of labeled samples. In the other words, we obtained the first conclusion that the timeliness of AL indeed exists.

The results shown in Fig. 20 and in the 1st to 4th lines of Table 7 illustrate that RMQCAL is not only significantly better than the random approach but also much better than MARGIN, DIVERSITY and QBC, which are the components of both RMQCAL and the conventional AL methods with a single criterion. The 5th line in Table 7 shows that RMQCAL wins or ties account for nearly 93 percent of the total, further confirming that through RMQCAL, users can obtain a better classification model with a lower labeling cost by implementing a combination of multiple appropriate AL methods rather than any individual AL method. Therefore, RMQCAL is indeed effective, confirming our second conclusion.

Regarding MQCAL with various integration criteria strategies (i.e., IDE, DUAL, SMQCAL, PMQCAL, CSAL and MCDMAL), as the results shown in the 6th to 10th lines of Table 7, although we focused our best efforts on tuning their related weight parameters or directly using the recommended values provided in the corresponding studies, these methods are still not better than RMQCAL; moreover, they are sometimes inferior to conventional AL methods based on a single criterion. We suggest that the relatively low uniformity and generality of these MQCAL methods, caused by excessive dependence on empirical parameters and the tuning process, are the reasons for their suboptimal performance. In addition, from the perspective of these results, our third conclusion positing that RMQCAL achieves superiority is also validated.

The rest of experimental results from Fig. 20 and the 11th to 14th lines of Table 7 provide further evidence of the above third conclusion. Even compared to the state-of-the-art AL methods DUAL and QUIRE, which have high citations, the proposed RMQCAL is still quite competitive, especially for the large-scale database tasks. QUIRE and DUAL are completely unusable to address big data issues because of our limited runtime environment, and their corresponding highly complex self-similarity acquisition and pre-clustering [42] process inside them. More research about the computational complexity of each method is discussed in Experiment D.

Additionally, the last four diagrams in Fig. 20 also indicate that except for the *EEG*+dataset, the AL performance curve on other large-scale database tasks grows and flattens with less than 100 samples, which also demonstrates the potential value of AL algorithms in big data problems.

Experiment D. Comparing the CPU time for our RMQCAL and another AL method

Description of Experiment D:

In Experiment D, each involved method is run ten times; the average CPU time for each query in each method is recorded (in seconds), as shown in Table 8. The control methods can be divided into three types: single-query criterion-based AL, MQCAL with different integration criteria strategies, and state-of-the-art MQCAL. There are two points worth mentioning. First, Null means that the corresponding AL methods are inestimable and cannot be used in this dataset under our experimental conditions. Second, the SQCs involved in our RMQCAL method are DIVERSITY and MARGIN in the RBF-SVM and QBC, and the integration criteria strategy is the MC2-based rank aggregation method.

Results of Experiment D:

Analysis of Experiment D:

The following conclusions are obtained from the results of Experiment D. (1) Unsurprisingly, all single criterion-based AL methods are more efficient than RMQCAL because they are also components of our RMQCAL, and the CPU time of RMQCAL is approximately equal to the sum of the times spent by each constituent SQC. (2) As expected, the Markov chain does not require more CPU time with the added step of sample truncation. (3) Compared with MQCAL with different integration criteria strategies and state-of-the-art MQCAL, RMQCAL is comparatively efficient and only second to SMQCAL. We believe that this result is acceptable because the efficiency of SMQCAL comes at the cost of performance. The multilayer filter, similar to the design of SMQCAL, can indeed reduce the operational time significantly. However, such a design will miss many of the samples with high comprehensive values, leading to a suboptimal result, as shown in Experiment C. (4) For the large-scale database, this article does not recommend the use of QUIRE, DUAL and PMQCAL. All of these are too inefficient and may even fail to work when the operational environment is not adequately established.

4. Discussion

In summary, through Experiment A and Experiment B, we determine the best rank aggregation methods and the most suitable SQC combination for the proposed RMQCAL process. Experiment C and Experiment D further indicate that, compared with other methods, RMQCAL can truly help the user establish a strong prediction model with lower labeling costs and less running time.

However, apart from the above positive views of the proposed RMQCAL, there are still three undesirable issues found in Experiment C that cannot be ignored: (1) RMQCAL seems to

Table 8
The comparison of CPU time between our RMQCAL method and another AL method.

	Single-query criterion-based AL			MQCAL with different integration criteria strategies				State-of-the-art MQCAL		
	Margin	Diversity	QBC	CSAL	SMCQAL	PMCQAL	MCDMAL	DUAL	QUIRE	RMQCAL
Austra	0.016	0.500	0.094	0.641	0.047	1.500	1.031	3.343	0.313	**0.719**
Isolet	0.069	0.517	0.469	1.031	0.063	2.094	1.156	13.721	0.250	**1.109**
Titato	0.031	0.750	0.109	1.109	0.047	2.938	1.609	3.199	1.453	**0.906**
Vehicle	0.003	0.313	0.094	0.469	0.016	0.500	0.531	0.377	0.109	**0.406**
Wdbc	0.006	0.422	0.078	0.688	0.016	1.250	0.813	0.677	0.172	**0.547**
LetterDP	0.031	1.406	0.156	1.828	0.031	9.766	3.703	10.835	7.875	**1.484**
LetterEF	0.034	1.266	0.156	1.500	0.063	8.891	3.230	1.815	6.950	**1.422**
LetterIJ	0.047	1.188	0.125	1.750	0.031	9.871	3.063	2.782	6.350	**1.406**
LetterMN	0.033	1.219	0.172	1.875	0.047	10.172	3.438	9.665	7.469	**0.929**
LetterUV	0.031	1.281	0.203	1.578	0.016	10.953	3.594	7.781	7.469	**0.971**
EEG	0.469	14.766	1.328	16.109	0.391	Null	37.063	Null	Null	**13.798**
Epileptic	0.094	3.969	0.844	5.594	0.125	150.650	7.094	397.214	242.540	**5.050**
Mocap	1.547	26.672	1.953	35.719	1.766	Null	185.578	Null	Null	**28.938**
Mushroom	0.172	6.328	0.625	13.500	0.188	Null	13.500	Null	Null	**7.234**

be a total failure with dataset *Titato**; (2) on the basis of the Win/Tie/Loss counts inside the brackets in the 12th line of Table 7, RMQCAL is slightly worse than QUIRE if we only consider the small-scale dataset cases where QUIRE is available; and (3) even without considering QUIRE and dataset *Titato**, it is very difficult for RMQCAL to consistently deliver the best performance in each iteration of the AL process.

From the first undesirable indication, we suspect that the failure of QBC and DIVERSITY simultaneously, caused by the special data distribution of dataset Titato*, is the main reason that the RMQCAL is invalid in this dataset. The feature vector of the samples in dataset Titato* is too small, and their feature value is just ternary (0, 1 or empty).

Regarding the second undesirable view, we discover that the RMQCAL is slightly worse than QUIRE in the small-scale dataset tasks. However, we still suggest that the second-place performance of the RMQCAL method relative to the other methods could be acceptable for the following reasons. First, the design intentions of RMQCAL and QUIRE are completely different. The main research content in QUIRE is the design of SQC for both representativeness and informativeness measures rather than how to combine them together. Conversely, RMQCAL focuses on the design of the integration criteria strategy, which looks more like CSAL, PMQCAL, SMQCAL and MCDMAL. Second, the above experiment confirms that RMQCAL can produce an effect similar to a well-designed QUIRE by combining several existing ordinary SQCs. However, the involved SQC can also be QUIRE with small improvements. The tests in the appendix prove that the combination of QUIRE and other SQCs via RMQCAL can provide better results than using QUIRE alone. Of course, this would be at the expense of increasing computational complexity. Third, compared to QUIRE, RMQCAL has higher efficiency and is available for big data tasks, as discussed in Experiment D. Moreover, the performance of QUIRE is very sensitive to empirical parameters. The performance of QUIRE can be worse than that of RMQCAL if it has inappropriate parameters. However, the acquisition of optimal parameters requires extra labeling costs to build the validation set.

To the third undesirable view, it is well known that the performance of AL methods depends not only on their design principles but also on many other factors, such as the first batch of labeled samples, the data distribution of the applied database, the current positive or negative labeled sample ratio, and iteration times. It is extremely difficult to have one AL method achieve the best results consistently in each iteration of the AL process. Therefore, as with similar implied views in the related literature [7–9], we also believe that one AL method could be regarded as the best only if its performance can exceed that of other methods with the maximum probability.

It is undeniable that the proposed RMQCAL will not always be able to deliver the best performance on every dataset and for each iteration of the AL process. However, because it can usually outperform most of the other methods and is superior to QUIRE on some special occasions, we remain convinced that the proposed RMQCAL still has a certain progressive significance.

5. Conclusion

In this paper, a means is presented for training data selection in AL problems. Unlike conventional AL methods, it can be ensured that the samples selected for labeling are overall valuable because multiple SQCs are involved in the proposed method and are combined by the introduction of a weighted rank aggregation.

The proposed RMQCAL avoids building a multi-layered filter-like process or solving complex optimization equations, and this capability is highlighted as the main contribution of this study. With respect to advantages, the proposed RMQCAL favorably inherits the merits of most existing MQCAL methods. When applying our RMQCAL, less human intervention is required and fewer empirical parameters are used, and any number and type of SQC can be used and blended into one through dynamical weighting.

To achieve optimal performance, several combinations of SQCs adapted from conventional AL methods were applied. Moreover, existing ranking aggregation methods (e.g., Bordas, Bucklin voting and Markov chain methods) were improved as a key facet of our RMQCAL process. In addition to applying these methods, we employed other ranking aggregation methods, including Thurstone's model, the cross-entropy Monte Carlo model, and the Condorcet model. However, as some methods oppose the AL method in theory or have a run time for realizing one AL iteration that is too long, these methods are not used in our MQCAL. Nevertheless, other more appropriate methods may exist.

Our experimental results show that our newly designed RMQCAL is more effective than the conventional SQC-based AL method. Relative to other MQCAL models, RMQCAL is also rated among the best. Either for a conventional classification task or a large-scale data classification task, the proposed RMQCAL has the ability to do well in helping users train a superior classification model with fewer labeling costs and less running time. Moreover, RMQCAL, in our view, can be an appropriate solution for practical issues, especially when there is no validation set in hand and the labeling cost of each sample is very expensive.

Our planned future work will focus on three main points. First, we will attempt to extend our method to more complex classification or regression problems, e.g., multiclass and multi-labeled problems, and our latest research indicates that RMQCAL also performs well in ordinal regression. Second, the theoretical proof of RMQCAL should be studied further. Finally, we will attempt to apply this approach to medical lesion recognition.

Acknowledgments

This research is partially supported by the National Key Research and Development Program (2016YFC0106200), the 863 National Research Fund (2015AA043203), and the Chinese NSFC Research Fund (61190120, 61190124 and 61271318). The authors are grateful to the University of South Florida and Sandia National Laboratories, who provide UCI as a resource for our experimental data. We also express our sincere gratitude to the Department of Computer Science & Technology of Nanjing University for providing their experimental results for comparison in this paper.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2019.03.029.

References

[1] B. Settles, in: Active Learning Literature Survey, 15, University of Wisconsin, Madison, 2010, pp. 201–221, doi:10.1.1.167.4245.

[2] I. Muslea, S. Minton, C.A. Knoblock, Active learning with multiple views, J. Artif. Intell. Res. 27 (2006) 203–233, doi:10.1613/jair.2005.

[3] N. Panda, K.-S. Goh, E.Y. Chang, Active learning in very large databases, Multimed. Tools Appl. 31 (2006) 249–267, doi:10.1007/s11042-006-0043-1.

[4] Z. Xu, K. Yu, V. Tresp, X. Xu, J. Wang, Representative sampling for text classification using support vector machines, in: F. Sebastiani (Ed.), Advances in Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 393–407.

[5] H. Huang, C. Zhang, Q. Hu, P. Zhu, Multi-view representative and informative induced active learning, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, pp. 139–151, doi:10.1007/978-3-319-42911-3_12.

[6] E. Lughofer, Hybrid active learning for reducing the annotation effort of operators in classification systems, Pattern Recognit. 45 (2012) 884–896, doi:10.1016/j.patcog.2011.08.009.

[7] P. Donmez, J.G. Carbonell, P.N. Bennett, Dual strategy active learning, in: Proceedings of the European Conference on Machine Learning, ECML 2007, 2007, pp. 116–127, doi:10.1007/978-3-540-74958-5_14.

[8] S.J. Huang, R. Jin, Z.H. Zhou, Active learning by querying informative and representative examples, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 1936–1949, doi:10.1109/TPAMI.2014.2307881.

[9] R. Wang, S. Kwong, Active learning with multi-criteria decision making systems, Pattern Recognit. 47 (2014) 3106–3119, doi:10.1016/j.patcog.2014.03.011.

[10] S. Dasgupta, D. Hsu, Hierarchical sampling for active learning, in: Proceedings of the 25th International Conference on Machine Learning - ICML 08, 2008, pp. 208–215, doi:10.1145/1390156.1390183.

[11] Y. Jiao, P. Zhao, J. Wu, Y. Shi, Z. Cui, A multicriterion query-based batch mode active learning technique, in: Foundations of Intelligent Systems, Springer, 2014, pp. 669–680.

[12] S. Jiang, O.U. Qing-Yu, Batch-mode active learning approach of computer viruses classifier based on information density, J. Nav. Univ. Eng. 4 (2015) 31–35.

[13] K. Yu, J. Bi, V. Tresp, Active learning via transductive experimental design, in: Proceedings of the 23rd International Conference on Machine Learning, ICML 06, 148, 2006, pp. 1081–1088, doi:10.1145/1143844.1143980.

[14] D. Cai, X. He, Manifold adaptive experimental design for text categorization, IEEE Trans. Knowl. Data Eng. 24 (2012) 707–719, doi:10.1109/TKDE.2011.104.

[15] C.C. Chang, B.H. Liao, Active learning based on minimization of the expected path-length of random walks on the learned manifold structure, Pattern Recognit. 71 (2017) 337–348, doi:10.1016/j.patcog.2017.06.001.

[16] Z. Wang, S. Yan, C. Zhang, Active learning with adaptive regularization, Pattern Recognit. 44 (2011) 2375–2383, doi:10.1016/j.patcog.2011.03.008.

[17] A. Holub, P. Perona, M.C. Burl, Entropy-based active learning for object recognition, in: Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, pp. 1–8, doi:10.1109/CVPRW.2008.4563068.

[18] N. Roy, A. Mccallum, M.W. Com, Toward optimal active learning through Monte Carlo estimation of error reduction, in: Proceedings of the International Conference on Machine Learning (ICML), 2001, pp. 441–448.

[19] Y. Freund, H.S. Seung, E. Shamir, N. Tishby, Selective sampling using the query by committee algorithm, Mach. Learn. 168 (1997) 133–168, doi:10.1023/A:1007330508534.

[20] Q. Zhang, S. Sun, Multiple-view multiple-learner active learning, Pattern Recognit. 43 (2010) 3113–3119, doi:10.1016/j.patcog.2010.04.004.

[21] Y. Baram, R. El-Yaniv, K. Luz, Online choice of active learning algorithms, J. Mach. Learn. Res. 5 (2004) 255–291, doi:10.1017/CBO9781107415324.004.

[22] P. Auer, N. Cesa-Bianchi, Y. Freund, R.E. Schapire, Gambling in a rigged casino: the adversarial multi-armed bandit problem, in: Proceedings of the 36th Annual Symposium on foundations of Computer Science, 1995, pp. 322–331.

[23] D. Shen, J. Zhang, J. Su, G. Zhou, C.-L. Tan, Multi-criteria-based active learning for named entity recognition, in: Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL04), 2004, pp. 589–596, doi:10.3115/1218955.1219030.

[24] S. Patra, K. Bhardwaj, L. Bruzzone, A spectral-spatial multicriteria active learning technique for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (2017) 5213–5227, doi:10.1109/JSTARS.2017.2747600.

[25] B. Demir, C. Persello, L. Bruzzone, Batch-mode active-learning methods for the interactive classification of remote sensing images, IEEE Trans. Geosci. Remote Sens. 49 (2011) 1014–1031, doi:10.1109/TGRS.2010.2072929.

[26] S. Patra, L. Bruzzone, A novel SOM-SVM-based active learning technique for remote sensing image classification, IEEE Trans. Geosci Remote Sens. 52 (2014) 6899–6910, doi:10.1109/TGRS.2014.2305516.

[27] B. Demir, L. Bruzzone, A multiple criteria active learning method for support vector regression, Pattern Recognit. 47 (2014) 2558–2567, doi:10.1016/j.patcog.2014.02.001.

[28] C.T. Symons, N.F. Samatova, R. Krishnamurthy, B.H. Park, T. Umar, D. Buttler, T. Critchlow, D. Hysom, Multi-criterion active learning in conditional random fields, in: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI06, IEEE, 2006, pp. 323–331, doi:10.1109/ICTAI.2006.90.

[29] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, D. Tao, Exploring representativeness and informativeness for active learning, IEEE Trans. Cybern. 47 (2017) 14–26, doi:10.1109/TCYB.2015.2496974.

[30] A. Samat, P. Gamba, S. Liu, P. Du, J. Abuduwaili, Jointly informative and manifold structure representative sampling based active learning for remote sensing image classification, IEEE Trans. Geosci. Remote Sens. 54 (2016) 6803–6817, doi:10.1109/TGRS.2016.2591066.

[31] X. Xu, J. Li, S. Li, Multiview intensity-based active learning for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 56 (2018) 669–680, doi:10.1109/TGRS.2017.2752738.

[32] J. Zhang, D. Chen, H. Xie, S. Zhang, L. Gu, I dont know: double-strategies based active learning for mammographic mass classification, in: Proceedings of the Life Sciences Conference (LSC), 2017 IEEE, IEEE, 2017, pp. 182–185, doi:10.1109/LSC.2017.8268173.

[33] S. Lin, Rank aggregation methods, Wiley Interdiscip. Rev. Comput. Stat. 2 (2010) 555–570, doi:10.1002/wics.111.

[34] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, Q. Tian, Query-adaptive late fusion for image search and person re-identification, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, pp. 1741–1750, doi:10.1109/CVPR.2015.7298783.

[35] T.A. Solgård, P. Landskroener, Municipal voting system reform: overcoming the legal obstacles, Choice 1 (2002) 3rd.

[36] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 613–622.

[37] H.P. Young, Condorcets theory of voting, Am. Political Sci. Rev. 82 (1988) 1231–1244, doi:10.2307/1961757.

[38] S. Niu, Y. Lan, J. Guo, X. Cheng, Stochastic rank aggregation, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2013, pp. 478–487.

[39] M. Kendall, Rank correlation methods., 1948.

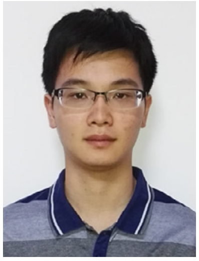
[40] P. Diaconis, R.L. Graham, Spearmans footrule as a measure of disarray, J. R. Stat. Soc. Ser. B (Methodological) 39 (1977) 262–268, doi:10.2307/2984804.

[41] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. (2001) 45–66, doi:10.1162/153244302760185243.

[42] HT Nguyen, A Smeulders, Active learning using pre-clustering, in: Proceedings of the Twenty-first International Conference on Machine Learning, ACM, 2004, p. 79, doi:10.1145/1015330.1015349.

Yu Zhao received his Master's degree with honors in biomedical engineering at the University of Shanghai for Science and Technology in 2014. Currently, he is now pursuing the Ph.D. degree at Shanghai Jiao Tong University. His current research interests include Computer aided diagnosis, image processing, and active learning.

Zhenhui Shi received the Bachelor's degree in University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently working toward the Master degree in Shanghai Jiao Tong University, Shanghai, China.



Jingyang Zhang received the B.S. degree in Biomedical Engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently pursuing the M.S. degree with the school of Biomedical Engineering, Shanghai Jiaotong University, Shanghai, China. His current research interests include active learning and computer vision.

Dong Chen received the M.S. degree in Measurement Technique and Automation Equipment from the Hebei University, Baoding, China, in 2010. He is currently pursuing the D.S. degree with the School of Biomedical Engineering, Shanghai Jiaotong University, Shanghai, China. His current interests include Virtual Reality, Respiratory Motion Modeling and Machine Learning.

Lixu Gu received the Ph.D. degree in computer science from the Toyohashi University of Technology, Toyohashi, Japan, in 1999. He was with Robarts Research Institute, ON, Canada, for three years, where he was responsible for the research and software development of the medical image analysis. In 2003, he moved to Shanghai to join the Research Group in Biomedical Engineering of Shanghai Jiao Tong University, Shanghai, China, His research interests include pattern recognition, computer vision, medical image processing, computer graphics, virtual reality, image guided surgery and therapy and medical robotics. He is the author of more than 200 papers on related research area.