Contents lists available at ScienceDirect

Pattern Recognition

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

An active learning-based SVM multi-class classification model

Husheng Guo [a], Wenjian Wang [a,b,*]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan, 030006 Shanxi, China

[b] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, 030006 Shanxi, China

ARTICLE INFO

Article history:
Received 21 May 2014
Received in revised form
18 November 2014
Accepted 6 December 2014
Available online 18 December 2014

Keywords:
Multi-class classification with unknown categories
Active learning
Support vector machine
MC_SVMA model

ABSTRACT

Traditional multi-class classification models are based on labeled data and are not applicable to unlabeled data. To overcome this limitation, this paper presents a multi-class classification model that is based on active learning and support vector machines (MC_SVMA), which can be used to address unlabeled data. Firstly, a number of unlabeled samples are selected as the most valuable samples using the active learning technique. And then, the model quickly mines the pattern classes for unlabeled samples by computing the differences between the unlabeled and labeled samples. Moreover, to label the unlabeled samples accurately and acquire more class information, the active learning strategy is also used to select compatible, rejected and uncertain samples, which are labeled by experts. Thus, the proposed model can determine as many classes as possible while requiring fewer samples to be manually labeled. This approach permits an unlabeled multi-classification problem to be translated into a classical supervised multi-classification problem. The experimental results demonstrate that the MC_SVMA model is efficient and exhibits good generalization performance.

1. Introduction

In real-world applications, there are many multi-class classification problems with unknown categories. Due to the rapid growth in webpage data, multi-class classification approaches are needed for automatic webpage annotation. However, it is impossible to obtain labels for all webpages, and the attainable labels may not contain all possible cases. As an example, labeled webpages may only contain two categories (political and economic), but the categories of unlabeled webpages are usually uncertain. Additionally, disease diagnosis is another multi-class classification case with unknown classes. Similarly, we can only label the vital signs of existing diseases. When the vital signs from an incoming patient are different from the statistical data, we must determine whether a new disease is occurring and if a new pattern should be built. However, it is not realistic for experts to label all of the training samples one by one. Several reasons for this include (1) the labeling cost is always high for large amounts of data, (2) the problem itself may be not suitable for large scale labeling, such as fault detection (especially dud detection in the military), (3) if only a part of the unlabeled samples are extracted for experts to label, we must know which ones are important or how many samples are enough to be worthwhile, and (4) experts may not directly choose which samples to label if the samples are provided by stream mode. Therefore, efficient approaches to solve these problems need to be designed.

Although many multi-class classification approaches, such as One-Versus-One (OvO), One-Versus-Rest (OvR), Directed Acyclic Graph (DAG) and Global Optimal Classification (GOC) algorithms, have been presented, they are limited to solving multi-class classification problems containing known categories [1–4]. If the classes are not identified before training, typical multi-class classification approaches cannot be applied directly. Although the unsupervised clustering technique is a way to obtain the categories, the evaluation of clustering validity is highly important and is a difficult problem. Additionally, clustering for imbalanced data is another serious issue that must be dealt with. Hence, clustering is more suitable for data analysis, not for classification. Manual intervention may be a better choice. Active learning is such a strategy, which selects certain data as the most valuable samples to be labeled by experts. It reduces the labeling cost in a complementary way by querying the labels of the most informative points. Thus, instead of being a passive recipient of the data to be processed, the active learner can control what data needs to be added to the training dataset. In this way, high classification accuracy can be achieved for only few labeled samples, translating the above problem into a traditional multi-class classification problem.

This paper presents a novel multi-class classification approach based on SVM active learning (MC_SVMA). This approach is not a simple combination of the active learning and the SVM multi-class

[*] Corresponding author at: School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China.
Tel.: +86 351 7017567; fax: +86 351 7018176.
E-mail address: wjwang@sxu.edu.cn (W. Wang).

http://dx.doi.org/10.1016/j.patcog.2014.12.009
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

classification approach but allows the former to participate during the entire learning processes, including the initial pattern classes mining and subsequent SVM training. The proposed MC_SVMA model can quickly mine the pattern class in the unlabeled samples by computing the *Discrepancy* between the unlabeled and labeled samples. Next, the problem can be translated into a traditional multi-class classification problems and the initial hyperplane can be obtained by SVM learning. Because the rejected, compatible and uncertain samples are generally difficult to classify, three measurement factors *Rejection*, *Compatibility* and *Uncertainty* are defined to determine which samples will serve as the most valuable samples and be labeled by experts. Located areas of these three types of samples are referred to as classification blind areas (*CBA*), classification compatible areas (*CCA*) and classification uncertain areas (*CNA*), respectively. These areas are introduced by analyzing the relationships between the unlabeled samples and the obtained approximate hyperplanes. After extracting a small part of the most valuable samples in the *CBA*, *CCA* and *CNA* to label, all of the rejected, compatible and uncertain samples can be classified effectively. In this way, the convergence speed and generalization performance can be improved synchronously.

We begin by presenting the background knowledge, including the active learning and multi-class classification models in Section 2. In Section 3, we describe how to mine the pattern classes based on the *Discrepancy*, introduce three the most valuable samples extraction techniques based on *Rejection*, *Compatibility* and *Uncertainty*, and summarize the proposed MC_SVMA algorithm. In Section 4, we simulate experiments and discuss the proposed model with regards to efficiency and performance. In the last section, we present this work's conclusions and discuss future research.

2. Background knowledge

2.1. Active learning

Active learning [5], first presented by Simon in 1974, is an effective machine learning method that originated to solve unlabeled binary classification problems. The basic function is to select the most valuable samples, label them by experts and add them to the training set. This process is executed iteratively through a certain number of loops. Presently, active learning has been used in many fields, including image classification [6], target detection [7] and text classification [8], which has become a research hotspot in the machine learning field.

Because the most valuable samples extracted by active learning directly affect learning efficiency and generalization performance, the most crucial step for active learning is designing the rules for selecting the most valuable samples. Most early classical studies about active learning focused mainly on binary classification tasks, such as the active binary classification methods that extract the most valuable samples based on various criteria: measuring the distance between a hyperplane and samples [9], derived importance weight [10], degree of uncertainty [11], Query-by-Committee (QBC) [12–15], neural network [16] or decision tree [17].

SVM introduced by Vapnik [18] is an effective method to solve pattern recognition and regression problems such as hand-written digit recognition, face image recognition and time series prediction. Similarly, one key of SVM active learning is designing the most valuable samples extraction technique. Fig. 1 shows an intuitive example in which the most valuable samples directly affect the performance of the final classifier. The samples denoted by the black solid triangles and the black solid circle in Fig. 1(b) are more significant for obtaining the good classification results than those in Fig. 1(a). In other words, if we can select some "good" samples to label, we can obtain optimal classifiers for all unlabeled samples at a very low labeling cost.

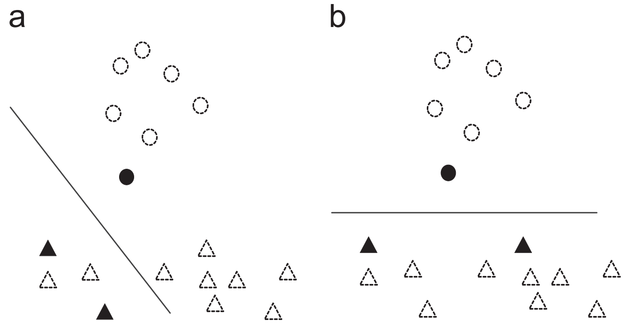

Fig. 1. Extraction of the most valuable samples. (a) Unreasonable the most valuable sample extraction and (b) reasonable the most valuable sample extraction.

Generally, if the unlabeled samples near the initial approximate hyperplane are selected, the hyperplane after retraining may be slightly changed. On the contrary, if the unlabeled samples far away from the hyperplane are selected, the hyperplane may be drastically improved. Therefore, the initial extraction of the most valuable samples is extremely important for SVM active learning. Recently, many SVM active methods are being designed by effective extraction of the most valuable samples for expert labeling, which effectively improves the efficiency and accuracy of binary classification [19–22].

2.2. Multi-class classification models

For a multi-class classification problem, assume first that the training dataset is $X=\{(x_i,y_i)\}_{i=1}^{l}$, $x_i\in R^n$ and $y_i\in\{1,2,\cdots,c\}$. For the existing multi-class classification models, the OvR model is the most commonly used [23]. Samples belonging to the jth class are labeled with $+1$ and all other samples are labeled with -1, and one can subsequently obtain c classifiers, and the jth classifier is determined by solving the following quadratic optimization problem:

$$\min \tfrac{1}{2}\|w^j\|^2+C\left(\sum_{i=1}^{l}\xi_i^j\right)$$
$$\text{s.t.}\begin{cases}(w^j\Phi(x_i))+b^j\geq 1-\xi_i^j, if(y_i=j)\\(w^j\Phi(x_i))+b^j\leq 1-\xi_i^j, if(y_i\neq j)\\\xi_i^j\geq 0\\i=1,2,\cdots,l\end{cases}. \tag{1}$$

According to formula (1), one can solve the c quadratic optimization problems and obtain c decision functions. When we predict an unlabeled sample x, we need only compute c function classifiers values. The corresponding class maxima and positive distinguishing function value is selected to label the sample x (see formula (2)):

$$y_c\equiv\arg\max_{j=1,\cdots,c} f_j(x). \tag{2}$$

Because the OvR needs to train only c binary classifiers, and the number of classifiers is much smaller than training samples, it is considered as a type of simple model with a fast testing speed. However, it may be unbounded due to some samples being incorrectly classified. For example, samples located in some areas may not belong to any class (here the "area" is referred to as a *classification blind area*, *CBA* – see Fig. 2(a)), or belong to different classes at same time (we call this a *classification compatibility area*, *CCA* – see Fig. 2(b)), or even be unsure as to what type of classes they belong to (for the unlabeled sample x in this area, there exists a classifier f_j such that the distance between the sample x and f_j is

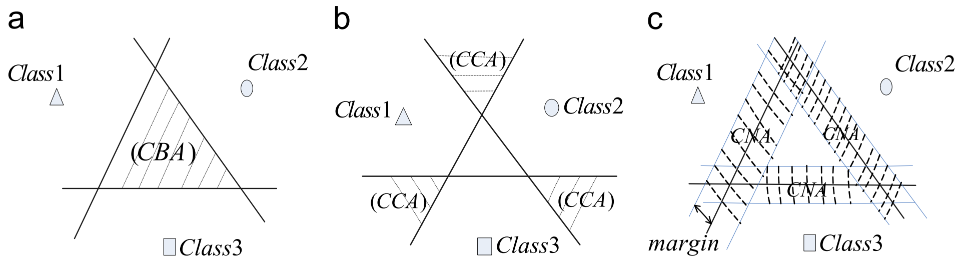

Fig. 2. The three areas where samples cannot be classified correctly. (a) *CBA*, (b) *CCA* and (c) *CAN*.

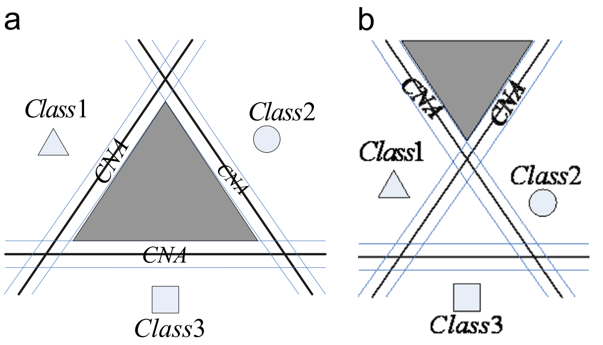

Fig. 3. The difference region between *CBA* (*CCA*) and *CAN*. (a) Samples in *CBA* not belonging in *CNA* and (b) Samples in *CCA* not belonging in *CNA*.

smaller than the $margin_j$; this area is denoted as a *classification uncertain area*, *CNA* – see Fig. 2(c)). Although some of the most valuable samples may locate in the intersection of regions *CBA* and *CNA*, many samples in *CBA* may be not in *CNA* (See the shadow region in following Fig. 3(a)). Similarly, some of the most valuable samples may belong in the intersection of regions *CCA* and *CNA*, but many of them in *CCA* do not locate in *CNA* (See the shadow region in following Fig. 3(b)). At present, some existing SVM-based active learning techniques can select the most valuable samples from the *CNA* region by incorporating suitable diversity criteria [24,25], but there are no effective methods to distinguish between samples in *CBA* and *CCA*.

Additionally, there are also other typical multi-class classification models. The OvO model constructs all the possible $N = c(c-1)/2$ binary classifiers [26]. The number of classifiers obtained by the OvO model is often too large, and the model is too complex. This model is therefore not suitable for solving multi-class classification problems containing many classes. On the basis of the OvO model, the DAG model decides the category of a new sample by building a decision tree [27]. The GOC model obtains *c* decision functions by solving a quadratic optimization problem which may be too complex for large-scale problems [4]. Other multi-class classification models are generally improvements upon these traditional models [28–30].

2.3. Multi-class classification models based on active learning

In recent years, there have been many research studies of multi-class classification models based on active learning, such as the multi-class classification active learning method that extracts the most valuable samples by margin-based disagreement, uncertainty sampling-based disagreement, or specific disagreement [31]. Furthermore, converting an active multi-class classification problem into a series active binary classification problem is also an efficient path [32].

Some scholars have developed SVM active multi-class classification methods. For example, Patra et al. [24] presented the novel batch-mode active learning technique for solving multi-class classification problems by using the SVM classifier with the multi-class classification method. The uncertainty of each unlabeled sample is measured by defining criteria that not only consider the smallest distances to the decision hyperplanes but also take into account the distances to other hyperplanes. Chen et al. [25] presented a novel multi-class classification algorithm for music annotation problems. This method can select multiple most valuable samples in each iteration process, and it solves problems like reducing redundancy and avoiding selecting outliers within the selected examples. Although these methods can solve classical supervised multi-class classification problems, they cannot be applied immediately to multi-class classification problems with unknown classes. To the best of our knowledge, obtaining accurate category labels at an acceptable cost has not yet been discussed.

3. SVM active multi-class classification model

At present, the majority of the research on SVM active multi-class classification learning focuses on extracting the most valuable samples in *CNA*. Work on the most valuable samples extraction in *CBA* and *CCA* is still lacking. Additionally, most current methods are focused on multi-class classification with known categories. Corresponding effective methods for multi-class classification problems with unknown categories are missing. To address these needs, this paper presents an effective SVM active multi classification model. The basic task of it is to select samples that are as good as possible to label at the lowest labeling cost. According to the labeled samples, one can obtain the initial class information. Next, the final classifiers are obtained after classifying the most valuable samples into the *CBA*, *CCA* and *CNA* categories.

3.1. Pattern class mining

For unlabeled data, the categories are not acknowledged before training, and research has been performed on mining pattern classes. Obviously, the random sampling method is a simple and direct way for sample labeling. Because more samples may need to be selected to obtain class information, this method is low in mining efficiency and high in labeling cost, especially for large-scale and complicated problems. Clustering provides another way to select samples. The important samples in each cluster (the center of the clusters) can be identified served as the most valuable samples for labeling. However, the optimal class mining result may be not obtained due to bottlenecks in the clustering method itself. Currently, certain of the machine learning methods for a variety data mining tasks are based on farthest-first traversal for the purpose of extracting the important information that is in large datasets. For example, Basu et al. pick the disjoint neighborhoods that are obtained from clustering by farthest-first traversal to solve semi-supervised clustering tasks [33]. In this paper, we use farthest-first-traversal technology, active learning, to mine initial pattern classes. A factor referred to as the *Discrepancy*

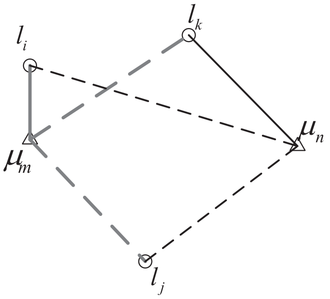

Fig. 4. A large *Discrepancy* may easily lead to the identification of a new class.

measures the difference between an unlabeled sample and labeled sample as follows:

Definition 1 (Discrepancy). The *Discrepancy* of the unlabeled sample x_j is defined as:

$$Discrepancy(x_j) = \min_{x_s \in Label\_set} d(x_j, x_s). \quad (3)$$

where *Label_set* is the set of labeled samples and $d(x_j, x_s)$ is the Euclidean distance between the samples x_j and x_s. If the *Discrepancy* of an unlabeled sample is large (or exceeds a given threshold), it shows that the sample is far from each labeled sample and may serve as a new pattern class. Fig. 4 provides an intuitive explanation. In Fig. 4, the labeled sample set is $Label\_set = \{l_i, l_j, l_k\}$ and the unlabeled sample set is $Unlabel\_set = \{u_m, u_n\}$. According to Definition 1, $Discrepancy(u_m)$ is equal to $d(u_m, l_i)$ and $Discrepancy(u_n)$ is equal to $d(u_n, l_k)$. Because $d(u_m, l_i) < d(u_n, l_k)$, it is more likely to produce a new class using u_n rather by using u_m.

Because computing the *Discrepancy* of an unlabeled sample requires at least two labeled samples, the proposed pattern class mining algorithm (PM_D) includes two parts: (a) The extraction of the initial two samples to be labeled by experts and (b) the extraction of as many classes as possible contained in the whole dataset.

The PM_D algorithm is summarized as follows. In Algorithm 1, *Label_set* is the set of labeled samples, *Unlabel_set* is the set of unlabeled samples, *Class_set* is the set of pattern classes and $\overline{Step}$ is the iteration loop number. The variable $\overline{Step}$ may affect the number of mining classes, which directly affects the final learning efficiency. The influence of $\overline{Step}$ is discussed in the specifically in the experimental section.

Algorithm 1. Pattern class mining based on *Discrepancy* (PM_D)

```
Initialize:$X = \{x_i\}_{i=1}^{l}$, $Label\_set = \phi$, $Unlabel\_set = X$,
  $Class\_set = \phi$, and the given iteration loops $\overline{Step}$.
Part1: Extract initial two samples.
Step1: Compute the center $\mu$ from the samples in the
  $Unlabel\_set$.
            $\mu = (1/l)\sum_{p=1}^{l} x_p$
Setp2: Extract the first sample.
      Setp 2.1 $x_1^* = \arg\max_{x_s \in Unlabel\_set} d(x_s, \mu)$.
      Setp 2.2 Label the sample $x_1^*$ using experts and obtain
  the label $y_1^*$ of $x_1^*$.
      Setp 2.3 $Label\_set = Label\_set \cup \{x_1^*\}$,
  $Unlabel\_set = Unlabel\_set \backslash \{x_1^*\}$,
      Step2.4 $Class\_set = Class\_set \cup \{y_1^*\}$.
Step3: Extract the second sample.
      Step3.1 $x_2^* = \arg\max_{x_s \in Unlabel\_set} d(x_s, x_1^*)$.
      Step3.2 Label the sample $x_2^*$ using experts and obtain
  the label $y_2^*$ of $x_2^*$.
      Step3.3 $Label\_set = Label\_set \cup \{x_2^*\}$,
  $Unlabel\_set = Unlabel\_set \backslash \{x_2^*\}$.
      Step3.4 If $(y_2^* \neq y_1^*)$
                  $Class\_set = Class\_set \cup \{y_2^*\}$
Part2: Mine other pattern classes
Step4: Mine other pattern classes and update $Label\_set$,
  $Unlabel\_set$ and $Class\_set$.
      Step4.1: $Step = 0$.
      Setp4.2: $x_o^* = \arg\max_{x_s \in Unlabel\_set} Discrepancy(x_s)$.
      Step4.3: Label the sample $x_o^*$ using experts and obtain
  the label $y_o^*$ of $x_o^*$.
                  $Label\_set = Label\_set \cup \{x_o^*\}$,
  $Unlabel\_set = Unlabel\_set \backslash \{x_o^*\}$.
      Step4.4: if $(y_o^* \in Class\_set)$
                  $Step = Step + 1$
              else $\{Step = 0, Class\_set = Class\_set \cup \{y_o^*\}\}$
      Setp4.5: if $(Step \leq \overline{Step})$
                  go to Step4.2.
Step5: End the algorithm and obtain the $Label\_set$, $Unlabel\_set$
  and $Class\_set$.
```

In Algorithm 1, samples are selected using the *Discrepancy* instead of random sampling, which provides a way to identify as many classes as possible using as few samples as possible.

To test the performance of PM_D, the pattern class mining based on a random method (PM_R) and clustering method (PM_C) (samples that are the nearest to the centers of clusters being selected to be labeled) are compared.

The PM_R method is summarized as follows.

Algorithm 2. Pattern class mining algorithm based on random selection (PM_R)

```
Initialize: $X = \{x_i\}_{i=1}^{l}$, $Label\_set = \phi$, $Unlabel\_set = X$,
  $Class\_set = \phi$, and the given iteration loops $\overline{Step}$.
Step1: Select a sample $x_i$ randomly from the initial unlabeled
  samples set.
Step2: Label the sample $x_i$ using experts and obtain the label $y_i$.
      $Label\_set = Label\_set \cup \{x_i\}$,
  $Unlabel\_set = Unlabel\_set \backslash \{x_i\}$.
Step3: if $(y_i \in Class\_set)$
            $Step = Step + 1$
      else
            $\{Step = 0, Class\_set = Class\_set \cup \{y_i\}\}$.
Step4: if $(Step \leq \overline{Step})$
            go to Step1.
Step5: End the algorithm and obtain the $Label\_set$, $Unlabel\_set$
  and $Class\_set$.
```

The PM_C method is summarized as follows.

Algorithm 3. Pattern class mining algorithm based on clustering (PM_C)

```
Initialize: $X = \{x_i\}_{i=1}^{l}$, $Label\_set = \phi$, $Unlabel\_set = X$,
  $Class\_set = \phi$, the given iteration loops $\overline{Step}$, and the given the
  initial clustering parameter $k$.
Step1: Divide $X$ into $k$ categories $\{X_1, \cdots X_i, \cdots, X_k\}$.
Step2: Compute the center $\mu_i$ for each $X_i$.
Step3: Compute the distance $d(x_{i_j}, \mu_i)$ between sample $x_{i_j}$ of $i$th
  cluster and the center $\mu_i$, and select
  $x_i^* = \arg\min_{x_{i_j} \in X_i} Discrepancy(x_{i_j}, \mu_i)$ so that $X^* = \{x_i{}^*, i = 1, \cdots, k\}$
  can be constructed.
```

```
Step4: Label all the samples in $X^*$ and obtain the corresponding
  categories $Y^* = \{y_i^*, i = 1, \cdots, k\}$.
            $Label\_set = Label\_set \cup X^*$,
  $Unlabel\_set = Unlabel\_set \backslash X^*$
Step5: If $(Y^* \subseteq Class\_set)$
            $Step = Step + 1$.
       else
            $\{Step = 0, Class\_set = Class\_set \cup Y^*\}$.
Step6: If $(Step \le \overline{Step})$
            $k + 1 \rightarrow k$, and goto Step1.
       else
            { End the algorithm and obtain the $Label\_set$,
  $Unlabel\_set$ and $Class\_set$}
```

3.2. The most valuable samples extraction

Many effective approaches for extracting the most valuable samples for active learning have been presented in [2,6–9,13–14,19,26]. They are based on two ideas: extracting uncertain samples (such as those near the hyperplane) as the most valuable samples, and selecting the most valuable samples by some evaluation criteria determined by the committee. Either way, these can effectively classify difficult samples only into *CNA* because *CBA* and *CCA* are not available for binary classification problems. Therefore, the means to extract the most valuable samples for multi-class classification problems is worth further research.

As mentioned in Section 2.2, for multi-class classification problems, those samples in three *CBA*, *CCA* and *CNA* are difficult to classified and are most likely to be the most valuable samples. These samples may be many in number, and the labeling costs will increased if all these samples are selected. Hence, only a few part of rejected samples, compatible samples and uncertain samples located in *CBA*, *CCA* and *CNA* should be serve as the most valuable samples for active learning, and some new classes that not be mined in Section 3.1 may be found as well.

3.2.1. The most valuable samples extraction in CBA

Suppose the set of a series of classifiers $F = \{f_1, f_2, \cdots, f_c\}$ is obtained after the initial pattern class mining using Algorithm 1. The *Rejection* of an unlabeled sample in *CBA* is defined first.

Definition 2 (Rejection). The *Rejection* of an unlabeled sample x_i in the *CBA* is:

$$Rejection(x_i) = \frac{1}{\sqrt{\sum_{j=1}^{c}\left(d\left(x_i, f_j\right) - \left(\sum_{j=1}^{c} d(x_i, f_j)/c\right)\right)^2}}. \tag{4}$$

where $d(x_i, f_j)$ denotes the distance between x_i and the hyperplane f_j. For an extreme case, if all the $d\left(x_i, f_j\right)$ are equal to the average value $\sum_{j=1}^{c} d(x_i, f_j)/c$, a sufficiently large value is assigned to $Rejection(x_i)$.

If the *Rejection* of a sample is large, then the distances between this sample and all hyperplanes $f_1, f_2, \cdots, f_c$ obtained by *Label_set* are close. Otherwise, the sample must be closer to at least one of the hyperplanes $f_1, f_2, \cdots, f_c$, meaning that a sample with a large *Rejection* will be more difficult to classify into one of known classes, or may produce a new class. Fig. 5 presents is a simple example. In Fig. 5, the labeled dataset is $Label\_set = \{l_i, l_j, l_k\}$, the unlabeled dataset is $Unlabel\_set = \{\mu_m, \mu_n\}$, and f_i, f_j, f_k are obtained as approximate hyperplanes based on the labeled samples. The distances between μ_m and f_i, f_j, f_k are similar, but the distances between μ_n and these three hyperplanes are different (μ_n is the closest to f_j). According to Definition 2, we may determine that $Rejection(\mu_m)$ is larger than $Rejection(\mu_n)$. Because the standard deviation of the distance between μ_m and the three hyperplanes is smaller than that of μ_n, the probability of $y(\mu_m) \notin Class\_set$ is larger than that of $y(\mu_n) \notin Class\_set$. In other words, μ_m is more likely to contribute to mining a new class. However, if $y(\mu_m) \in Class\_set$ and $y(\mu_n) \in Class\_set$, when μ_m is extracted as a most valuable sample, the updating range for the hyperplane may be larger than that of μ_n (see Fig. 6). Hence, the classification information for μ_m is greater than that of μ_n. Therefore, the samples in the *CBA* with a larger *Rejection* should be extracted as the most valuable samples.

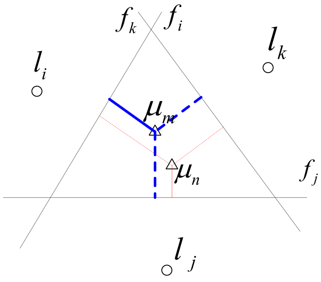

Fig. 5. Rejected samples.

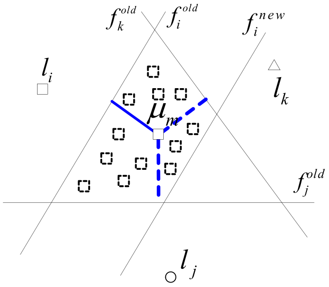

Fig. 6. Hyperplane updating based on a large *Rejection*

On the basis of this analysis, the sample that is nearest to the center of the *CBA* should be selected as the most valuable sample because its *Rejection* is the biggest. By doing so, a new category may be produced, or the corresponding hyperplane may be greatly adjusted. The algorithm for classifying rejected samples is summarized as follows.

Algorithm 4. Classify rejected samples in the CBA

```
Initialize: $Label\_set$, $Unlabel\_set$, $Class\_set = \{1, 2, \cdots, c\}$, K: RBF
  kernel.
Step1: Obtain initial classifiers.
            Train SVM using all samples in $Label\_set$
  based on OvR method and obtain the initial classifier set:
  $F = \{f_1, f_2, \cdots, f_c\}$.
Step2: Construct $Blind\_set = \{x_i | x_i \in Unlabel\_set$, for each $f_j \in F$
  such that $f_j(x_i) < 0\}$.
Step3: Select the most valuable rejected samples and retrain
  SVM.
        Step3.1: $x_r^* = \arg \max_{x_i \in Blind\_set} Rejection(x_i)$.
        Step3.2: Label the sample $x_r^*$ using experts and obtain
  the label $y_r^*$ of $x_r^*$.
            $Label\_set = Label\_set \cup \{x_r^*\}$,
            $Unlabel\_set = Unlabel\_set \backslash \{x_r^*\}$,
            $Blind\_set = Blind\_set \backslash \{x_r^*\}$,
            If $(y_r^* \notin Class\_set)$
            $\{Class\_set = Class\_set \cup \{y_r^*\}\}$
        Step3.3: Retraining SVM on the new $Label\_set$ and
    obtain the new F.
        Step3.4: If ($Blind\_set \neq \phi$)
                Goto Step2.
  Step4: End the algorithm and obtain the final F.
```

From Algorithm 4, $Blind\_set = \phi$ denotes that most of the samples in the *CBA* are dispelled and only a very small number of the rejected samples in the *CBA* need to be labeled. This shows that active learning can correctly classify all the unlabeled samples for a very low labeling cost.

3.2.2. The most valuable samples extraction in CCA

As mentioned in Section 2.2, samples in the *CCA* may be classified in several categories simultaneously. Hence, they should also be taken as the most valuable samples and labeled by experts. To determine the most valuable samples in the *CCA*, a criterion is defined to measure the compatibility of an unlabeled sample as follows.

Definition 3 (Compatibility). Suppose the classifiers $f_1, f_2, \cdots, f_c$ are obtained from the labeled samples, and an unlabeled sample x_i in the *CCA* belongs to c' categories ($c' \le c$). The compatibility of the sample x_i is defined as:

$$Compatibility(x_i) = \frac{1}{\sqrt{\sum_{c'} \left(d(x_i, f_k) - \sum_{c'} d(x_i, f_j)/c' \right)^2}}. \qquad (5)$$

where $d(x_i, f_k)$ is the same as in Eq. (4). Similarly, if all the $d(x_i, f_k)$ are equal to the average value $\sum_{c'} d(x_i, f_j)/c'$, a sufficiently large value is assigned to $Compatibility(x_i)$.

If the *Compatibility* of a sample is large, then the differences in the various distances between this sample and its corresponding hyperplanes are small, which means the close degree of the sample to corresponding hyperplanes are similar. Because those samples with large a *Compatibility* may be difficult to determine class information on, they should be extracted as the most valuable samples. Fig. 7 shows an intuitive example. In Fig. 7, the set of labeled samples is $Label\_set = \{l_i, l_j, l_k\}$, the set of unlabeled samples is $Unlabel\_set = \{\mu_m, \mu_n\}$, and f_i, f_j, f_k are the obtained approximate hyperplanes. Obviously, $d(\mu_m, f_i) \approx d(\mu_m, f_k)$ and $d(\mu_n, f_i) < d(\mu_n, f_j)$. According to Definition 3, $Compatibility(\mu_m)$ is larger than $Compatibility(\mu_n)$. Compared with μ_n, when μ_m is extracted as the most valuable sample, the updated range of hyperplanes may be larger. The convergence speed of the active learning process will be increased (see Fig. 8).

The classification algorithm for compatible samples is summarized as follows.

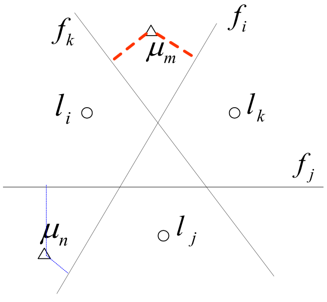

Fig. 7. Compatible samples.

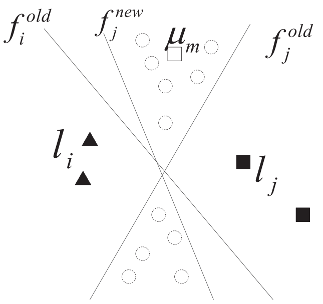

Fig. 8. Hyperplane updating based on the *Compatibility*

Algorithm 5. Classify compatible samples in *CCA*

```
  Initialize:$Label\_set$, $Unlabel\_set$, $Class\_set$, $Classifiers\_set$ F, K:
    RBF kernel.
  Step1: Construct $Compatibility\_set = \{x_i | x_i \in Unlabel\_set$, so
    there at least exists $f_m, f_n \in F$ such that $f_n(x_i) > 0$ and
    $f_m(x_i) > 0\}$.
  Step2: Extract the most valuable compatible samples and
    retrain SVM.
        Step2.1: $x_c^* = \arg\max_{x_i \in Compatibility\_set} Compatibility(x_i)$.
        Step2.2: Label the sample $x_c^*$ using experts and obtain
    the label $y_c^*$ of $x_c^*$.
                $Label\_set = Label\_set \cup \{x_c^*\}$,
                $Unlabel\_set = Unlabel\_set \backslash \{x_c^*\}$,
                $Compatibility\_set = Compatibility\_set \backslash \{x_c^*\}$,
                If ($y_c^* \notin Class\_set$)
                    $Class\_set = Class\_set \cup \{y_c^*\}$
        Step2.3: Retraining SVM on the new $Label\_set$ and
    obtain the new $F$.
        Step2.4: If ($Compatibility\_set \neq \phi$)
                Go to Step1.
  Step3: End the algorithm and obtain the final $F$.
```

Similarly to Algorithm 4, in Algorithm 5, there are only a small number of samples in the *CCA* that are labeled by experts, and all samples in the *CCA* will be removed automatically. Therefore, those unlabeled samples in the *CCA* can be classified with a very low labeling cost.

3.2.3. The most valuable samples extraction in CNA

Class information on those samples in the *CNA* may be uncertain, so that they should also be taken as the most valuable ones. To find the most valuable samples in the *CNA*, a criterion used to measure the uncertainty of an unlabeled sample in *CNA* is introduced as follows.

Definition 4 (Uncertainty). Suppose a classifier set $F = \{f_1, f_2, \cdots, f_c\}$ is obtained based on the labeled samples. The *Uncertainty* of an unlabeled sample x_i in the *CNA* is defined as:

$$Uncertainty(x_i) = \begin{cases} 0, & d(x_i, f_j) \ge margin_j \\ \frac{1}{\min_{j \in \{1,2,\cdots,c\}} (d(x_i, f_j))}, & d(x_i, f_j) < margin_j \end{cases}. \qquad (6)$$

where $d(x_i, f_j)$ is identical to that in Eq.(4). Similarly, if $\min_{j \in \{1,2,\cdots,c\}} (d(x_i, f_j)) = 0$, then the sample x_i is on the hyperplane f_j, and a sufficiently large value is assigned to the $Uncertainty(x_i)$. Generally, if the $Uncertainty(x_i)$ is large, then the distance of x_i to the nearest classifier f_j is small, meaning that the uncertainty that the sample belongs to the jth class (or other classes) is large. Therefore, this sample should be extracted as the most valuable sample.

Because different sample distributions in *CNA* influence the most valuable samples extracted using *Uncertainty*, some hyperplanes may be updated continuously, whereas others may never be updated (See Fig. 9). In Fig. 9, suppose $d(\{A_1, A_2, B_1, B_2\}, f_1) < margin_1$,

$d(\{B_3,C_1,C_2\},f_2) < margin_2$ and $d(D_1,f_3) < margin_3$, but $d(\{A_1,A_2,B_1,B_2\},f_1) \ll d(D_1,f_3)$ and $d(\{B_3,C_1,C_2\},f_2) \ll d(D_1,f_3)$, so $Uncertainty(\{A_1,A_2,B_1,B_2,B_3,C_1,C_2\}) \gg Uncertainty(D_1)$. If the most valuable samples are only extracted based on the *Uncertainty* directly, the classifiers f_1 and f_2 may be updated continuously but f_3 will not be updated. Moreover, the sequential implementation for hyperplane updating may lead to slow convergence.

To solve the unbalanced hyperplane-updating problem, an improved batch-mode method for most uncertain valuable sample extraction criteria, the “Round Robin” principle, is used. With this method, the extraction process for the most uncertain valuable sample is balanced and more efficient. To demonstrate this method accurately, an improved measure of uncertainty is defined to measure the *Uncertainty* of an unlabeled sample in the *CNA*, which relies on a classifier first.

Definition 5 (Improved Uncertainty). Suppose the classifier set $F=\{f_1,f_2,\cdots,f_c\}$ is obtained based on the labeled samples and the margin of jth classifier is $margin_j$. The *Uncertainty* of an unlabeled sample x_i on the jth hyperplane in *CNA* is redefined as:

$$Uncertainty(x_i,f_j)=\begin{cases}0, & d(x_i,f_j)\ge margin_j\\ \frac{1}{d(x_i,f_j)}, & d(x_i,f_j)< margin_j\end{cases}. \tag{7}$$

Based on the “Round Robin” principle, an algorithm for extracting the most valuable uncertain samples is presented and is summarized as follows.

Algorithm 6. Classify uncertain samples in the *CNA* based on the “Round Robin” principle

```
Initialize: $Label\_set$, $Unlabel\_set$, $Class\_set$, $Classifiers\_set$ F, K: RBF kernel.
Step1: Construct $Uncertainty\_set=\{x_i|x_i\in Unlabel\_set$, so that there exists some $f_j\in F$, such that the $Uncertainty(x_i,f_j)>0\}$ and
  $F'=\{f_j|f_j\in F$, so that there exists some $x_i\in Unlabel\_set$, such that $Uncertainty(x_i,f_j)>0\}$.
Step2: Extract the most valuable uncertain samples for every hyperplane.
    Step2.1: Construct $Uncer\_set=\{x_u|x_u=\arg\max_{x_i\in Uncertainty\_set,} Uncertainty(x_i,f_j), f_j\in F'\}$.
    Step2.2: Extract the most valuable uncertain samples of $f_j^*$ and retrain the SVM.
        Step2.2.1: $x_u^*=\arg\max_{x_u\in Uncer\_set} Uncertainty(x_u,f_j)$,
        Step2.2.2: Label the sample $x_u^*$ using experts and obtain the label $y_u^*$ of $x_u^*$,
            $Label\_set=Label\_set\cup\{x_u^*\}$,
            $Unlabel\_set=Unlabel\_set\backslash\{x_u^*\}$,
            $Uncer\_set=Uncer\_set\backslash\{x_u^*\}$,
            If $(y_u^*\notin Class\_set)$
            $Class\_set=Class\_set\cup\{y_u^*\}$,
        Step2.2.3: Retraining SVM on new $Label\_set$ and obtain the new $F$,
        Step2.2.4: If $(Uncer\_set\neq\phi)$
                Go to Step2.2.1.
    Step2.3: $Uncertainty\_set=Uncertainty\_set\backslash Uncer\_set$,
    Step2.4: If $(Uncertainty\_set\neq\phi)$
                Go to Step1.
Step3: End the algorithm and obtain the final classifiers $F$.
```

In Algorithm 6, there are $|Uncertainty\_set|$ uncertain samples near all of the classifiers, but only a few of them will be identified as the most valuable samples. Moreover, there are c' to $2c'$ hyperplanes that will be improved in a “Round Robin” process (c' is the class number of the *Uncer_set*). It means that at least two hyperplanes are adjusted for cases (3) and (4), and then the convergence speed will increase significantly. Additionally, the degree of each hyperplane adjustment is relatively large, which may also lead to increase the convergence speed. Fig. 10 shows an intuitive explanation of the proposed Algorithm 6. From Fig. 10, the samples that are denoted by triangles belong to the jth class (labeled by $+1$), and others are denoted by gray squares (labeled by -1) and belong to the other classes. Suppose that the initial obatained classifier is f_j^{old}, the margin of classifier f_j^{old} is $margin_j^{old}$, and the distance between x_u^* and the old hyperplane f_j^{old} is $+d(x_u^*,f_j^{old})$ or $-d(x_u^*,f_j^{old})$. The $+d(x_u^*,f_j^{old})$ represents that the sample x_u^* is on the positive side (the jth class) of the hyperplane f_j^{old}, i.e., $f_j^{old}(x_u^*)>0$. Similarly, $-d(x_u^*,f_j^{old})$ represents that sample x_u^* is on the negative side (the other class) of the hyperplane f_j^{old}, i.e., $f_j^{old}(x_u^*)<0$. Generally, $d(x_u^*,f_j^{old})\ll margin_j^{old}$. Then, x_u^* should be extracted as the most valuable uncertain sample.

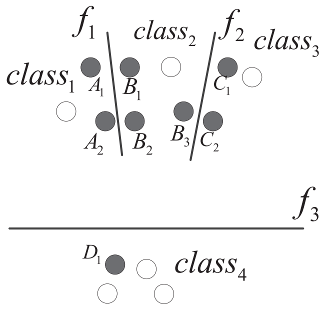

Fig. 9. Imbalanced hyperplanes updating.

(1). If $f_j^{old}(x_u^*)>0$ and $y_u^*=class_j$ (See Fig. 10(a)), then f_j^{new} can be obtained by shifting f_j^{old} in parallel. Those unlabeled samples that are located on the upper side (denoted by dashed circles) within $[+d(x_u^*,f_j^{old}),+margin_j^{old}/2]$ of hyperplane f_j^{old} are unnecessary to be labeled by the experts and can be classified into the jth class automatically. The hyperplane updating range is

$$\Delta_1(f_j^{new},f_j^{old})\approx\frac{d(x_u^*,f_j^{old})+margin_j^{old}/2}{2}-d(f_j^{old},x_u^*)$$
$$=\frac{margin_j^{old}-2d(x_u^*,f_j^{old})}{4}. \tag{8}$$

(2). If $f_j^{old}(x_u^*)>0$ and $y_u^*\neq class_j$ (See Fig. 10(b)), then the unlabeled samples that are located on the lower side (denoted with dashed circles) within $[-margin_j^{old}/2,+d(x_u^*,f_j^{old})]$ of hyperplane f_j^{old} are also unnecessary to be labeled because they cannot belong to the jth class. The hyperplane updating

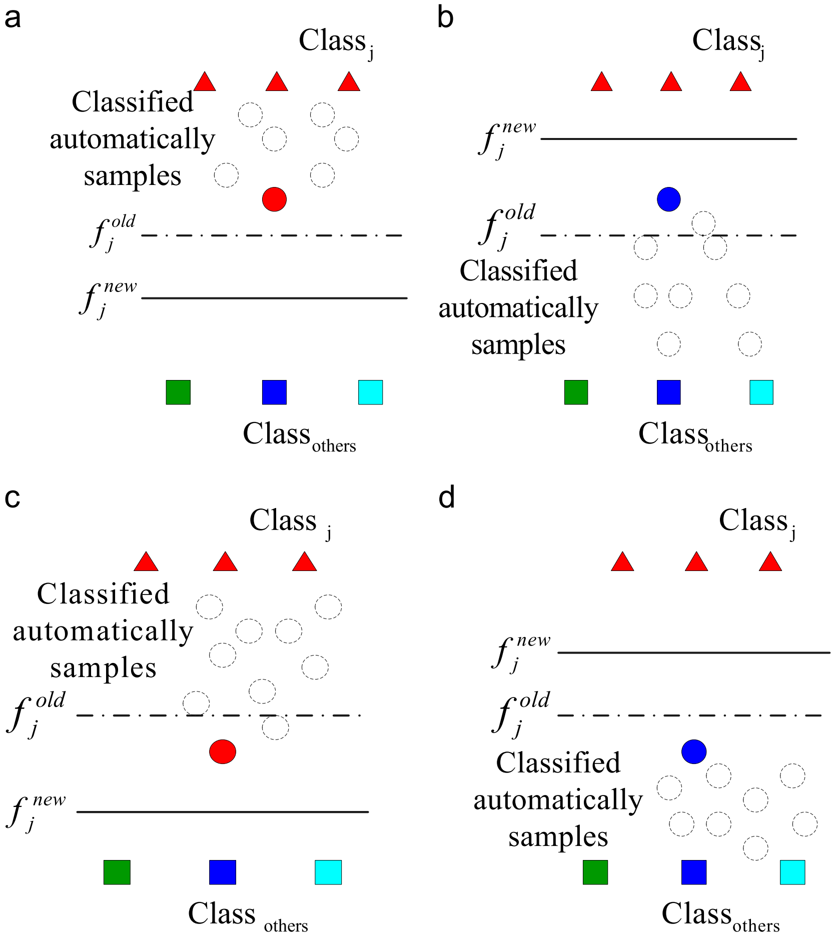

Fig. 10. Hyperplane updating based on the *Uncertainty* (a) $f_j^{old}(x_u^*) > 0$, $y_u^* = class_j$, (b) $f_j^{old}(x_u^*) > 0$, $y_u^* \neq class_j$, (c) $f_j^{old}(x_u^*) < 0$, $y_u^* = class_j$, (d) $f_j^{old}(x_u^*) < 0, y_u^* \neq class_j$.

range is

$$\Delta_2(f_j^{new}, f_j^{old}) \approx \frac{margin_j^{old}}{2} + d(x_u^*, f_j^{old}). \tag{9}$$

(3). If $f_j^{old}(x_u^*) < 0$ and $y_u^* = class_j$ (See Fig. 10(c)), then the unlabeled samples that are located on the upper side (denoted with dashed circles) within $[-d(x_u^*, f_j^{old}), +margin_j^{old}/2]$ of hyperplane f_j^{old} are also unnecessary to be labeled because they can be classified into the *j*th class automatically. The hyperplane updating range is

$$\Delta_3(f_j^{new}, f_j^{old}) \approx \frac{margin_j^{old}/2 - d(x_u^*, f_j^{old})}{2} + d(x_u^*, f_j^{old}) = \frac{margin_j^{old} + 2d(x_u^*, f_j^{old})}{4}. \tag{10}$$

(4). If $f_j^{old}(x_u^*) < 0$ and $y_u^* \neq class_j$ (See Fig. 10(d)), then the unlabeled samples that are located on the lower side (denoted with dashed circles) within $[-margin_j^{old}/2, -d(x_u^*, f_j^{old})]$ of hyperplane f_j^{old} are also unnecessary to be labeled because they cannot belong to the *j*th class. The hyperplane updating range is

$$\Delta_4(f_j^{new}, f_j^{old}) \approx \frac{margin_j^{old}}{2} - \frac{margin_j^{old}/2 + d(x_u^*, f_j^{old})}{2} = \frac{margin_j^{old} - 2d(x_u^*, f_j^{old})}{4}. \tag{11}$$

For practical problems, the hyperplane updates may not be parallel. Therefore, the above analysis is only an approximation, and $\Delta_1(f_j^{new}, f_j^{old})$, $\Delta_2(f_j^{new}, f_j^{old})$, $\Delta_3(f_j^{new}, f_j^{old})$ and $\Delta_4(f_j^{new}, f_j^{old})$ are greater than zero because $d(x_u^*, f_j^{old}) \ll margin_j^{old}$ such that the degree of each hyperplane update is relatively large.

3.3. MC_SVMA method

To solve the multi-class classification problems containing unknown classes, the proposed MC_SVMA algorithm first mines the possible pattern classes contained in unlabeled samples using the *Discrepancy*. Then, three factors (*Rejection*, *Compatibility* and *Uncertainty*) are used to extract the most valuable samples in the *CBA*, *CCA* and *CNA*, respectively. By analyzing Algorithm 4, it can be seen that the rejected valuable sample extracted by this algorithm is not always the most valuable sample in *CNA* because it has the largest rejection value and it may be near the center of *CNA*. Similarly, by analyzing Algorithm 5, it can be seen that the compatible sample extracted by this algorithm is not always the most valuable sample in *CNA*, because it has the largest compatibility value and it may be near the intersection region of two classifiers. Because the mining process in these three areas is not in fixed order, we testify the different mining process sequences in *CBA*, *CCA* and *CNA*. By classifying these certain difficult distinguished samples, good classification performance can be obtained. The process for the MC_SVMA algorithm is shown in Fig. 11.

The MC_SVMA algorithm contains two steps: initial pattern class mining and SVM active learning. The main steps in the proposed MC_SVMA algorithm are summarized as follows.

Algorithm 7. MC_SVMA algorithm

```
Initialize: Unlabeled training set $X = \{x_i\}_{i=1}^{l}$, K: RBF Kernel.
```

Step1: Mine the pattern classes based on Algorithm1 and obtain initial Label_set, *Class_set*, and the set of a series classifiers *F*.
Step2: Extract and classify the most valuable samples in CBA, CCA and CNA using Algorithms 4, 5 and 6.
Step3: Obtain the final multiple classifiers F.

3.4. Complexity analysis

The proposed MC_SVMA method includes two procedures: the pattern class mining and sample classification. Assume that the sample set size is l. In the first process (PM_D algorithm), the complexity of labeling the initial two categories samples is $O(l)$. Suppose that the iteration loops in the other pattern class process is m ($m<l$) and that the number of unlabeled samples is $l-i-1$ in the ith iteration loop, which implies that the complexity is $O(l-i-1)$ for the ith iteration loop. Therefore, the complexity of the PM_D algorithm is $O((2lm-m^2-3m)/2)+O(l)=O(lm)$. Because the complexity of the initial clustering processing of the PM_C algorithm is $O(l^2)$, it is not used to solve large-scale practical problems. However, PM_D and PM_R both have high learning efficiency and they can be used to solve large-scale class-mining tasks. Although the complexity of PM_R is less than PM_D and PM_C, the labeling cost of PM_R is higher.

To verify the efficiency of the most valuable samples extraction using MC_SVMA, five commonly used active multi-class classification methods are used to compare references. MC_BA [24] selects the most uncertain valuable samples by defining criteria that not only consider the smallest distance to the decision hyperplanes but also take into account the distances to other hyperplanes, if the sample is within the margin of their decision boundaries. MC_HA [34] selects samples that are nearest to the current hyperplane to be the most valuable samples, MC_PDA [35] extracts the most valuable samples through the probability distribution over the unlabeled samples, and Shannon entropy (MC_SEA) and Informational entropy (MC_IEA) [36] are used to extract the most valuable samples.

In the second multi-class classification process, the complexity of the construction process of *Blind_set* is $O(c|Unlabel\_set|)$, where c is the number of categories, and the complexity of the SVM training of Step3.3 in algorithm 4 is $O(|Label\_set|^2)$; thus, the complexity of extracting and classifying the most valuable samples in *CBA* is

$$O(Algorithm4)=O\left((|Label\_set|^2+c|Unlabel\_set|)\cdot t_1\right) \tag{12}$$

where t_1 is the loop number of Algorithm 4. Although the loop end condition is $Blind\_set=\phi$, the *Blind_set* is reconstructed on every loop step; thus, the loop number is not equal to the initial value of $|Blind\_set|$ and $t_1 \lll |Blind\_set|$. Similarly, the complexities of extracting and classifying the most valuable samples in *CCA* and *CNA* are as follows, respectively:

$$O(Algorithm5)=O\left((|Label\_set|^2+c|Unlabel\_set|)\cdot t_2\right) \tag{13}$$

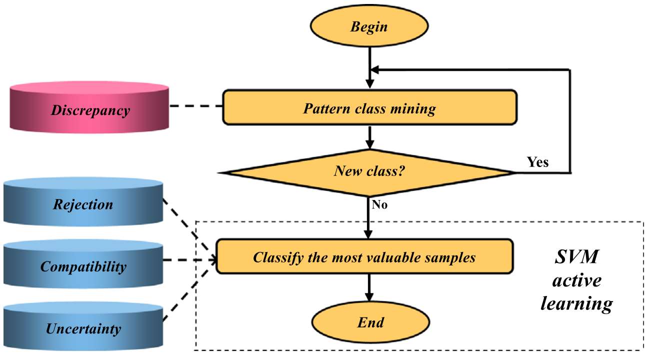

Fig. 11. MC_SVMA algorithm operating process.

$$O(Algorithm6)=O\left((|Label\_set|^2+c|Unlabel\_set|)\cdot t_3\right) \tag{14}$$

where t_2 and t_3 are the loop numbers of Algorithms 5 and 6, respectively, and $t_2 \lll |Compatibility\_set|$, $t_3 \lll |Uncertainty\_set|$.

Firstly, the MC_BA method selects the most uncertain valuable samples by defining a custom criterion based on the literature and extracts the most uncertain valuable samples using a “Round Robin” (Batch-mode) extraction method [24]. Therefore, its complexity is approximately:

$$O(MC\_BA)\approx O\left((|Label\_set|^2+c|Unlabel\_set|)\cdot t\right) \tag{15}$$

where t is the extraction loop number. Thus, the complexity of MC_BA is approximately equal to extracting and classifying the most valuable samples in the *CNA* of MC_SVMA method. However, extracting the most valuable samples by MC_BA not only considers the smallest distance to the decision hyperplanes but also takes into account the distances to other hyperplanes. It also adds a diversity step using kernel k-means clustering algorithm to improve learning efficiency, but the most valuable samples in *CBA* and *CCA* are not considered and it cannot solve the multi-class classification problem with unknown categories.

Secondly, because the complexity of the most valuable samples extraction process by the other four algorithms MC_HA, MC_PDA, MC_SEA and MC_IEA is $O(c|Unlabel\_set|)$ and the complexity of the SVM training is also $O(|Label\_set|^2)$. The complexities of these four methods are approximately:

$$O(otheralgorithms)\approx O\left((|Label\_set|^2+c|Unlabel\_set|)\cdot|Unlabel\_set|\right) \tag{16}$$

Clearly, *Blind_set*, *Compatibility_set* and *Uncertainty_set* are subsets of *Unlabel_set*, which means that $t_1 \lll |Unlabel\_set|$, $t_2 \lll |Unlabel\_set|$ and $t_3 \lll |Unlabel\_set|$. Therefore, the complexity of MC_SVMA is smaller than that of the other four active multi-class classification algorithms.

4. Simulation experiments and discussion

In this section, three aspects are verified: initial pattern class mining, extracting the most valuable samples and confirming the classification results. Ten UCI benchmark datasets [37] are used (Listed in Table 1) for the experiments. Each dataset is randomly divided into two parts: the training set and the testing set (The dataset *Machine* is specific because only one sample is labeled by the fifth category and to make the classifiers be representative, this sample serves only as the training datum), and the average experiment results of three times randomly dividing datasets are evaluated. The experiments were conducted on a PC (2.66 Ghz CPU, 1 G RAM) running Matlab7.0.

4.1. Pattern class mining

4.1.1. Effectiveness of pattern class mining

To illustrate the effectiveness of the proposed PM_D algorithm (Algorithm 1), we compare the results with the classical algorithms PM_R (Algorithm 2) and PM_C (Algorithm 3). Unlike PM_D and PM_C, the results of algorithm PM_R are not unique. Therefore, for PM_R, all the experiments are carried out five times, and the mean result is used as the final result. Fig. 12 shows the class mining results from using the benchmark datasets.

As seen in Fig. 12, as the number of new labeled samples increases, all the classes may be mined using these three algorithms. The number of mined categories is always increased using PM_D and PM_R. However, for PM_C, the results of mining some datasets produce slight fluctuations (see the mining results on datasets *Balance_scale*, *Glass*, *Letter* and *Segment*). This finding is observed because the clustering process is executed repeatedly, and the categories mined in former steps may be not appear in the following steps.

The average numbers of labeled samples are 18.9 for PM_D, 48.7 for PM_R and 28.9 for PM_C using these datasets. Compared with PM_R and PM_C, with the exception of the *Letter* dataset, PM_D reduces the number of selected samples. For the *Letter* dataset, there are only 54 samples labeled when 25 classes are mined using PM_D. However, the last class is found after 77 samples are labeled, which is larger observed with PM_C. These experimental results demonstrate that PM_D can measure the *Discrepancy* between samples effectively, and the classes determined from these samples can be mined quickly for a very low labeling cost.

To further illustrate the effectiveness of PM_D, we investigated how many new samples need to be labeled to find each class. A comparison of the results determined from the above three algorithms is shown in Fig. 13.

It can be observed from Fig. 13 that the number of new samples that need to be labeled is different when a new class is found. Generally, fewer samples are needed at the beginning of class mining. In later periods (especially where one or two classes are waiting to be mined), more samples will be required for mining the next class. In most cases, the slopes in the curves derived by PM_D are less than is observed for other two algorithms. The number of new labeled samples in each loop does not exceed four in seven datasets, and the maximum is 23. For PM_R, the number of newly labeled samples in each loop does not exceed to four in two datasets, and the maximum number even reaches 150. For PM_C, that value does not exceed four in four datasets, and the maximum number reaches 41. We can see that PM_D algorithm is very high performing with exception of the *Segment*. This is because the similarities in samples are not measured effectively by employing the Euclidean distance. In other words, two samples may not belong to a same class if their Euclidean distance is notably small.

Table 1
Benchmark datasets used in the experiments.

Datasets	Training data	Testing data	Features	classes
Balance scale	125	500	4	5
Glass	100	114	10	6
Iris	50	100	4	3
Letter	2000	18,000	16	26
Machine	100	109	7	7
Page block	1000	4473	10	5
Segment	1000	1310	19	6
Vehicle	300	546	18	4
Vowel	220	770	10	15
Wine	78	100	13	3

Because the relationship between unlabeled samples and the mined classes are adequately considered, the new algorithm PM_D can mine new pattern classes in a very short time. Therefore, it cannot only improve the class mining efficiency but also find as many classes as possible in the first stage, which plays a decisive role in the following classification task.

4.1.2. Influence of $\overline{Step}$ on class mining

For the unlabeled multi-class classification problem, the class mining results of the first step may directly affect the performance of the subsequent processes. Therefore, setting a suitable iteration step threshold $\overline{Step}$ for finding a new class is crucial. If $\overline{Step}$ is too small, then the class mining may be insufficient and some classes will not be found. However, if $\overline{Step}$ is too large, the class mining efficiency may be low and act against the original intention of active learning. Here, the influence of $\overline{Step}$ on the performance is tested before the whole performance of the unlabeled class mining is verified.

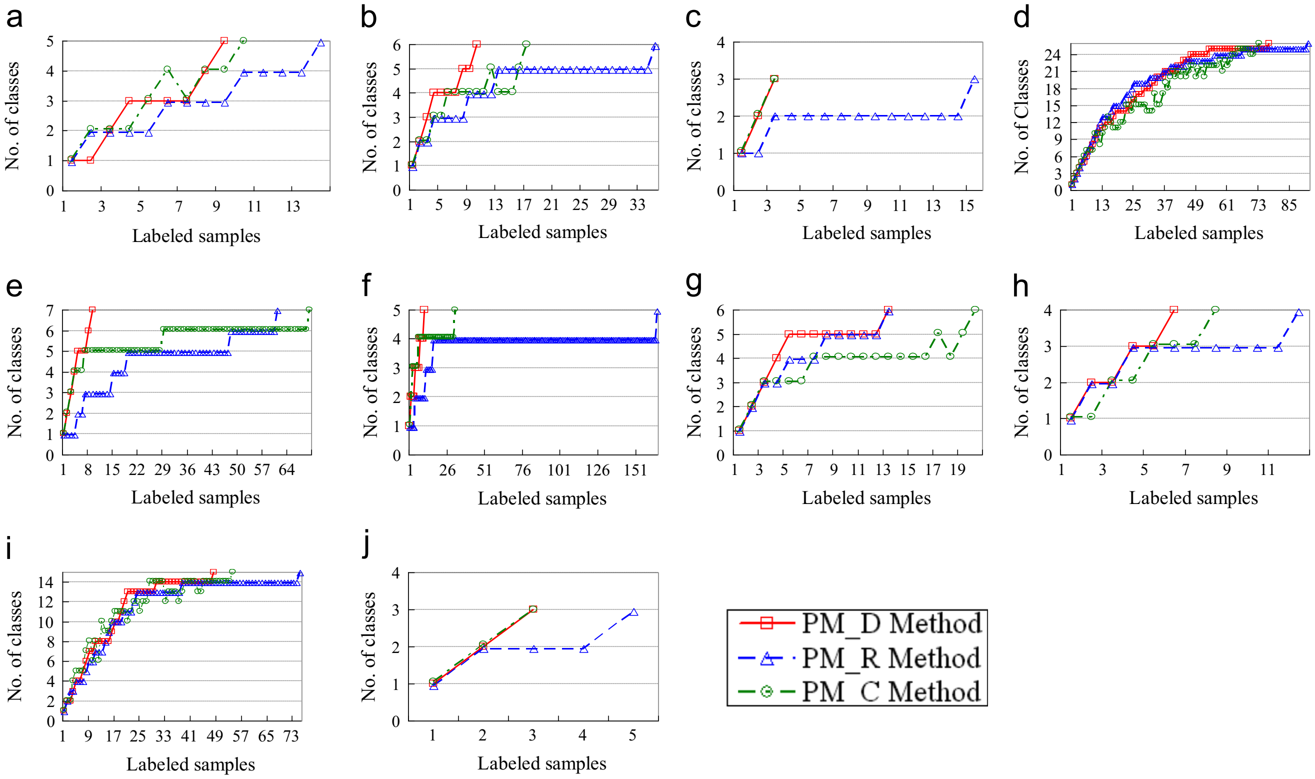

Fig. 12. Class mining results using the benchmark datasets.(a) *Balance_scale* (b) *Glass* (c) *Iris* (d) *Letter*, (e) *Machine* (f) *Page_block* (g) *Segment* (h) *Vehicle*, (i) *Vowel* (j) *Wine*.

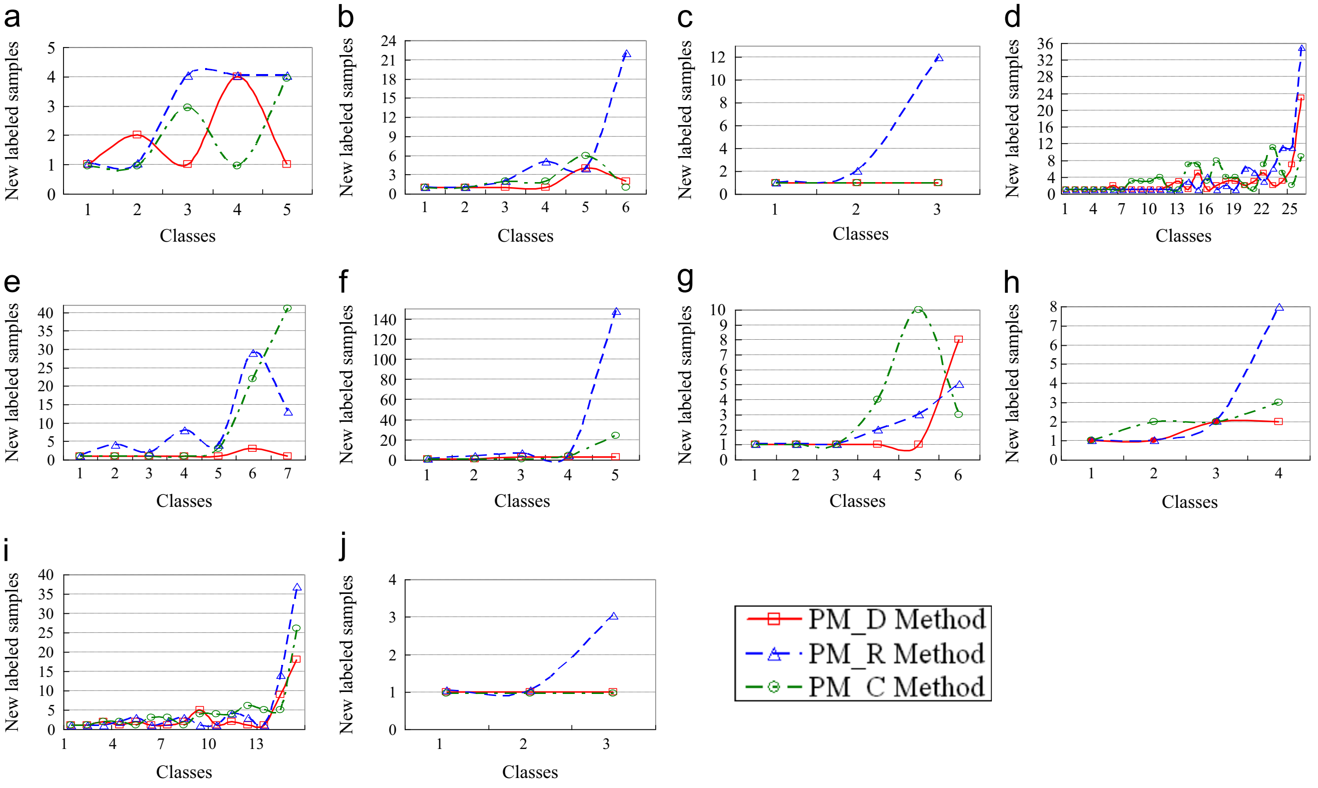

Fig. 13. Newly labeled samples from each class mining. (a) *Balance_scale*, (b) *glass*, (c) *iris* (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel* and (j) *wine*.

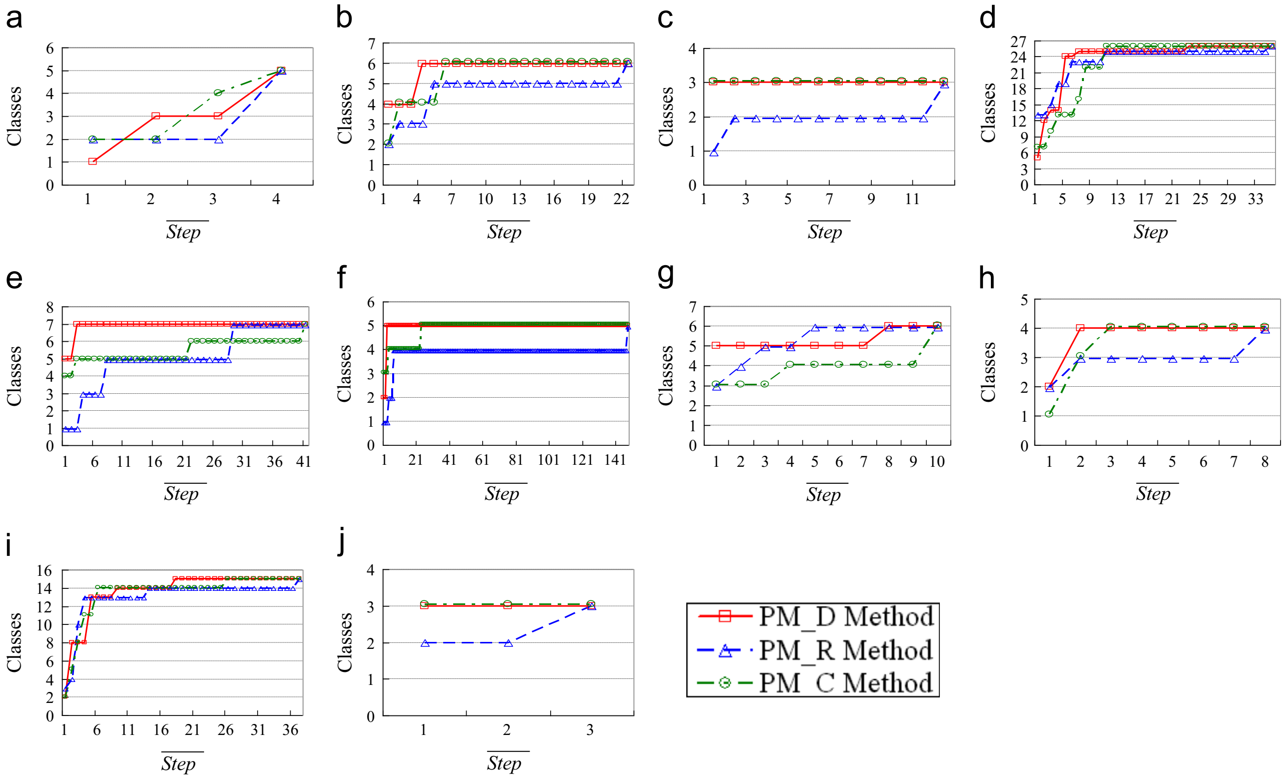

Fig. 14. The influence of the parameter $\overline{Step}$ on the mined classes. (a) *Balance_scale*, (b) *glass*, (c) *iris*, (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel* and (j) *wine*.

Fig. 14 shows the varying trends of mined class number using $\overline{Step}$ for the above three methods. It can be observed that all the classes can be found with the increment in $\overline{Step}$ for the three methods. From the eight datasets, $\overline{Step}$ determined in the PM_D is smaller than or equal to the other two methods when all the classes have been found. For the *Letter* dataset, when all the classes are found, the $\overline{Step}$ values for PM_D, PM_R and PM_C are 23, 35 and 11, respectively. For the *Segment* dataset, the $\overline{Step}$ values of the three methods are 8, 5 and 10 respectively. Because the number of the samples labeled using experts are dependent on $\overline{Step}$, the labeling cost of PM_D is significantly lower than for the other two methods. Therefore, when $\overline{Step}$ is set to a small value, good class mining results and a low labeling cost can be obtained by the PM_D.

4.1.3. Robustness of class mining

Because the sample sizes of the different categories are different for most datasets, the sample distribution is often imbalanced. For example, there is only one sample belonging to the sixth category in the *Machine* dataset. If we select samples labeled by experts randomly or by clustering, those samples belonging to categories containing more samples are most likely to be selected, which will lead greater labeling costs and slower learning times to find as many classes as possible. The PM_D selects samples by considering the distances between the labeled and unlabeled samples, but not the data distribution. Hence, those samples belong to categories with fewer samples with the same chance to be selected. In this subsection, three imbalanced datasets *Glass*, *Machine* and *Page_block* are investigated. In Fig. 15(a) is the statistical results of the sample number for each class in the three datasets and (b), (c) and (d) are the distributions of the extracted samples during the class mining process for the three class mining methods, respectively.

There are three categories with more samples than the others in the *Glass* dataset, and samples belonging to one category from the *Machine* and *Page_block* datasets.

For PM_R, the distribution of the selected samples is identical to that of the original samples and is especially obvious for *Machine* and *Page_block*. The ratios of the selected samples belonging to the large scale categories are 70.6%, 68.3% and 96.3% for these three datasets, respectively, whereas those ratios from the original data are 65%, 67% and 88.8%, respectively. This finding means that categories including more samples are more easily found. Nevertheless, those samples belonging to these categories are easily selected during next new class mining period. These samples are not helpful at finding a new class, but may lead to convergence slowly even if they cannot find enough categories. The results from PM_C are similar to that of PM_R. Although the distribution of the selected samples by PM_C are not always

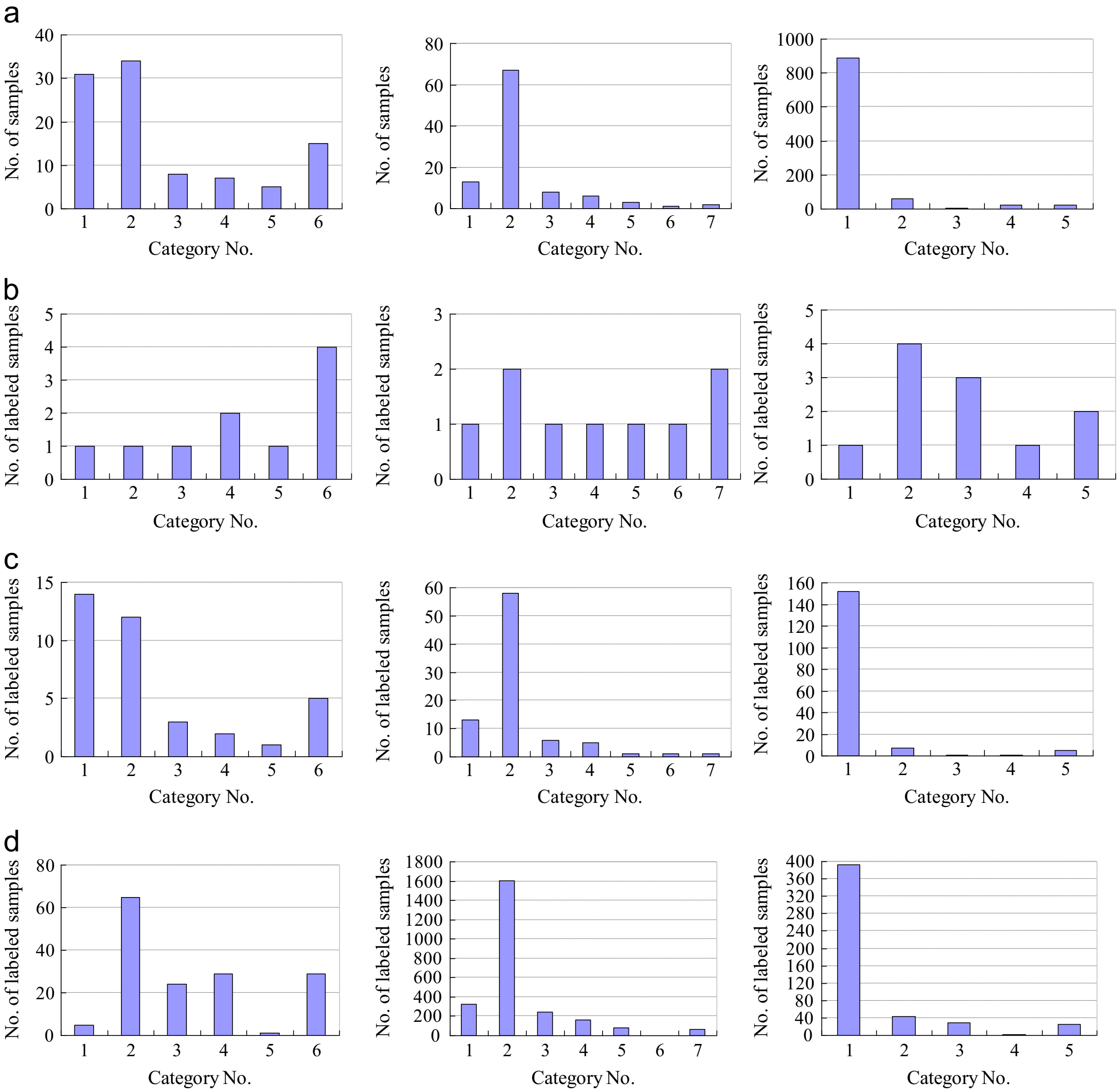

Fig. 15. Distributions of the extracted samples during the class mining process. (a) Samples in each class of the three imbalanced datasets (*Glass*, *Machine* and *Page_block*), (b) PM_D method, (c) PM_R method, and (d) PM_C method.

identical to that of the original samples, it is affected by original datasets such as the samples in second category of the *Glass* dataset, the second category of *Machine* dataset and the first category of *Page_block*. For the PM_D, the distribution of the selected samples is well-proportioned for the above three datasets, so that fewer samples need to be labeled by experts and more categories can be found synchronously. The experimental results support the assertion that the proposed PM_D is effective and robust in spite of the data distribution.

4.2. *The most valuable samples extraction*

4.2.1. *Influences of different sequences on the most valuable samples extraction*

Firstly, we testify the different sequences of the most valuable samples extraction methods. Here, the Gaussian kernel function is adopted with parameter 1.0, and the a penalization parameter is 200. Because the extraction process of the MC_SVMA method is not in a fixed order, six different extraction process sequences of the most valuable samples in *CBA*, *CCA* and *CNA* are experimental. Three measurements, the rejection samples ratio *Rejection_rate*, the compatibility samples ratio *Compatibility_rate* and the uncertainty samples ratio *Uncertainty_rate* are introduced to measure the efficiency of the most valuable samples extraction, respectively.

$$Rejection\_rate_i = \frac{|Blind\_set_i|}{|UX_i|}. \tag{17}$$

$$Compatibility\_rate_i = \frac{|Compatibility\_set_i|}{|UX_i|}. \tag{18}$$

$$Uncertainty\_rate_i = \frac{|Uncertainty\_set_i|}{|UX_i|}. \tag{19}$$

where $|Blind\_set_i|$, $|Compatibility\_set_i|$ and $|Uncertainty\_set_i|$ represent the numbers of unlabeled samples in the *CBA*, *CCA* and *CNA*, respectively, and $|UX_i|$ is the number of unlabeled samples at the *i*th iteration step.

For the MC_SVMA, the iterative steps during the extraction of the most valuable rejected and compatible samples are set 20, and it is set as twice the number of categories for the uncertain sample extraction. When the *Rejection_rate* or the *Compatibility_rate* is twice that in last step, the loop is terminated early.

To express simply, "*R*", "*C*" and "*U*" represent the extraction of rejected samples, compatible samples and uncertain samples, respectively, and "MC_SVMA_{RCU}", "MC_SVMA_{RUC}", "MC_SVMA_{CRU}", "MC_SVMA_{CUR}", "MC_SVMA_{URC}" and "MC_SVMA_{UCR}" represent the MC_SVMA methods with six different extraction sequence methods, respectively.

From the simulations that were conducted on all of the datasets (see Figs. 16–18), we found that under most circumstances, the *Rejection_rate*, *Compatibility_rate* and *Uncertainty_rate* of the MC_SVMA decrease stably. For example, the *Rejection_rate* decreases during all of the most valuable samples extraction procedure on the experimental datasets with the exception of *Vehicle* by the method MC_SVMA_{UCR}, despite of its fluctuations. The *Compatibility_rate* also decreases stably for all of the datasets on the seven datasets except for *Glass*, *Vehicle* and *Segment*. The *Uncertainty_rate* from MC_SVMA is stable on all of the datasets. The experimental results support that the rejected, compatible and uncertain valuable samples that were extracted using the MC_SVMA are effective and efficient under different extraction sequences.

4.2.2. *Performance verifications under different extraction sequences*

Fig. 19 shows the changes and trends in the testing results using the different extraction sequences of the most valuable samples from the MC_SVMA method on all of the datasets. It can be observed that all of the cases obtain similar results under different extraction sequences. However, some cases are not stable when compared with binary active classification problems because the most valuable samples extraction processes for multi-class classifications are difficult in nature.

Fig. 20 is comparisons of the average testing accuracy and standard deviation to measure the overall performance. The upper standard deviation and under standard deviation are defined as follows.

$$UPSD = \sqrt{\frac{1}{n_{upper}-1}\sum_{a_i \ge \overline{A}} (a_i - \overline{A})^2} \tag{20}$$

$$UNSD = \sqrt{\frac{1}{n_{under}-1}\sum_{a_i < \overline{A}} (a_i - \overline{A})^2} \tag{21}$$

where n_{upper} (or n_{under}) represents the numbers of the testing accuracy values larger (or smaller) than average values $\overline{A}$. The n_{upper} of *Iris* data by MC_SVMA_{RUC} and the n_{under} of *Wine* data by former three methods are equal to one, so the *UPSD* or *UNSD* of these circumstances are not existed.

It can be observed that the average testing accuracy results of the MC_SVMA method by a variety of the most valuable samples extraction sequences are closed, and of those, the *UPSD* and *UNSD* are small except for *Page_block*. For *Page_block*, the testing accuracy values of the above training steps are obviously smaller than the average values, except for the MC_SVMA_{URC} and MC_SVMA_{UCR}.

Table 2 gives the number of rounds for producing the statistics shown in Fig. 19. It can be observed that for more categories of datasets (such as *Letter* and *Vowel*), the number of rounds is large. If the categories number of a dataset is large, then the classifier number may be large and complex. Thus, the extractions of rejected, compatible and uncertain valuable samples are then complex, and the extraction process may be executed many times. However, if the categories number is small but the dataset size is small, the rounded number is not very large (such as *Segment* and *Page_block*). Therefore, for the datasets that have too many categories, the model should be simple.

4.3. *Model selection*

All of the above MC_SVMA methods with the RBF kernel have two parameters: kernel parameter *p* and penalty parameter *C*, and these two parameters should be tuned to obtain better generalization of the methods [38–41]. It can be observed from the above that there is a minor variation in the testing accuracy among the different most valuable samples extractions. To simplify the analysis, this part executes the model selection only for MC_SVMA_{RCU} except for *Page_block* because of its large standard deviation in the testing accuracy. For *Page_block*, MC_SVMA_{UCR} is selected to analyze. For the sake of brevity, we present only the concrete analysis process of the *Wine* datasets in this paper.

Table 3 shows the influences of *p* on the maximal testing accuracy (MAX), the average testing accuracy (AVE) and the number of algorithm executing rounds (ROU) on the *Wine* dataset by the MC_SVMA_{RCU} method. From Table 3, it can be observed that when *p* takes a small value (e.g., $p=0.1$), the testing accuracy is small (MAX=92%, AVE=78.3333%). Then, the testing results increas with increments in *p*, and the MAX and AVE remain large values with [88%, 98%] and [82.25%, 91.6%] for a certain range of *p* (i.e., [0.5, 10]). Again, the testing results tend to decrease as *p* continues to increment. After $p \ge 20$, the testing accuracy varies mildly, and the number of algorithm executing rounds becomes a constant. Additionally, when $p \ge 30$, all of the testing accuracy values reach 40%. A similar trend in the number of algorithm executing rounds can also be observed from Table 3. By the presented approach, the optimal p is 2.0.

Secondly, the penalty parameter C is tuned to improve the generalization performance of the model. Here, the kernel parameter takes the value 2.0. Fig. 21 shows the average testing accuracy versus C. From Fig. 21, it can be seen that the value of the average testing accuracy is small when C has a small value (i.e., the average testing accuracy is 25% when C is 0.01). Then, the average testing accuracy increases steeply when C varies from 0.01 to 10. At this moment, the average testing accuracy reaches 82.25%. Finally, the average testing accuracy changes mildly after $C \geq 10$. When $C \geq 90$, the average testing accuracy achieves 91.6% and does not change. Hence, it can be concluded that the testing accuracy is scarcely influenced by C. To make the learning process stable, a large value should be set up for C (e.g., $C = 100$ for the *Wine* dataset).

Because the testing accuracy is scarcely influenced by C, Table 4 gives only the kernel parameter optimization results of the other datasets. Here, the penalty parameter C is adopted with parameter 200. It can be observed that the kernel parameter p takes a value in the range [0.8, 2.5] and the model can obtain good generalization performance. The reason is that the undetermined parameters optimization method can obtain only a locally optimal solution but not globally optimal results. Thus a grid search method can be used to optimize the model parameters and improve the generalization performance for the small size datasets, although this approach is inefficient.

4.4. Comparison with other active multi-class classification methods

Fig. 22 shows the changes and trends in the testing results using the above presented serial MC_SVMA methods (*MC_SVMA*$_{UCR}$ for *Page_block* and *MC_SVMA*$_{RCU}$ for the other datasets) and other active multi-class classification methods presented above (in Section 3.4), namely, MC_BA, MC_HA, MC_PDA, MC_SEA and MC_IEA), on all of the datasets. All of the parameters of these methods are tuned by the algorithm of Section 4.3. For the serial MC_SVMA methods, the sample query end conditions are the same as in Section 3.2.1, and for the other five active multi-class classification methods, the sample query end conditions are

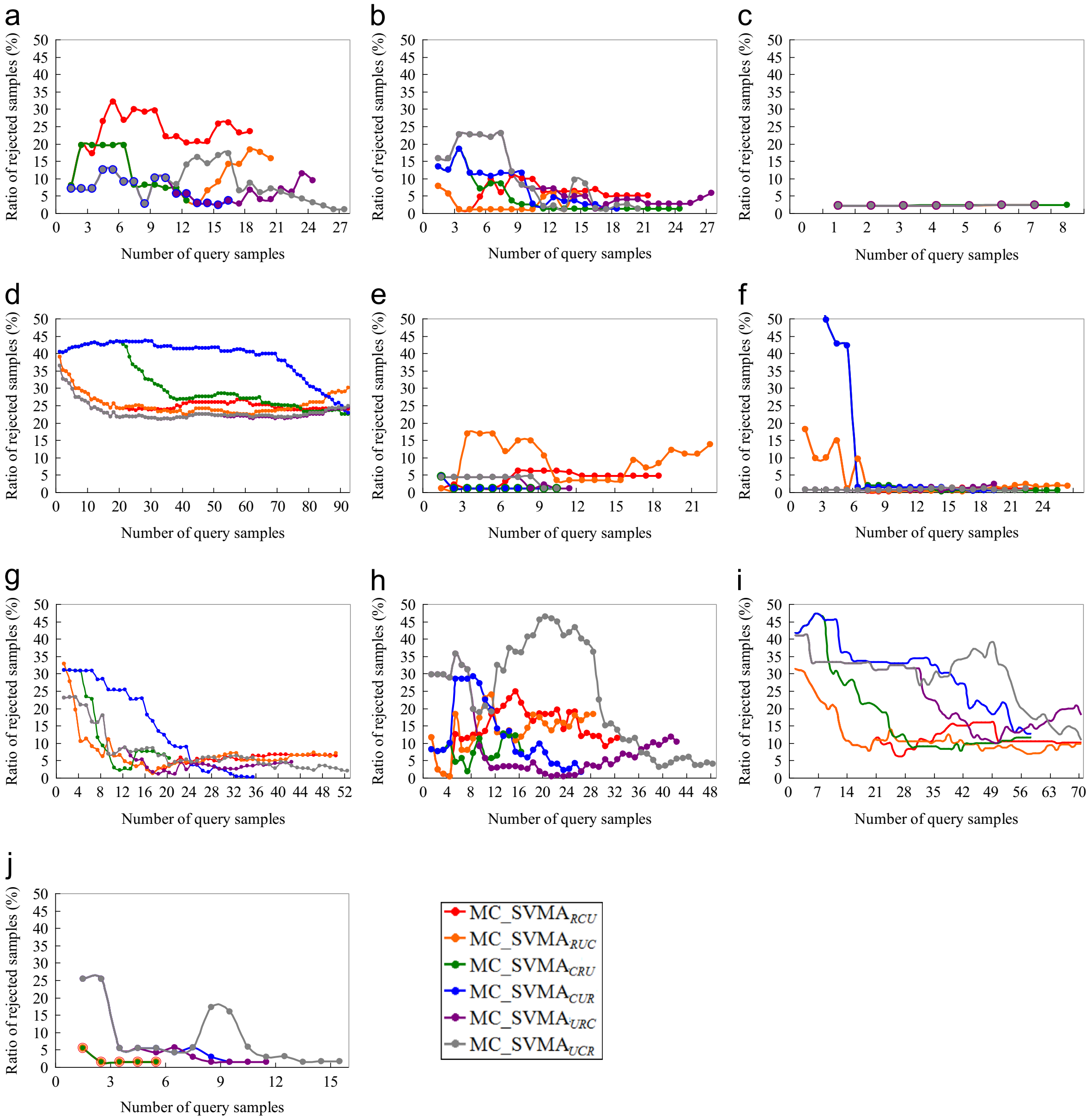

Fig. 16. Change tendencies in the *Rejection_rate* (a) *Balance_scale*, (b) *glass*, (c) *iris*, (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel*, *and* (j) *Wine*.

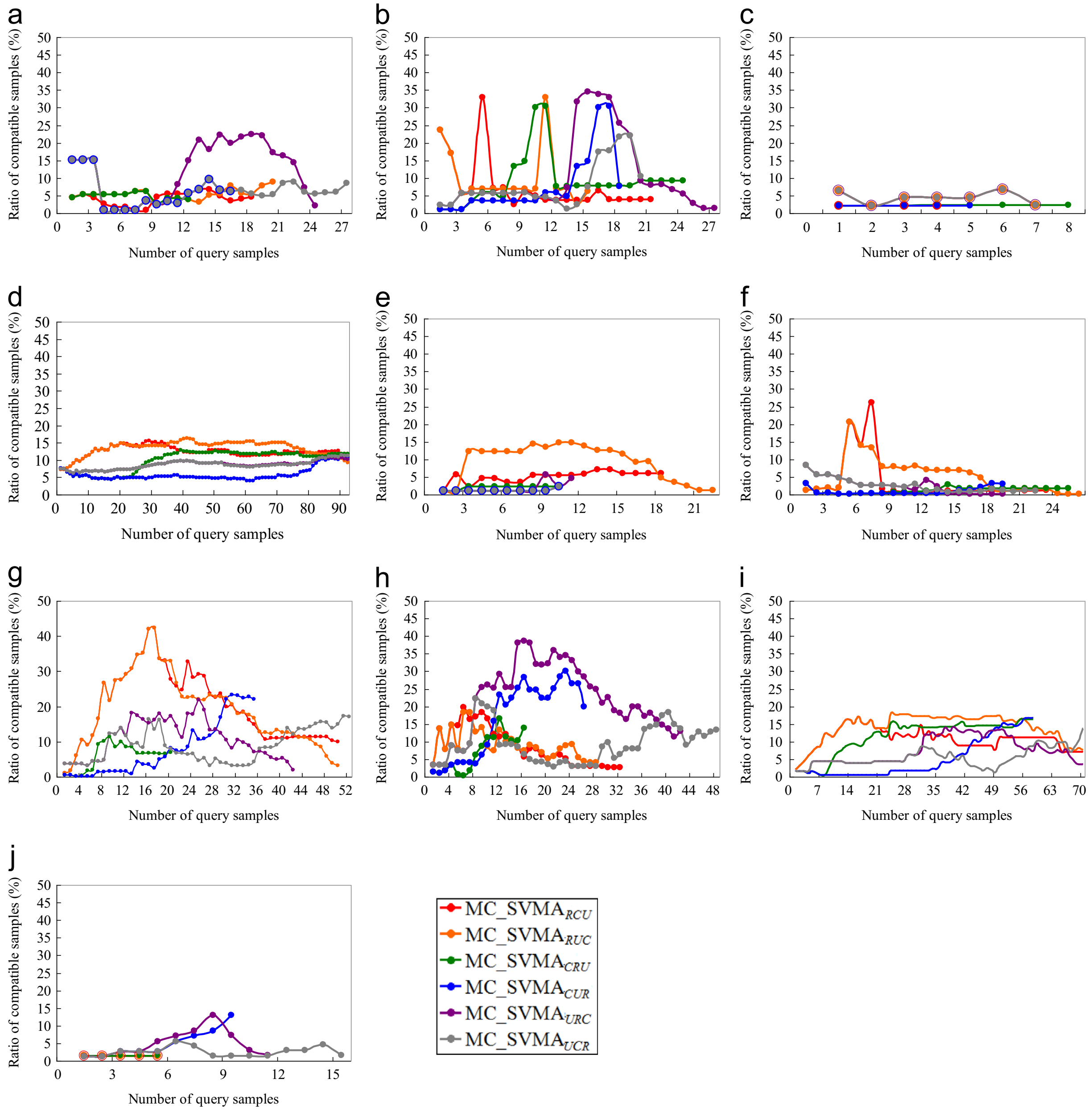

Fig. 17. Change tendencies in the *Compatibility_rate*. (a) *Balance_scale*, (b) *glass*, (c) *iris*, (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel*, and (j) *Wine*.

set to be the same as in the corresponding MC_SVMA except on the dataset *Vowel*. For the *Vowel* dataset, there is not enough data because the most uncertain valuable samples fall within the margin of the decision boundaries for MC_BA and MC_HA, and they are terminated early without extracting the 70 unlabeled samples.

The maximal testing accuracy using MC_SVMA is found to be the best among these methods except on the datasets *Machine*, *Segment* and *Vowel*. On the *Machine* dataset, the maximal testing accuracy of MC_SVMA appears to be the best of the results that are observed using all of the methods. On the *Vowel* dataset, the maximal testing accuracy value using the MC_SVMA is only slightly lower than what is observed when using the other four active multi-class classification methods. On the *Segment* dataset, the maximal testing value using the MC_SVMA is only slightly lower than what is observed when using the MC_BA. The experimental results support that MC_SVMA obtains good generalization performance among the active multi-class classification learning models.

Fig. 23 shows comparisons of the average testing accuracy and standard deviation from measuring the overall performance of these six methods. It can be observed that the average testing accuracy results of the MC_SVMA method are greater than or equal to the other four active multi-class classification methods on seven datasets except for the *Page_block*, *Segment* and *Wine* datasets. Moreover, the standard deviation values on some of the datasets of the MC_SVMA method are large, such as the UNSD of *Page_block* and the UPSD of *Vehicle*. Clearly, this finding is caused by the testing accuracy values being too low for the earlier steps of *Page_block* and too high for the last steps of *Vehicle*. However, these values do not impact the good generalization performance of the MC_SVMA method because we focused mainly on the most optimal testing accuracy of the model.

Furthermore, we find from the experiments that the training times of active multi-class classification methods MC_HA, MC_PDA, MC_SEA and MC_IEA are longer than MC_SVMA and MC_BA because the former methods need to extract only one sample in

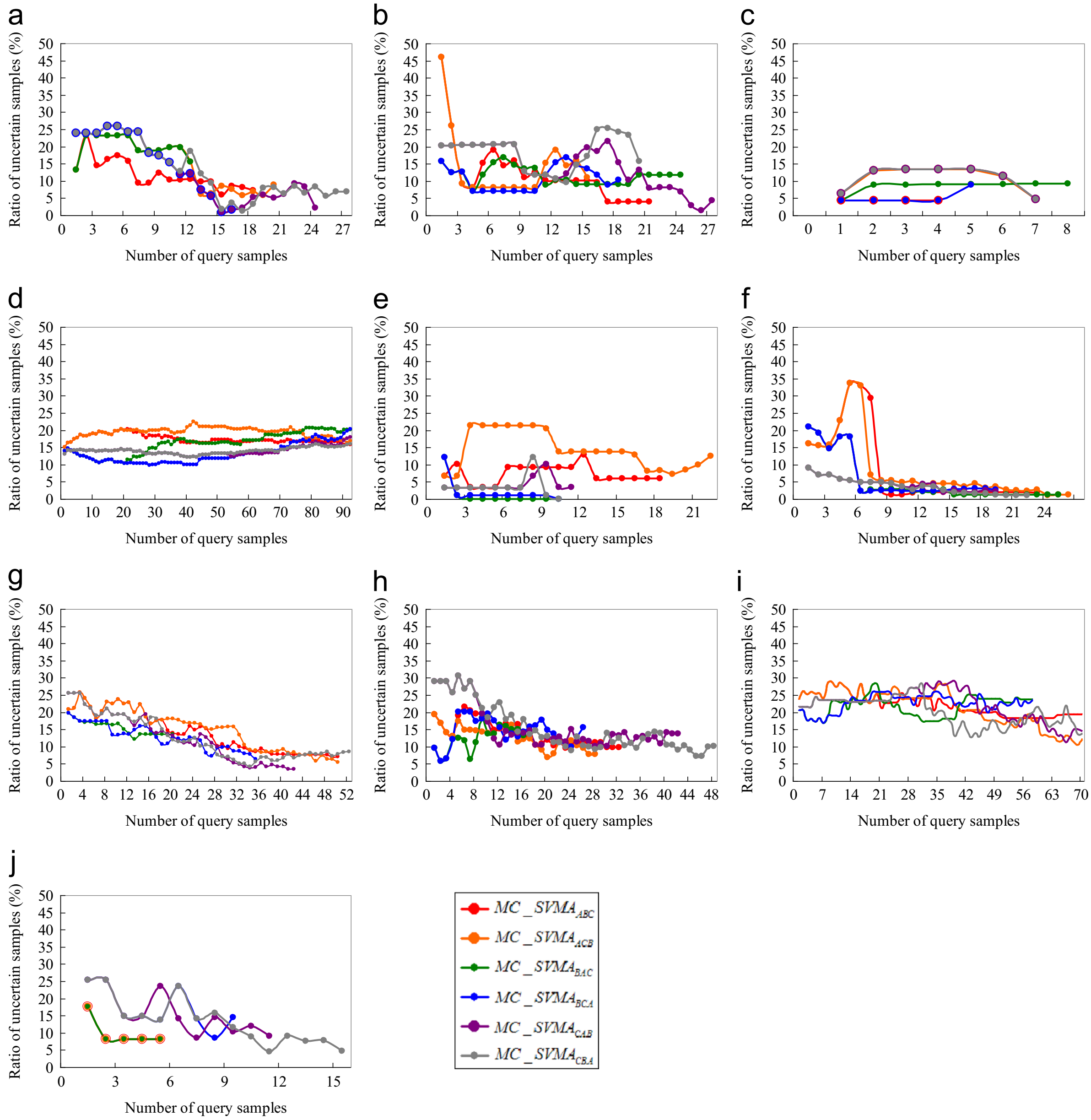

Fig. 18. Change tendencies in the *Uncertainty_rate*. (a) *Balance_scale*, (b) *glass*, (c) *iris*, (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel* and (j) *wine*.

an extraction loop, but the "Round Robin" method is used by MC_SVMA and MC_BA to extract the uncertain valuable samples.

4.5. Comparison with traditional multi-class classification methods

The traditional multi-class classification methods that do not use active learning are compared. Firstly, it must be stated that these traditional multi-class classification methods cannot solve directly the problems that this paper proposes, i.e., multi-class classification problems with unknown categories. Thus, in this section, we compare only the generalization performance and learning efficiency of MC_SVMA and traditional multi-class classification methods. We assume that all of the training datasets have labels for the traditional multi-class classification methods. Here, the OvR [23], OvO [24], DAG [25] and the multi-class classification method based on decision tree (DT) [26], are compared. The kernel function and penalty parameters are tuned in the earlier experiments.

Table 5 shows the testing accuracy results of these methods. For the MC_SVMA active learning method, only the maximal testing accuracy is meaningful and is obtained for comparison. In the table, the unlined values that represent the corresponding test accuracy values are better than the other four traditional multi-class classification methods. It can be observed that for five datasets, *Balance_scale*, *Glass*, *Iris*, *Vehicle* and *Wine*, the testing accuracy values of MC_SVMA are higher than for the other traditional multi-class classification methods, and the testing accuracy values of MC_SVMA on *Machine*, *Page_block* and *Segment* are only slightly lower than that of the traditional methods. It means that many of the most valuable samples are in *CBA*, *CCA* and *CNA*, and they are difficult to correctly classify automatically by traditional multi-class classification methods (such as OvO, OvR, DAG and DT). For OvO and OvR, the most valuable samples in *CBA*, *CCA* are classified into a category randomly, and for DAG and DT methods, these samples are classified blindly into one class. However, for the MC_SVMA method, most valuable samples are not classified into any class but only extracted, and then the expert-machine interaction method is used to maintain the high classification accuracy of these important samples. In summary, the MC_SVMA obtains optimal

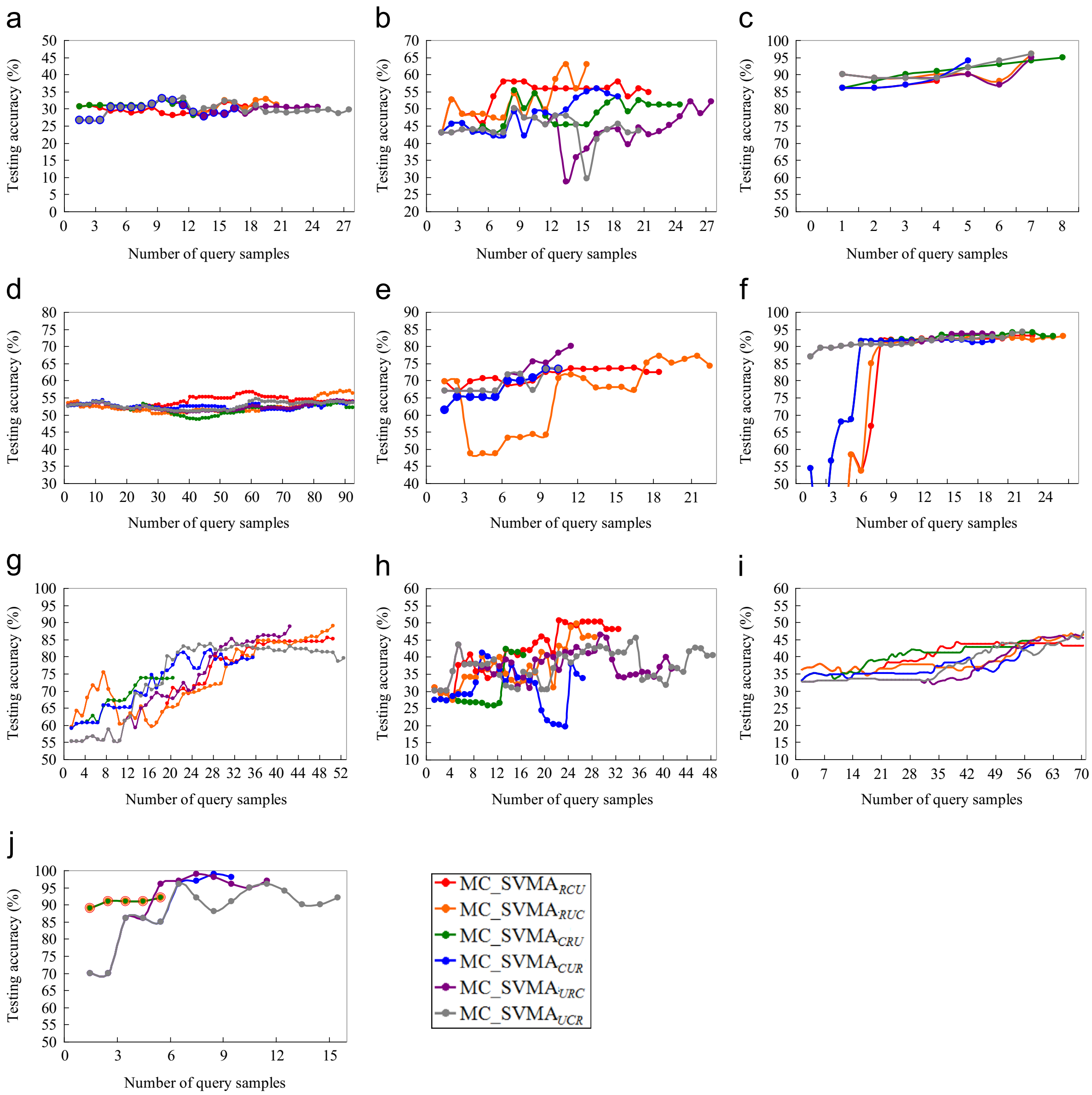

Fig. 19. Change tendencies in the testing accuracy. (a) *Balance_scale*, (b) *glass*, (c) *iris*, (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel*, and (j) *wine*.

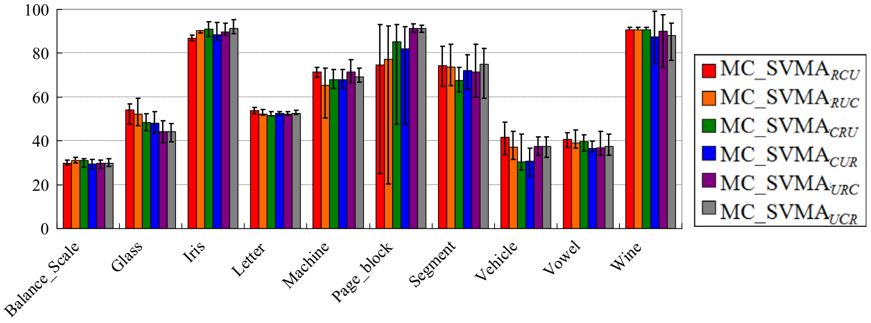

Fig. 20. Comparison results of different series MC_SVMA methods.

results on most datasets when compared with traditional multi-class classification methods because it can extract the most valuable samples with more important classification information and obtain higher testing accuracy results.

Table 6 shows the training time for the five methods. It can be observed that on all of the datasets, the training time of MC_SVMA is shorter than that of the traditional methods. The learning efficiency of MC_SVMA is on average 9 to 230 times as fast as the OvR method on all of the datasets, and the learning efficiency of MC_SVMA is on average 6 to 93 times as fast as the OvO and DAG methods except for the *Letter* dataset. The DT learning efficiency of DT is slightly high, but its generalization performance is not perfect.

4.6. Discussion about the combination of MC_SVMA and OvO

This paper proposes MC_SVMA to solve multi-class classification problems with unknown classes by combining active learning with the OvR based on the SVM. From Table 6, it can be observed that the learning efficiency of OvR is lower than the OvO method. Especially if the training set is large, OvO can obtain higher learning efficiency. For example, the traditional OvO is nearly ten learning times as fast as OvR. However, the complexity of MC_SVMA is dependent on the number of classifiers. For a classification problem with the same number of categories, the number of classifiers of OvO is larger than the OvR method. Especially if the number of categories in the dataset is large, the difference is more obvious. Here, we take *machine* dataset as an example; if the OvR is used to construct the MC_SVMA model, then the number of classifiers is 7. But when the OvO is used, the number of classifiers becomes 21. The reason is that the complexity of extracting the most valuable samples in the *CBA*, *CCA* and *CNA* processes is directly dependent on the number of classfiers. Thus, although the OvO is used in this method, the learning efficiency might be not obviously improved and the generalization performance could be affected by the complex extraction process.

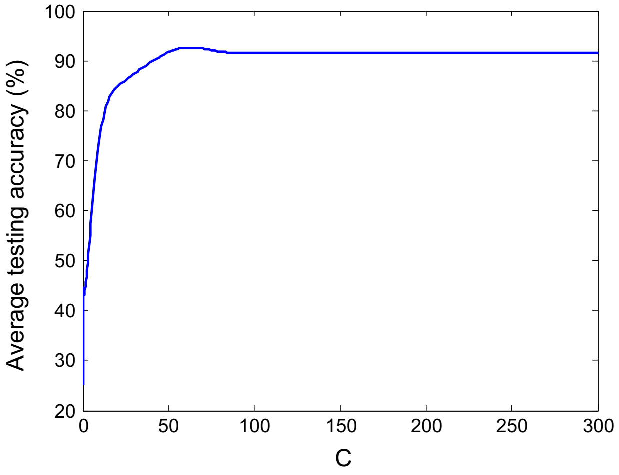

Fig. 21. The average testing accuracy versus penalty parameter *C*.

5. Conclusions

This paper proposes using MC_SVMA to solve multi-class classification problems with unknown classes by combining active learning with SVM. The MC_SVMA has several advantages: (1) it can mine class information from given unlabeled samples and obtain categories in different levels according to the different requirements. (2) based on the labeled samples, certain unlabeled

Table 4
Parameter optimization results.

Datasets	*p*	ROU	MAX	AVE(%)
Balance scale	1.5	13	33.4	32.4
Glass	1.0	21	57.9	54.0
Iris	2.5	6	95	91.5
Letter	1.5	70	56.6	53.8
Machine	0.8	21	76.2	74.1
Page block	1.0	23	93.1	74.6
Segment	1.0	50	85.5	74.3
Vehicle	1.5	34	59.7	39.6
Vowel	1.5	70	46.2	42.6
Wine	2.0	15	98	91.6

Table 2
the number of rounds for producing the results of the MC_SVMA methods

Datasets	*MC_SVMA*$_{RCU}$	*MC_SVMA*$_{RUC}$	*MC_SVMA*$_{CRU}$	*MC_SVMA*$_{CUR}$	*MC_SVMA*$_{URC}$	*MC_SVMA*$_{UCR}$
Balance scale	18	20	12	16	24	27
Glass	21	15	24	18	27	20
Iris	4	7	8	5	7	7
Letter	92	92	92	92	92	92
Machine	18	22	10	10	11	10
Page block	23	26	25	19	19	22
Segment	50	50	20	35	42	52
Vehicle	32	28	16	26	42	48
Vowel	70	70	58	58	70	70
Wine	5	5	5	9	11	15

Table 3
The testing results from different values of the kernel parameter *p*.

p	ROU	Testing accuracy with query samples (%)	MAX (%)	AVE (%)
0.1	3	60, 83, 92	92	78.3333
0.5	7	68, 60, 82, 95, 95, 95	95	82.5
1.0	5	89, 91, 91, 91, 91	91	90.6
1.5	14	79, 72, 85, 89, 76, 74, 90, 88, 88, 88, 92, 92, 92, 92	92	85.5
*2.0	15	79, 77, 82, 93, 94, 92, 93, 94, 90, 92, 96, 98, 98, 98, 98	98	91.6
2.5	16	79, 80, 82, 93, 95, 94, 94, 98, 94, 97, 96, 97, 97, 97, 97, 97	98	87.4375
3.0	14	79, 78, 90, 89, 80, 85, 88, 89, 93, 92, 93, 93, 93, 93	93	88.21429
5.0	8	84, 76, 75, 80, 80, 83, 91, 89	91	82.25
10.0	7	85, 77, 84, 84, 79, 88, 87	88	83.42857
15.0	7	64, 64, 56, 40, 40, 40, 40	64	49.14286
20.0	7	42, 41, 40, 40, 40, 40, 40	42	40.42857
30.0	7	40, 40, 40, 40, 40, 40, 40	40	40
50.0	7	40, 40, 40, 40, 40, 40, 40	40	40
100.0	7	40, 40, 40, 40, 40, 40, 40	40	40

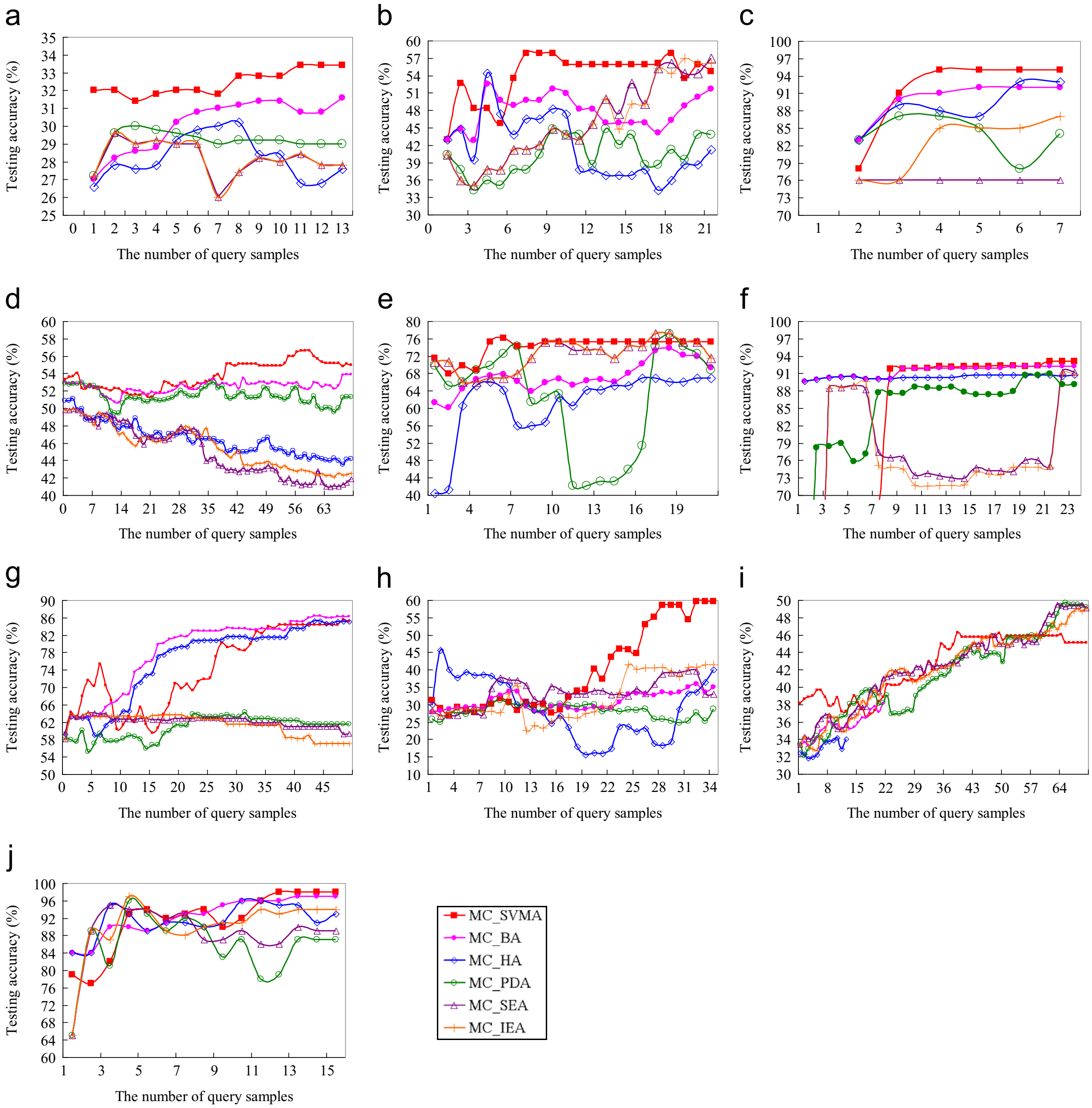

Fig. 22. Comparison testing accuracy with the other active multi-class classification methods. (a) *Balance_scale*, (b) *glass*, (c) *iris*, (d) *letter*, (e) *machine*, (f) *page_block*, (g) *segment*, (h) *vehicle*, (i) *vowel*, and (j) *wine*.

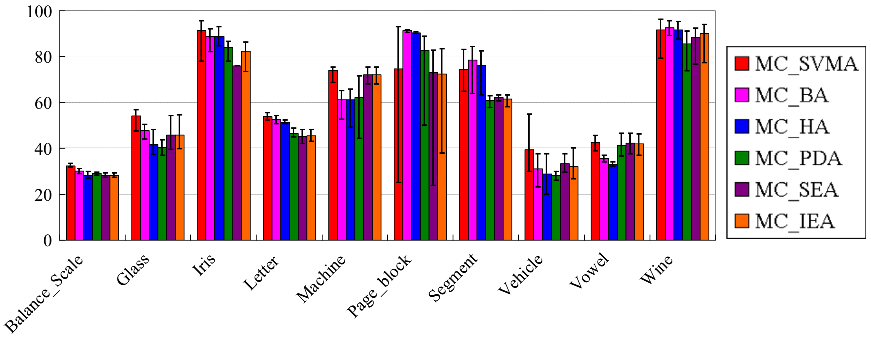

Fig. 23. Comparison results of different active multi-class classification methods.

samples that are difficult to distinguish can be classified, and certain new categories may be found synchronously, (3) unlike the clustering technique, the MC_SVMA requires experts to participate in the whole learning process to guide the process properly but for a small labeling cost, and (4) the MC_SVMA may provide another way to process large quantities data.

Table 5
Comparison of the testing accuracy of various methods (%).

Datasets	MC_SVMA	OvR	OvO	DAG	DT
Balance_scale	**33.4**	27.6	28	28.8	26
Glass	**57.9**	47.4	55.3	51.8	45.6
Iris	**95**	91	91	91	91
Letter	56.6	68.9	81.7	82.9	74.6
Machine	76.2	79.8	83.5	81.7	75.2
Page_block	93.1	94.7	94.6	92.8	87.6
Segment	85.5	95.6	96.5	93.4	82.0
Vehicle	**59.7**	29.5	31.7	32.6	28.8
Vowel	46.2	69.9	73.9	71.2	69.0
Wine	**98**	97	97	97	95

Table 6
Comparison of the training time of various methods (s).

Datasets	MC_SVMA	OvR	OvO	DAG	DT
Balance_scale	0.31	25.2	11.28	12.37	7.84
Glass	0.36	8.25	3.17	4.50	2.79
Iris	0.03	1.45	0.63	0.75	0.74
Letter	914.4	8215.02	876.39	1238.7	596.5
Machine	0.81	8.64	4.31	6.375	3.30
Page_block	13.41	3078.9	1240.88	1143.8	865.8
Segment	16.68	2156.7	461.19	820.3	477.91
Vehicle	1.19	74.03	25.19	30.62	22.6
Vowel	11.13	223.0	86.05	128.12	64.48
Wine	0.05	2.49	0.95	1.81	0.91

Because the pattern class mining of unlabeled data is an uncertain problem, dynamic class mining may be more useful for practical problems. Categories should be identified on more abstract levels (fewer classes) or more concrete levels (more classes) automatically by the specific cognitive level of the users. Therefore, research on dynamic class granule mining will constitute our future work. Finally, the similarities between the samples in some special datasets may not be effectively measured using the Euclidean distance, and determining how to classify these samples using the MC_SVMA method is also an important issue.

Conflict of interest

None.

Acknowledgment

The authors are grateful to the editors and the anonymous referees for their thoughtful comments and valuable suggestions, which helped to improve this paper. The work described in this paper was partially supported by the National Natural Science Foundation of China (Nos. 60975035, 71031006, and 61273291), Research Project Supported by Shanxi Scholarship Council of China (No. 2012-008), Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201305).

References

[1] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.

[2] H.Y. Lin, Efficient classifiers for multi-class classification problems, Decis. Support Syst. 53 (3) (2012) 473–481.

[3] C.C. Chang, L.J. Chien, A novel framework for multi-class classification via ternary smooth support vector machine, Pattern Recognit. 44 (6) (2011) 1235–1244.

[4] C. Bourke, K. Deng, S.D. Scott, et al., On reoptimizing multi-class classifiers, Mach. Learn. 71 (2–3) (2008) 219–242.

[5] S. Dasgupta, Two faces of active learning, Theor. Comput. Sci. 412 (19) (2011) 1767–1781.

[6] G. Qi, H.S. Hua, Y. Rui, et al., Two-dimensional multi-label active learning with an efficient online adaptation model for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1880–1897.

[7] C. Dima, M. Hebert, A. Stentz. Enabling learning from large datasets: applying active learning to mobile robotics, in: Proceedings of the International Conference on Robotics and Automation, Piscataway, NJ, IEEE,2004, 108–114.

[8] L.H. Li, X.M. Jin, S.J. Pan, et al. Multi-domain active learning for text classification, in: Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD2012), ACM, 2012.

[9] S. Tong, Active Learning: Theory and Applications, Doctor Thesis, Standford University, California, USA, 2001.

[10] A. Beygelzimer, S. Dasgupta, J. Langford. Importance weighted active learning, in: Proceedings of the 26th International Conference on Machine Learning, Wisconsin, 2009.

[11] J.B. Zhu, H.Z. Wang, T.S. Yao, et al. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, 2008, pp. 1137–1144.

[12] H.S. Seung, M. Opper, H. Sompolinsky. Query by committee, in: Proceedings of the 15th Annual ACM Workshop on Computational Learning Theory, California, 1992, pp. 287–294.

[13] S.S. Ho, H. Wechsler, Query by transduction, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1557–1571.

[14] M.F. Abdel Hady, F. Schwenker, Combining committee-based semi-supervised and active learning, J. Comput. Sci. Technol. 25 (4) (2010) 681–698.

[15] Y. Freund, H.S. Seung, E. Samir, et al., Selective sampling using the query by committee algorithm, Mach. Learn. 28 (23) (1997) 133–168.

[16] E.W. Saad, J.J. Choi, J.L. Vian, et al., Query-based learning for acrospace applications, IEEE Trans. Neural Netw. 14 (2003) 128–139.

[17] K. Dwyer, R. Holte. Decision tree instability and active learning, in: Proceedings of the European Conference on Machine Learning (ECML), Warsaw, 2007, pp. 128–139.

[18] V. Vapnik, Statistical Learning Theory, Wiley Press, New York (1998) 493–520.

[19] G. Schohn, D. Cohn, Less is more: active learning with support vector machines, in: Proceedings of the 17th International Conference on Machine Learning, San Francisco, Morgan Kaufmann, 2000, pp. 45–66.

[20] O.L. Mangasarian, D.R. Musicant, Active support vector machine classification, Advances in Neural Information Processing Systems, 13, MIT Press (2000) 577–583.

[21] B. Demir, L. Bruzzone, A multiple criteria active learning method for support vector regression, Pattern Recognit. 47 (7) (2014) 1558–1567.

[22] W.J. Wang, H.S. Guo, Y.F. Jia, et al., Granular support vector machine based on mixed measure, Neurocomputing 101 (2013) 116–128.

[23] L. Bottou, C. Cortes, J. Denker, et al. Comparison of classifier methods: a case study in handwriting digit recognition, in: Proceedings of the International Conference on Pattern Recognition (ICPR), 1994, pp. 77–87.

[24] S. Patra, L. Bruzzone, A batch-mode active learning technique based on multiple uncertainty for SVM classifier, IEEE Trans. Geosci. Remote Sens. Lett. 9 (1) (2012) 497–501.

[25] G. Chen, T. Wang, L. Gong, et al., Multi-class support vector machine active learning for music annotation, Int. J. Innov. Comput. Inf. Control 6 (3) (2010) 921–930.

[26] U. Kreoel, Pairwise classification and support vector machines, in: B. Schokopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 255–268.

[27] J.C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin DAG's for multi class classification, Advances in Neural Information Processing Systems, 12, MIT Press, Cambridge, MA (2000) 547–553.

[28] C.W. Hsu, C.J. Lin, A comparison of methods for multiple support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.

[29] M.H. Horng, Multi-class support vector machine for classification of the ultrasonic images of supraspinatus, Expert Syst. Appl. 36 (4) (2009) 8124–8133.

[30] F. Lauer, Y. Guermeur., MSVMpack: a multi-class support vector machine package, J. Mach. Learn. Res. 12 (2011) 2293–2296.

[31] K. Christine, W. Stefan. Multi-class ensemble-based active learning, in: Proceedings of the European Conference on Machine Learning (ECML), Berlin, 2006, pp. 687–694.

[32] A.J. Joshi, Scalable active learning for multiclass image classification, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2259–2273.

[33] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of SIAM International Conference on Data Mining, 2004, pp. 333–344.

[34] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (2002) 45–66.

[35] P. Jain, A. Kapoor, Active learning for large multi-class problems, in: Proceedings of the 2009 IEEE Computer Vision and Pattern Recognition, 2009, pp. 219–224.

[36] G.J. Qi, X.S. Hua, Y. Rui, et al. Two-dimensional active learning for image classification, in: Proceedings of the 2008 IEEE Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[37] UCI Benchmark, UCI Machine Learning Repository, Available from: 〈http://www.ics.uci.edu/mleam/MLRepository.html〉, 2010.

[38] A. Shawkat, K.A. Smith-Miles, A meta-learning approach to automatic kernel selection for support vector machines, Neural Netw. 24 (1–3) (2006) 173–186.

[39] Z. Wang, S.C. Yan, C.S. Zhang, Active learning with adaptive regularization, Pattern Recognit. 44 (10–11) (2011) 2375–2383.

[40] P. Wittek, C.L. Tan, Compactly supported basis functions as support vector kernels for classification, IEEE Trans. Pattern Anal. Mach. Intell. 33 (10) (2011) 2039–2050.

[41] W.J. Wang, Z.B. Xu, V.Z. Lu, et al., Determination of the spread parameter in the Gaussian kernel for classification and regression, Neurocomputing 55 (3–4) (2003) 643–663.

Husheng Guo obtained his Ph.D. degree from Department of Computer and Information Technology, Shanxi University in 2014. Now he is lecturer at School of Computer and Information Technology, Shanxi University. He has published more than 10 academic papers on machine learning and support vector machine. His research interests include support vector machine, kernel methods and machine learning, etc.

Wenjian Wang obtained her Ph.D. degree from Institute for Information and System Science, Xi'an Jiaotong University in 2004. Now she is a professor and Ph.D. supervisor at School of Computer and Information Technology and Key Laboratory of Computational Intelligence and Chinese Information Processing, Shanxi University. She has published more than 70 academic papers on machine learning, computational intelligence and data mining. Her current research interests include support vector machine, neural networks, machine learning and environmental computations, etc.