Knowledge-Based Systems 198 (2020) 105922

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A hybrid scheme-based one-vs-all decision trees for multi-class classification tasks

Jianjian Yan, Zhongnan Zhang *, Kunhui Lin *, Fan Yang, Xiongbiao Luo

School of Informatics, Xiamen University, Xiamen 361005, China

ARTICLE INFO

Article history:
Received 27 June 2019
Received in revised form 30 March 2020
Accepted 14 April 2020
Available online 20 April 2020

Keywords:
Decision tree
One-vs-all
Split criteria
Hybrid scheme
Multi-class classification

ABSTRACT

Decision tree algorithms have been proved to be a powerful and popular approach in classification tasks. However, they do not have reasonable classification performance in multi-class scenarios. In the present study, decision tree algorithms are combined with the one-vs-all (OVA) binarization technique to improve the generalization capabilities of the scheme. However, unlike previous literature that has focused on aggregation strategies, the present study is focused on the process of building base classifiers over the OVA scheme. A novel split criterion, entitled by the splitting point correction matrix (SPCM), is proposed in this regards, which can effectively deal with the unbalance problem caused by the OVA scheme.

The SPCM is a kind of hybrid scheme, which integrates distribution and permutation information from the training data at each splitting point. Therefore, compared to other classical split criteria, such as the C4.5, the proposed method can make the right choice about the optimal splitting point at the root or internal nodes from multi-angle.

In order to evaluate the effectiveness of the SPCM approach, extensive experiments are carried out compared to the classical and state-of-the-art methods. The experiments are performed on sixteen datasets, where the effectiveness and the accuracy of the proposed method is verified. It is concluded that the SPCM method not only has excellent classification performance but also produces a more compact decision tree. Moreover, it is found that the SPCM method has especially a considerable improvement in the depth of tree.

1. Introduction

Classification problems are one of the most classical problems of data mining communities and have been widely studied in various fields [1–5]. A classifier is generated by a learning function over the training set. Then it performs on a new example in turn to predict the corresponding class label. Usually, classification problems are divided into binary and multi-class problems based on the number of involved classes in the classification process. Literally, the binary classification problems consider those problems between pair of classes. On the other hand, the multi-class classification problems refer to consider more than two classes. It should be indicated that a multi-class classification problem is intrinsically more complex than a binary problem. Because the generated classifier must be able to separate the data into a higher number of classes, which increases the chance of classification errors.

Decision trees have been proved to be powerful and popular approaches in the data science for discovering useful models. Since applying the tree structure approaches the classification rules to human reasoning, decision trees are well inter interpretability. Constructing a decision tree is usually a recursive procedure, where it repeatedly optimizes a function and partitions the training data in the root and internal nodes until a termination condition is met. This function is usually referred as *split criterion* [6]. Reviewing the literature indicates that several classical split criteria, including ID3 [7], C4.5 [8] and classification and regression tree (CART) [9] have been proposed. Among the aforementioned criteria, ID3 and C4.5 are based on the information entropy, while the CART adopts the Gini index. It should be indicated that some other split criteria are also introduced, but they are not classified as independent schemes [10–15].

Decision trees have been proved to be powerful and popular approaches in the data science for discovering useful models. Since applying the tree structure approaches the classification rules to human reasoning, decision trees are well inter interpretability. Constructing a decision tree is usually a recursive procedure, where it repeatedly optimizes a function and partitions the training data in the root and internal nodes until a

* Corresponding authors.
E-mail addresses: jackyan@stu.xmu.edu.cn (J. Yan), zhongnan_zhang@xmu.edu.cn (Z. Zhang), khlin@xmu.edu.cn (K. Lin), yangfan@xmu.edu.cn (F. Yang), xbluo@xmu.edu.cn (X. Luo).

https://doi.org/10.1016/j.knosys.2020.105922
0950-7051/

termination condition is met. This function is usually referred as *split criterion* [6]. Reviewing the literature indicates that several classical split criteria, including ID3 [7], C4.5 [8] and classification and regression tree (CART) [9] have been proposed. Among the aforementioned criteria, ID3 and C4.5 are based on the information entropy, while the CART adopts the Gini index. It should be indicated that some other split criteria are also introduced, but they are not classified as independent schemes [10,11,13–15].

Different decomposition strategies can be found in the literature [16], while the most common strategies are called one-vs-one (OVO) [17] and one-vs-all (OVA) [18]. These two strategies are unified within the error correcting output code (ECOC) [19]. The present study only focuses on the OVA strategy, which divides the multi-class problem into as many binary problems as the number of the classes. Then, each classifier is learnt to discern the examples of one class from the examples of the remaining classes.

Recently, studies on the OVA have been concentrated on aggregation strategies, which investigate how to combine the outputs of the base classifiers [20,21]. However, they ignored the class imbalanced problem that leads to the inferior classification performance of the OVA system [22]. This severe drawback has been criticized by many researchers [23–25]. On the other hand, the class imbalanced problem is inherent in the OVA decomposition scheme; that is, examples of one class are considered as *positive* $(+1)$, while examples from the remaining classes are regarded as *negative* (-1) in the training set. The OVA strategy is applied on the data preprocessing stage. Therefore, the imbalanced problem has a substantial impact on building the base classifiers and reduces to have a bad effect on the final classification performance of the OVA system. In spite of this, Rifkin and Klautar [26] fine-tuned the classifiers and showed that the OVA strategy is as accurate as any other approaches.

In the present study, a new approach is proposed to build decision trees and solves the class imbalanced problem. The proposed approach is called splitting points correction matrix (SPCM), which is analogous to the error correction output codes (ECOC). Because both schemes have the common characteristic of correcting errors. The former is utilized to correct the irrational splitting point in the learning phase, while the latter has a certain tolerance and correction effect on the error outputs of classifiers in the validation stage. It should be indicated that the proposed SPCM scheme is a hybrid one, which combines the distribution information and the permutation information from training data to determine the optimal splitting point. Compared to other split criteria, e.g., C4.5 and DCSM [27], which only consider a part of information of training data, the SPCM method has corrective effect in choosing the optimal splitting point. The contributions of the present study can be summarized as the following:

- This is the first application of the OVA decomposition strategy on the class imbalanced problem for multi-class classification tasks.
- A novel split criterion is proposed, which can effectively address the class imbalance problems originating from the OVA strategy.
- Comparison to previous split criteria, which determine the optimal splitting point from a single perspective, the SPCM scheme initially evaluates a splitting point from multi-angle viewpoint. Therefore, the SPCM approach can effectively correct errors made by other split criteria.
- The experimental results show that, compared to other split criteria, the decision tree introduced by the SPCM method not only achieves excellent classification accuracy, but has a much smaller depth in most data sets.

The rest of this paper is organized as follows. In Section 2, we recall some decomposition techniques for dealing with multi-class problems and describes basic concepts of decision trees. Next, Section 3 introduces our proposed Splitting Points Correction Matrix model in detail. The experimental framework set-up is presented in Section 4. We carry out the experiments that validate the propositions and compare them with the state-of-the-art methods in Section 5. Finally, we make our concluding remarks in Section 6.

2. Literature survey

In this section, two decomposition techniques are introduced initially for solving the multi-class problems. Then, the one-vs-one (OVA) strategy is described. Finally, the decision trees and researches in the field of the imbalanced scenarios are introduced.

2.1. Decomposition for multi-class classification problems

The multi-class classification problems are intrinsically complex which are due to various factors, including separability of classes, class overlaps and between-class and within-class imbalance. Although, there are some learners, which directly manage multiple classes [28,29]. However, their classification performance is not satisfactory. In contrast to the multi-class classifiers, the binary classifiers often easily find the decision boundary to distinguish between two classes. The binarization techniques of using multiple binary subtasks instead of directly tackling a multiple classes problem have attracted extensive attention.

The OVO and OVA decomposition strategies are known to be the most common approaches. The former consists of learning a binary classifier to distinguish each pair of classes, whereas the latter is composed of multiple binary classifiers and each one is used to separate each single class from the other classes. The simplest combination is the application of the voting strategy, in which each classifier gives a vote for a class and therefore, the largest number of votes is given as output (in OVA only the positive output is used for a classifier). Allwein et al. [30] proposed a unifying framework integrating both approaches where they are encoded within a code-matrix. Then a new example is submitted to the classifiers and their outputs are encoded a code-word to obtain the final prediction by comparing with the code-words in the code-matrix based on an Error Correcting Output Code [31]. Many proposals are studied regarding ECOC, where automatic design of the code-matrix [32] is explored and different error correcting codes [33] are used.

2.2. One-vs-all decomposition strategy

The OVA strategy produces m binary problems for an m class problem and each for one class. Each problem is done by a binary classifier, which is responsible for identifying one class from all the others. Consequently, the whole training data is used in learning phase, considering the examples of a class as positive and the examples of the remaining classes as negative. In the validation stage, all unknown examples are submitted to all of the classifiers. Normally, each example performs all of the binary classifiers and produces a corresponding positive output, which indicates its class label. However, in many cases, the positive output is not unique due to classification error and some tie-breaking techniques are required. Instead of the positive output, the confidence of classification is an alternative, which declares the class label of the new example from a more microscopic perspective. The outputs of classifiers are sorted in a score vector

(where $r_i \in [0, 1]$ is the confidence for class i produced by *ith* classifier), which presents as follows:

$$R = \{r_1, r_2, \ldots, r_i, \ldots, r_m\} \quad (1)$$

Then, the final prediction is the class label of the classifier giving the largest confidence.

Furthermore, Hong et al. [34] proposed a method considering a dynamic order of classifiers to bypass the ties using a prior by the Naive Bayes classifier. Guan et al. [21] proposed a multi-view OVA model (MVDT), which expressed the leaf node in a probabilistic fashion using all of the classes in training data not only for positive leaf nodes but also for all of the negative leaf nodes. When validating a new example, the final output of this model is the class label concerned with the largest probability after summing up the outputs of all the classifiers. The more details of this aggregation strategy is referred as [21]. However, all these methods focus on the validation phase. In addition, it should be noted that the OVA scheme inevitably gives rise to the class imbalanced problem [22]. For this reason, the lower classification performance of the OVA system comparing with the OVO is usually attributed to such an important difficulty [25]. Since it is well-known that the class imbalanced problem set usually causes some side effects on the derived classifiers.

2.3. Decision trees

Decision trees are one of the fundamental learning algorithms in the data mining community, which have been successfully applied to multi-class classification problems. The decision tree is a hierarchical structure and consists of three types of nodes, such as the root, internal and leaf nodes. In general, building a decision tree is a recursive partition program, starting from the root node (including all training data) and terminating at the leaf nodes by repeatedly optimizing a split criterion, which widely uses impure measurements [35]. Once a decision tree is built, an unseen example can be tested by flowing through the best path from the root to the leaf node. The final class label of the test example is determined by the representation of the reached leaf node.

Normally, the split criteria are not designed to solve imbalanced scenarios, so they tend to yield unsatisfactory performance. One way of solving the class imbalanced problem is to modify the class distribution of the training data by re-sampling [36,37], then a standard method is performed on training a classifier. Besides, there are some specifically designed "imbalanced data oriented" algorithms, which can perform well on the original unmodified imbalanced data sets. Liu et al. [38] proposed an insensitive skew measurement for the decision tree (CCPDT), which uses class proportion rather than Shannon's entropy [39]. The CCPDT exhaustively searched for each attribute and selected the optimal splitting point based on the best class confidence proportion. Cieslak et al. [40,41] utilized the Hellinger distance as a split criterion and the authors proved that the Hellinger distance is less sensitive to the class imbalanced problem. Recently, a new impurity measurement called minority entropy is proposed, which is based on the information of the minority class [42].

3. Splitting points correction matrix (SPCM) based decision trees

In this section, the splitting points correction matrix (SPCM) is introduced. The proposed scheme combines the distribution information with permutation information from training data for each splitting point at the root or internal nodes. First, the Hellinger distance is initially discussed. Then, the permutation information is studied and a new concept of split ratio is introduced. Moreover, the mathematical description of the SPCM approach is derived. Finally, an illustrative example is presented to show how the SPCM technique works. Fig. 1 shows the schematic framework of the OVA decision trees based on the SPCM. The boundaries of the SPCM scheme is specified by the blue dashed box. Throughout this section, the proposed hybrid scheme is demonstrated and only the binary decision tree is considered in the case of the OVA strategy.

3.1. Hellinger distance

The Hellinger distance is a measurement technique of the distributional divergence [43], which was first introduced to the decision tree as a splitting criterion in [40]. A splitting point with the maximum Hellinger distance is identified as the optimal one. For classification tasks, it can be formulated by considering the true positive rate *tpr* and the false positive rate *fpr* of some sets of label assignments. The Hellinger distance [41,44] is computed as the following:

$$d_H(tpr, fpr) = \sqrt{(\sqrt{tpr} - \sqrt{fpr})^2 + (\sqrt{1 - tpr} - \sqrt{1 - fpr})^2} \quad (2)$$

Where the $(\sqrt{tpr} - \sqrt{fpr})^2$ and $(\sqrt{1 - tpr} - \sqrt{1 - fpr})^2$ denote corresponding Hellinger distances for two subsets, which is split by a splitting point. Eq. (2) indicates that if these two subsets are pure notes, the Hellinger distance yields the maximum value so that $d_H = \sqrt{2}$. On the other hand, if the two subsets have the same number of positive and negative examples in each subset, the Hellinger distance results in the minimum value, where $d_H = 0$.

However, the Hellinger distance is unable to solve the splitting points competition problem. Studies [45,46] showed that there are two inherent factors in this regards:

1. As a splitting criterion, the Hellinger distance regarded is a heuristic method, which only considers a local optimum value by measuring the split quality at the root or internal nodes.
2. It is difficult to make the right choice when there are some splitting points with the same or similar maximal distance.

Therefore, if one can further excavate additional information of splitting points, it may avoid selecting the unreasonable splitting point. To the best of our knowledge, few researchers investigated this topic. Wang et al. [46] proposed the segment-based algorithm, which selected the optimal splitting point by means of the minimum number of expected segments from the candidate splitting points. Moreover, Yan et al. [47] introduced a hyper-parameter and presented a unified framework, which integrated information gain ratio and the number of expected segments to determinate the optimal point between several candidate splitting points. Although they have improved the classification performance, none of them is suitable for the class imbalanced scenarios.

Definition (*Splitting Points Competition*)**.** The existence of more than two candidate splitting points with the same or similar split quality, based on any of the split criterion, (e.g., the information gain ratio) makes the splitting point selection difficult. In this case, the correlation of these splitting points is referred to as splitting points competition.

3.2. Segments and split ratio

Suppose that $\mathbb{S} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is a training set, where $x_i \in \{a_1, a_2, \ldots, a_m\}$ denotes the ith example and

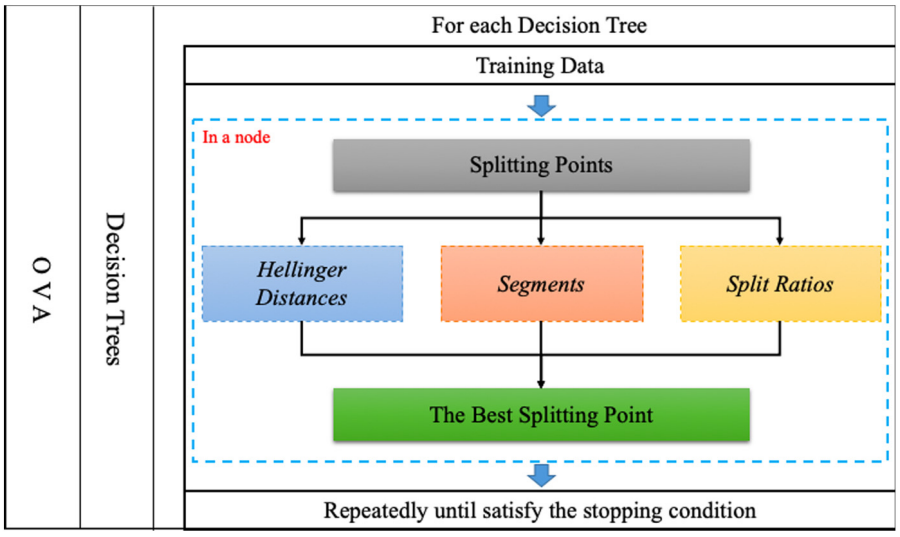


Fig. 1. Schematic framework of the OVA decision trees based on the SPCM method.

Labels:	-1	-1	+1		+1	-1	-1		-1	+1	-1
Attribute Values:	6	6	6		6	6	6		6	6	6
		(a)				(b)				(c)	

Fig. 2. Three possible permutations from different sorting algorithms for a bar with three examples.

$y_i \in \{c_1, c_2, \ldots, c_L\}$ is the class label of x_i. The permutation information means that a class label sequence is corresponding to a sorted example sequence with respect to a certain attribute. Throughout this work, it is mentioned that the permutation information of the training set refers to the ascending order of the attribute values. Table 1 shows an example about the permutation information. Elomma and Rousu [48] manifested that the optimal splitting point usually lies in *segment borders*. They showed that the obtained optimal point satisfies the well-behavedness, which is one of the most important properties of the split criteria [49, 50]. In other words, if there are less segments in internal nodes, it is more effective to build a decision tree with the same discriminative capability. An example sequence $\mathbb{L}$ is referred as a *segment* if it satisfies the following conditions:

- All the examples in $\mathbb{L}$ have the same class label.
- The class label of the leftmost example in $\mathbb{L}$ is different from that of its left example (if any).
- The class label of the rightmost example in $\mathbb{L}$ is different from that of its right example (if any).

Considering the concept of the *segment*, it is not hard to count the segments if all attribute values of the examples are different. However, for duplicated values, where each of them is corresponding to multiple examples, the concept of *bar* and the method of calculating the number of segments [46] is followed. Supposing that the training set $\mathbb{S}$ is sorted in an ascending manner by attribute a_i, t and u are the number of *bar* and *non-bar* sub-sequences in $\mathbb{S}$, respectively. Therefore, the number of segments in $\mathbb{S}$ is defined as the following:

$$Seg(\mathbb{S}, a_i) = \sum_{j=1}^{u} \|Seg(\mathfrak{S}_j, a_i)\| + \sum_{k=1}^{t} \|bSeg(\mathfrak{B}_k, a_i)\| \tag{3}$$

Where $\mathfrak{S}_j$ and $\mathfrak{B}_k$ are the jth *non-bar* sub-sequence and the kth *bar*, respectively. Moreover, $\|Seg(\mathfrak{S}_j, a_i)\|$ and $\|bSeg(\mathfrak{B}_k, a_i)\|$ denote the number of segments of $\mathfrak{S}_j$ and $\mathfrak{B}_k$. It should be indicated that the following three key points should be considered in each *segment*:

- The *segment* is only related to the class label. Considering the *segment* concept, it is insensitive to the class imbalanced problem. This is valid for the number of segment.
- It is easy to understand that different permutations of the training set may yield different number of segments. Therefore, the training data should be sorted with a certain attribute prior to calculate the number of segments. Otherwise, it is impossible to obtain a unique result.
- The number of segments is independent of any particular sorting algorithm. In other words, the same number of segment is obtained with respect to the permutation information regardless of the sorting algorithm. According to Eq. (3), the segment value is the sum of the segments of non-bars and bars. Therefore, only the number of segment in bar may change for different sorting algorithms. Fig. 2 shows three possible permutations for the bar with the attribute value 6 in Table 1. It seems that there are three segments in Fig. 2(c), while others have only two segments. In fact, the permutation with the maximal segment value is only considered, so that it is robust to the sorting algorithms.

An example is given to show how the number of segment is calculated in a node. Suppose that $\mathbb{S}$ is the sorted example set of a node, and Table 1 shows its permutation information. From Table 1, it can observe that there are 3 bars (with the attribute values of 2, 3 and 6) and 3 non-bar sub-sequences. Therefore, the number of segment in the $\mathbb{S}$ can be calculated as:

$$\begin{aligned} Seg(\mathbb{S}, a_i) =& \|bSeg(\mathfrak{B}_1, a_i)\| + \|bSeg(\mathfrak{B}_2, a_i)\| + \\ & \|bSeg(\mathfrak{B}_3, a_i)\| + \|Seg(\mathfrak{S}_1, a_i)\| + \\ & \|Seg(\mathfrak{S}_2, a_i)\| + \|Seg(\mathfrak{S}_3, a_i)\| \\ =& 1 + 1 + 3 + 1 + 2 + 5 \\ =& 13 \end{aligned}$$

Moreover, a concept of *split ratio* is proposed, which can be defined as the following:

Definition (*Split Ratio*)**.** Suppose that a training set $\mathbb{S}$, an attribute a_i and a splitting point T are given. Let $\mathbb{S}_1 \subset \mathbb{S}$ be a subset of $\mathbb{S}$ with attribute values $\leq$ T and $\mathbb{S}_2 = \mathbb{S} - \mathbb{S}_1$. Then the *split ratio*

Table 1
A binary classes dataset sorted in ascending order by a certain attribute.

Examples	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Labels	-1	-1	-1	-1	-1	$+1$	-1	-1	-1	$+1$	-1	-1	$+1$	-1	$+1$	-1	-1	-1	-1	-1
Attribute values	1	2	2	3	3	4	5	6	6	6	7	8	9	10	11	12	13	14	15	16

induced by T, namely $Ratio(\mathbb{S}, a_i, T)$, is defined as the following:

$$Ratio(\mathbb{S}, a_i, T) = \begin{cases} \dfrac{Seg(\mathbb{S}_2, a_i)}{Seg(\mathbb{S}_1, a_i)}, & Seg(\mathbb{S}_1, a_i) > Seg(\mathbb{S}_2, a_i) \\ \dfrac{Seg(\mathbb{S}_1, a_i)}{Seg(\mathbb{S}_2, a_i)}, & Seg(\mathbb{S}_1, a_i) \leq Seg(\mathbb{S}_2, a_i) \end{cases} \tag{4}$$

Where $Seg(\mathbb{S}_1, a_i)$ and $Seg(\mathbb{S}_2, a_i)$ represent the number of segment in $\mathbb{S}_1$ and $\mathbb{S}_2$ regarding attribute a_i, respectively. The purpose of Eq. (4) is to ensure that the value of *split ratio* is range in (0, 1]. The conductivity indicates that the *split ratio* is also insensitive to the unbalance problem.

3.3. Splitting points correction matrix

The optimal splitting point is determined based on the following idea: if there are more information about splitting points in a node, this may be conducive to make the right choice. Therefore, a hybrid split criterion, called the splitting points correction matrix (SPCM), is proposed in this study. The SPCM scheme synthesizes *Hellinger distance*, *segment* and *split ratio* to evaluate the optimal splitting point. The splitting points correction matrix is formulated as the following:

$$\begin{bmatrix} hd_1 & segments_1 & ratio_1 \\ hd_2 & segments_2 & ratio_2 \\ \cdots & \cdots & \cdots \\ hd_N & segments_N & ratio_N \end{bmatrix} \tag{5}$$

Where hd_i, $segments_i$ and $ratio_i$ represent the Helinger distance, the sum of segments from two subsets and the split ratio of the ith splitting point, respectively. Moreover, the subscript N denotes the number of splitting points in a node. In other words, the SPCM resemble to a pool, where it aggregates information of all the splitting points at each internal node. The procedure of determining the optimal splitting point is as following:

1. A splitting points correction matrix is constructed, where each row consists of the Hellinger distance, sum of segments of left and right subsets and the split ratio with respect to a splitting point.
2. The matrix is sorted in a descending order by the Hellinger distance and then a truncated matrix is obtained, which consists of the first $\hat{K}$ rows in the sorted matrix.
3. The splitting point is selected in the truncated matrix as the optimal one, which has the smallest segment and highest split ratio.

It is found that the proposed criteria has a significant strength compared to other split criteria. Because it utilizes more information to estimate the best certain splitting point. As a result, the proposed method can correct the mistake choices, which may be made by other split criteria. An illustrative example is given in Section 3.4 to describe the operational concept of the SPCM.

3.4. Illustrative examples

In this section, an illustrative example is presented to show how the splitting points correction matrix works. Let $\mathbb{S}$ be a node with twenty examples from two classes, e.g. $\{+1, -1\}$ in accordance with Table 1. Considering the small data size, $\hat{K}$ is set to 3, which shows the number of rows in the truncated matrix. Since there are many possible splitting points at the node, only the truncated matrix is presented. According to Eq. (5), the results show as following:

$$\begin{bmatrix} \mathbf{3.5} \\ \mathbf{11.5} \\ 12.5 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{0.4419} & 13 & 0.3333 \\ \mathbf{0.4419} & 13 & 0.0909 \\ 0.3536 & 14 & 0.0833 \end{bmatrix} \tag{6}$$

Where the right hand matrix is the truncated matrix, which is sorted in a descending order by the Hellinger distance, and their corresponding splitting points lie in the left column vector. From the truncated matrix, it is observed that it is hard to make choice between 3.5 and 11.5, because they have the same maximum Hellinger distance. In this case, a random selection between them is performed to complete the selecting process. Nevertheless, if there are other information about these two splitting points, such as the split ratio, it may relieve from this dilemma. Considering the larger *split ratio* value, it is confirmed that 3.5 is the best choice. The selection procedure is illustrated in the following:

$$\begin{bmatrix} \mathbf{3.5} \\ \sout{11.5} \\ 12.5 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{0.4419} & \mathbf{13} & \mathbf{0.3333} \\ 0.4419 & 13 & 0.0909 \\ 0.3536 & 14 & 0.0833 \end{bmatrix} \tag{7}$$

Consequently, the proposed method corrects the mistake aroused by applying only the maximum Hellinger distance.

4. Experimental design

In this section, the setup of the experimental framework is described. These experiments are designed to answer the following four issues:

- Does the proposed method outperform other methods regarding multi-class classification tasks?
- How does the parameter $\hat{K}$ affect the classification performance of the OVA system and the scale of decision trees?
- Does the OVA system consisting of the proposed method precede the OVO systems ?
- Is the proposed method more effective than the *Segment* [46] combining with the OVA strategy, which has a similar way about choosing the optimal splitting point?

The following subsections present the used benchmark data sets, the detailed algorithms and their parametric configuration and evaluation metrics.

4.1. Data sets

In the present study, twenty diverse multi-class datasets are selected from the machine learning repository of the university of California, Irvine (UCI). Table 2 shows details of the data sets, including the number of examples (*#Ex.*), attributes (*#Atts.*) and categories (*#Cl.*). Fig. 3 shows the class distribution of all the data sets. Each bar represents the class distribution of a data set and each color in a bar is the proportion of a class in the data set. From Fig. 3, it is easy to observe the class imbalance of each data set in the case of OVA decomposition strategy. In Table 2, all of the values in brackets are the original information, while others are the actual data information being utilized. For example, the corresponding number of examples, number of continuous attributes, number of classes, the maximum and the minimum imbalanced ratios for *ecoli* are 327, 5, 5, 1.86 and 7.15, respectively. The data set is preprocessed, prior to training the classifier. The preprocessing steps are as the following:

Table 2
Details of datasets used in the experiments.

No.	Dataset	#Ex.	#Atts.	#Cl.	Segment	SPES		Multi-Segment	SPCM
					$\hat{K}$	$\hat{K}$	α	$\hat{K}$	$\hat{K}$
1	vowel	990	10(13)	11	2	8	0.9	10	2
2	libras	360	90	15	2	10	0.9	15	2
3	yeas	1479(1484)	6(8)	9(10)	10	15	0.8	15	4
4	wine	178	12	3	4	2	0.9	6	2
5	thyroid	215	5	3	20	2	0.9	4	10
6	iris	150	4	3	4	2	0.9	2	10
7	vehicle	846	18	4	15	2	0.9	2	4
8	ecoli	327(336)	5(7)	5(8)	4	8	0.1	4	10
9	automobile	156(159)	16(25)	5(6)	2	2	0.9	8	2
10	segment	2310	16(19)	7	15	8	0.9	8	2
11	tae	151	3(5)	3	2	20	0.8	2	6
12	cleveland	297	5(13)	5	10	6	0.9	6	15
13	image	6435	36	6	20	8	0.8	8	15
14	waveform	5000	21	3	20	10	0.8	8	4
15	seeds	210	7	3	2	6	0.8	2	2
16	winequality_red	1589(1599)	11	5(6)	20	2	0.5	15	8
17	glass	214	9	6	15	15	0.6	2	20
18	penbased	5493	16	10	10	6	0.4	8	10
19	texture	5500	40	11	2	8	0.9	10	8
20	winequality_white	4893	11	6	20	15	0.6	2	20

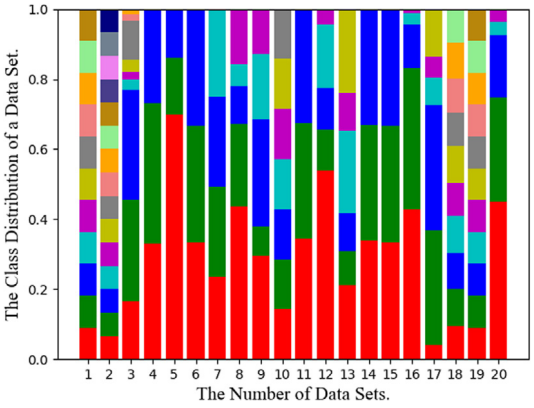


Fig. 3. The class distribution for all data sets. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Remove the classes with less than ten examples from the corresponding data set. There are two reasons to do this. First, it is hard to conduct cross-validation after OVA decomposition, if the example numbers of some classes are very small. The other reason is that it is not meaningful to train a classifier with fewer positive examples, but remaining examples are *negative*. For example, there are only 2 examples for two classes in *ecoli*.
- In the present study, all the experiments should be conducted on the data sets with continuous attributes. Considering these data sets, not all the attributes are continuous. Therefore, before conducting experiments, some symbol attributes and continuous attributes with less than ten unique attribute values are removed from their corresponding data set.
- For convenience, normalization operator has been applied to all the attribute values before training classifiers by using $1 - ((v_{max} - v)/(v_{max} - v_{min}))$ so that all the values range in [0,1]. It should be indicated that v_{max} and v_{min} are the maximum and minimum attribute values in an attribute, respectively. Moreover, v is the attribute value that needs to be normalized.

For a fair comparison between different methods, in this paper, all the experiments are carried out under the same conditions; that is to say, the same preprocessing has been done for all the data sets before they were used for training by each method. Additionally, twenty iterations of 2-fold cross-validation are run in the present study. In other words, the data set is divided into 2 folds, each one containing 50% of the examples of the data set. For each fold, the algorithm is trained with examples contained in the remaining fold and then it is tested with the other fold.

4.2. Configuration of the algorithms and parameter

In order to clarify the proposed method, several popular and the state-of-the-art classifiers are selected. In the present study, the multiple tree of the MVDT [21] is replaced with the binary tree. While for the SPES [47] scheme, only the SPES2 scheme is used. Moreover, the OVA + HD indicates that the Hellinger distance is used as split criterion in the case of the OVA strategy. [21] is followed and the leaf node is expressed in a fashion of membership probability regardless of the positive or negative nodes for all of the methods. Thus, the sum of membership probabilities at each leaf node is 1.0. Due to the different data sets with various number of classes, the number of examples of stopping partitioning in a node is set $\hat{N} = 5$ when the number of a dataset classes is less than five or it is the number of classes if it is more than five. For the Segment [46], SPES and SPCM methods, the number of candidate splitting points is set $\hat{K} = \{2, 4, 6, 8, 10, 15, 20\}$. Additionally, the weighting factor is set to $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ for the SPES, which is used to combine the splitting performance and the number of expected segments. For the OVO strategy, $\hat{N}$ is set to 5 to avoid the under-fitting when the data set has a small size of examples but with large number of classes. For example, the *libras* has 360 examples and 15 classes. In this case, a new example is evaluated with the same aggregation strategy for the OVA and OVO schemes [21]. In comparison with the Segment [46] scheme in the splitting point selection problem, the Segment is slightly modified by putting it in the OVA case. This method is called the "Multi-Segments". All algorithms and their configuration parameters are listed in Table 3.

All of the experiments are performed on Python 3, and executed on a computer with a 3.20 GHz Intel® Core(TM) i5-6500 CPU, a 4.00 GB memory and 64 bit Windows 7 system.

Table 3
Different classifiers with the corresponding parameters.

Algorithm	Parameters
C4.5[51]	$\hat{N} = 5$ or the number of classes
MVDT [21]	$\hat{N} = 5$ or the number of classes
OVA + HD	$\hat{N} = 5$ or the number of classes
Segment [46]	$\hat{N} = 5$ or the number of classes
	$\hat{K} = \{2, 4, 6, 8, 10, 15, 20\}$
SPES [47]	$\hat{N} = 5$ or the number of classes
	$\hat{K} = \{2, 4, 6, 8, 10, 15, 20\}$
	$\hat{\alpha} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
OVO [52] + C4.5	$\hat{N} = 5$
Multi-Segment	$\hat{N} = 5$ or the number of classes
	$\hat{K} = \{2, 4, 6, 8, 10, 15, 20\}$
OVA + SPCM	$\hat{N} = 5$ or the number of classes
	$\hat{K} = \{2, 4, 6, 8, 10, 15, 20\}$

4.3. Evaluation metrics

In order to analyze the performance of different methods, the proposed method is evaluated from the aspects of classification accuracy and model complexity. The accuracy is computed as the number of correctly classified examples relative to the total number of the test set. The model complexity is the scale of decision tree, which is expressed by the number of nodes and depth of the decision tree. Moreover, considering the OVA strategy, which reduces the multi-class data set into the binary imbalanced scenarios, the F-measure (F1) [53] and the MAUC [54] are adopted as evaluation measurements.

Let $\mathcal{C}(i, j)$ be the number of examples of c_i being classified to c_j. Then the precision P_i and recall R_i of c_i are define as:

$$P_i = \frac{\mathcal{C}(i,j)}{\sum_{j=1}^{L} \mathcal{C}(i,j)} \tag{8}$$

$$R_i = \frac{\mathcal{C}(i,j)}{n_i} \tag{9}$$

In the multi-class scenario, F-measure (F1) is defined as:

$$F_i = \frac{2P_iR_i}{P_i + R_i}, \quad F - measure = \frac{1}{L}\sum_{i=1}^{L} F_i \tag{10}$$

Where F_i, P_i and R_i are the F-measure of the ith classifier, classifier precision and the recall of ith classifier, respectively. It should be indicated that the average accuracy of the k classifiers is utilized as the multi-class F-measure score and the MAUC is defined as the following:

$$MAUC = \frac{2}{L(L+1)}\sum_{i<j} auc(i,j) \tag{11}$$

Where $auc(i, j)$ is the AUC of i and j classes.

5. Experimental results and analysis

In this section, the efficiency of the proposed method is studied. To do so, the comparison of the SPCM approach with the best parameters is performed with other methods in multi-class tasks. Moreover, this section is divided into three parts, which respectively correspond to the four questions mentioned in Section 4. The proposed approach is initially compared with other methods, such as the direct classification and the OVA decomposition schemes for the first issue. Then, the influence of the hyper-parameter $\hat{K}$ is discussed on the classification accuracy and complexity of decision tree for the second issue. Afterwards, an analysis between the proposed method and the OVO strategy is performed for the third issue. Finally, the splitting points competition problem between the Segment [46] and the SPCM in the OVA strategy is studied for the fourth issue.

5.1. Experiment 1: The effectiveness of the splitting points correction matrix

The first objective of this part is to study the effectiveness of the proposed SPCM scheme compared with other methods in multi-class scenarios. Tables 4 to 8 respectively show the experimental results in accuracy, tree scale, F1 score, AUC and training time for all the methods. All these results (except for tree scale and time comparison) are presented by average (± standard deviation) and the best parameters involved methods are listed in Table 2. In addition, the best result for each pair of method and data set is stressed in **bold-face**.

Table 4 shows the comparison between all the methods in accuracy for all the data sets. One can observe that the proposed method significantly outperforms other methods in sixteen out of twenty data sets. From Table 4, it is found that the accuracy of methods has been improved when the OVA strategy is applied. This suggests that the OVA strategy is a useful treatment to improve the classification performance of multi-class tasks. Additionally, the OVA + HD and SPCM schemes with the OVA

Table 4
The average accuracy of different methods. The highest accuracy is determined by the bold-face.

Dataset	C4.5	MVDT	OVA + HD	Segment + C4.5	SPES	SPCM
vowel	0.6562 ± 0.0193	0.6806 ± 0.0299	0.7077 ± 0.0285	0.6596 ± 0.0267	0.6679 ± 0.0239	**0.7267 ± 0.0329**
libras	0.4983 ± 0.0409	0.3578 ± 0.0458	0.5350 ± 0.0461	0.5117 ± 0.0500	0.5128 ± 0.0468	**0.5411 ± 0.0473**
yeast	0.5023 ± 0.0204	0.5418 ± 0.0142	0.5629 ± 0.0194	0.5233 ± 0.0214	0.5173 ± 0.0351	**0.5647 ± 0.0221**
wine	0.9202 ± 0.0371	0.9191 ± 0.0392	0.9011 ± 0.0645	0.9067 ± 0.0564	**0.9225 ± 0.0142**	0.9067 ± 0.0402
thyroid	0.9140 ± 0.0301	0.9037 ± 0.0318	0.9374 ± 0.0254	0.9327 ± 0.0232	0.9159 ± 0.0536	**0.9430 ± 0.0184**
iris	0.9413 ± 0.0171	0.9427 ± 0.0292	0.9453 ± 0.0234	0.9480 ± 0.0173	0.9453 ± 0.0152	**0.9487 ± 0.0207**
vehicle	0.6851 ± 0.0257	0.6915 ± 0.0164	0.7019 ± 0.0153	0.6967 ± 0.0150	0.6936 ± 0.0203	**0.7175 ± 0.0224**
ecoli	**0.7883 ± 0.0232**	0.7804 ± 0.0375	0.7785 ± 0.0257	0.7767 ± 0.0305	0.7847 ± 0.0182	0.7729 ± 0.0335
auto	0.6897 ± 0.0578	0.6308 ± 0.0540	0.6692 ± 0.0593	0.6654 ± 0.0539	**0.7103 ± 0.0327**	0.6266 ± 0.0836
segment	0.9474 ± 0.0080	0.9429 ± 0.0091	0.9486 ± 0.0089	0.9498 ± 0.0064	0.9502 ± 0.0104	**0.9571 ± 0.0073**
tae	0.4880 ± 0.0410	0.5040 ± 0.0421	0.4693 ± 0.0742	0.5013 ± 0.0311	**0.5160 ± 0.0301**	0.5013 ± 0.0431
cleveland	0.4182 ± 0.0303	0.5034 ± 0.0250	0.5297 ± 0.0216	0.4703 ± 0.0413	0.4534 ± 0.0312	**0.5365 ± 0.0350**
image	0.8327 ± 0.0058	0.8488 ± 0.0075	0.8652 ± 0.0047	0.8369 ± 0.0054	0.8332 ± 0.0116	**0.8661 ± 0.0040**
waveform	0.7222 ± 0.0073	0.7364 ± 0.0065	0.7611 ± 0.0076	0.7363 ± 0.0124	0.7252 ± 0.0217	**0.7618 ± 0.0081**
seeds	0.9095 ± 0.0234	0.8819 ± 0.0381	0.8895 ± 0.0277	0.8895 ± 0.0242	**0.9181 ± 0.0212**	0.9114 ± 0.0369
wine_red	0.5453 ± 0.0161	0.6040 ± 0.0154	0.6309 ± 0.0134	0.5583 ± 0.0205	0.5568 ± 0.0134	**0.6390 ± 0.0106**
glass	0.6091 ± 0.0405	0.4510 ± 0.0115	0.6735 ± 0.0372	0.6098 ± 0.0293	0.6618 ± 0.0335	**0.6833 ± 0.0423**
penbased	0.9261 ± 0.0049	0.9205 ± 0.0127	0.9408 ± 0.0080	0.9236 ± 0.0054	0.9311 ± 0.0021	**0.9622 ± 0.0052**
texture	0.8952 ± 0.0032	0.9001 ± 0.0075	0.9124 ± 0.0007	0.8989 ± 0.0022	0.9044 ± 0.0053	**0.9364 ± 0.0075**
wine_white	0.5224 ± 0.0026	0.4655 ± 0.0031	0.4980 ± 0.0131	0.5271 ± 0.0035	0.5334 ± 0.0001	**0.6063 ± 0.0082**

Table 5
The average results of the number of nodes and depths of decision trees for each method.

DataSet	C4.5		MVDT		OVA + HD		Segment		SPES		SPCM	
	Nodes	Depth	Nodes	Depth	Nodes	Depth	Nodes	Depth	Nodes	Depth	Nodes	Depth
vowel	190.20	42.10	43.58	14.45	**27.95**	9.45	188.00	43.30	184.00	41.00	29.25	**8.80**
libras	71.00	20.00	16.73	6.68	**9.44**	**4.02**	72.00	22.80	70.20	21.10	9.53	4.14
yeast	372.60	71.30	111.60	27.82	86.27	14.32	259.40	32.70	346.60	60.00	**77.02**	**12.18**
wine	11.80	4.00	11.47	3.73	**7.20**	**3.07**	12.00	4.30	13.00	4.10	7.67	3.23
thyroid	14.40	6.50	11.67	4.23	**8.53**	**3.43**	20.80	5.70	15.60	5.90	12.73	4.60
iris	9.20	4.00	9.07	2.77	**6.00**	**2.40**	8.80	3.40	10.40	3.40	11.67	4.10
vehicle	177.20	33.20	92.55	28.75	**65.85**	14.48	154.00	28.80	165.00	30.90	71.00	**12.73**
ecoli	50.40	10.70	25.32	7.18	**21.12**	6.54	49.40	10.00	49.80	11.10	25.10	**6.46**
auto	41.80	15.90	18.88	6.50	13.24	4.86	37.00	11.60	40.20	13.80	**11.47**	**4.22**
segment	93.40	23.60	36.37	10.29	**22.66**	**6.29**	94.00	14.20	91.00	22.30	23.97	6.51
tae	64.20	25.50	45.87	17.43	**39.00**	11.97	63.00	24.40	61.60	25.50	40.67	**9.57**
cleveland	98.20	26.70	48.00	17.18	**40.48**	11.38	83.00	14.70	96.20	23.20	48.28	**9.70**
image	659.40	69.70	240.73	51.85	**144.07**	17.23	546.40	40.50	618.80	64.50	180.77	**14.72**
waveform	801.40	135.90	494.40	75.47	**305.40**	19.10	678.20	101.00	778.60	121.80	331.67	**17.20**
seeds	15.80	6.00	13.47	4.73	**9.40**	**3.87**	16.20	6.00	17.60	5.70	11.40	4.47
wine_red	490.60	128.00	236.24	83.22	**142.96**	17.70	411.80	59.50	458.80	115.30	162.32	**15.68**
glass	47.50	15.10	21.40	6.80	**16.64**	**5.82**	48.00	14.40	42.40	10.70	18.04	5.92
penbased	265.50	27.25	77.90	13.00	**74.22**	10.59	239.50	18.75	246.50	26.00	75.72	**8.89**
texture	339.00	50.20	73.55	16.50	**35.00**	8.32	318.20	45.90	301.50	38.50	43.22	**8.03**
wine_white	1617.00	298.30	593.00	151.42	**381.53**	23.92	1544.00	257.00	1431.00	184.00	386.30	**23.55**

Table 6
The average F1 results for different methods.

DataSet	C4.5	MVDT	OVA + HD	Segment	SPES	SPCM
vowel	0.6558 ± 0.0187	0.6133 ± 0.0221	0.6815 ± 0.0209	0.6586 ± 0.0267	0.6675 ± 0.0121	**0.6943 ± 0.0118**
libras	0.4937 ± 0.0382	0.3459 ± 0.0449	0.5213 ± 0.0334	0.5067 ± 0.0505	0.5011 ± 0.0307	**0.5257 ± 0.0339**
yeast	0.3951 ± 0.0188	0.3806 ± 0.0210	0.4071 ± 0.0208	**0.4161 ± 0.0293**	0.4004 ± 0.0322	0.3820 ± 0.0290
wine	0.9190 ± 0.0393	0.9015 ± 0.0365	0.9033 ± 0.0427	0.9080 ± 0.0558	**0.9223 ± 0.0263**	0.9193 ± 0.0308
thyroid	0.8830 ± 0.0420	0.8574 ± 0.0515	0.9153 ± 0.0255	0.9085 ± 0.0295	0.8836 ± 0.0518	**0.9179 ± 0.0168**
iris	0.9410 ± 0.0160	0.9370 ± 0.0256	0.9408 ± 0.0230	**0.9481 ± 0.0164**	0.9453 ± 0.0084	0.9380 ± 0.0154
vehicle	0.6870 ± 0.0286	0.6819 ± 0.0121	0.6974 ± 0.0206	0.6950 ± 0.0149	0.6961 ± 0.0150	**0.6987 ± 0.0280**
ecoli	**0.7078 ± 0.0403**	0.6878 ± 0.0240	0.6811 ± 0.0357	0.6902 ± 0.0526	0.6993 ± 0.0541	0.6042 ± 0.0487
auto	0.6449 ± 0.0542	0.5780 ± 0.0514	**0.6554 ± 0.0769**	0.6199 ± 0.0560	0.6547 ± 0.0210	0.5483 ± 0.1112
segment	0.9471 ± 0.0079	0.9340 ± 0.0048	0.9540 ± 0.0066	0.9495 ± 0.0067	0.9499 ± 0.0073	**0.9555 ± 0.0054**
tae	0.4845 ± 0.0440	0.4979 ± 0.0462	0.4636 ± 0.0702	0.4993 ± 0.0316	**0.5148 ± 0.0129**	0.4903 ± 0.0384
cleveland	0.2352 ± 0.0353	0.2346 ± 0.0192	0.2623 ± 0.0351	0.2526 ± 0.0355	**0.2834 ± 0.0313**	0.2448 ± 0.0356
image	0.8052 ± 0.0055	0.8022 ± 0.0060	0.8276 ± 0.0050	0.8101 ± 0.0055	0.8073 ± 0.0140	**0.8300 ± 0.0069**
waveform	0.7224 ± 0.0072	0.7265 ± 0.0042	0.7529 ± 0.0063	0.7361 ± 0.0123	0.7254 ± 0.0196	**0.7549 ± 0.0062**
seeds	0.9086 ± 0.0242	0.8780 ± 0.0311	0.8931 ± 0.0246	0.8904 ± 0.0233	**0.9181 ± 0.0208**	0.8835 ± 0.0262
wine_red	0.3441 ± 0.0207	0.3646 ± 0.0128	0.3826 ± 0.0144	0.3486 ± 0.0197	0.3529 ± 0.0120	**0.3866 ± 0.0316**
glass	0.5521 ± 0.0455	0.5841 ± 0.0275	0.5905 ± 0.0668	0.5708 ± 0.0607	0.6062 ± 0.0205	**0.6260 ± 0.0413**
penbased	0.9262 ± 0.0049	0.8962 ± 0.0121	0.9287 ± 0.0056	0.9236 ± 0.0051	0.9310 ± 0.0053	**0.9486 ± 0.0020**
texture	0.8951 ± 0.0032	0.8867 ± 0.0003	0.9177 ± 0.0007	0.8987 ± 0.0019	0.9615 ± 0.0001	**0.9669 ± 0.0032**
wine_white	0.3575 ± 0.0066	0.3422 ± 0.0211	0.3723 ± 0.0077	0.3674 ± 0.0074	0.3699 ± 0.0064	**0.3839 ± 0.0124**

Table 7
The average AUC results for different methods.

DataSet	C4.5	MVDT	OVA + HD	Segment	SPES	SPCM
vowel	0.8506 ± 0.0145	0.9528 ± 0.0037	0.9614 ± 0.0063	0.8580 ± 0.0159	0.8615 ± 0.0135	**0.9661 ± 0.0052**
libras	0.7592 ± 0.0282	0.8824 ± 0.0210	0.9149 ± 0.0124	0.7826 ± 0.0314	0.7812 ± 0.0294	**0.9281 ± 0.0142**
yeast	0.7413 ± 0.0187	0.8630 ± 0.0131	0.8677 ± 0.0113	0.7823 ± 0.0235	0.7398 ± 0.0311	**0.8683 ± 0.0486**
wine	0.9458 ± 0.0231	0.9809 ± 0.0093	0.9717 ± 0.0239	0.9366 ± 0.0424	0.9511 ± 0.0386	**0.9851 ± 0.0203**
thyroid	0.9250 ± 0.0259	0.9372 ± 0.0320	0.9715 ± 0.0249	0.9304 ± 0.0332	0.9273 ± 0.0413	**0.9784 ± 0.0143**
iris	0.9776 ± 0.0049	**0.9848 ± 0.0080**	0.9760 ± 0.0143	0.9807 ± 0.0035	0.9768 ± 0.0029	0.9755 ± 0.0139
vehicle	0.8056 ± 0.0178	0.8851 ± 0.0091	0.8897 ± 0.0102	0.8251 ± 0.0070	0.8187 ± 0.0058	**0.8948 ± 0.0110**
ecoli	0.8518 ± 0.0161	**0.9239 ± 0.0121**	0.9205 ± 0.0165	0.8406 ± 0.0341	0.8575 ± 0.0176	0.9191 ± 0.0184
auto	0.8209 ± 0.0463	0.8610 ± 0.0434	0.8762 ± 0.0371	0.8151 ± 0.0360	0.8520 ± 0.0515	**0.8832 ± 0.0588**
segment	0.9751 ± 0.0044	0.9936 ± 0.0017	0.9950 ± 0.0027	0.9795 ± 0.0033	0.9767 ± 0.0027	**0.9951 ± 0.0017**
tae	0.6369 ± 0.0435	0.6443 ± 0.0515	0.6440 ± 0.0661	0.6608 ± 0.0310	0.6512 ± 0.0612	**0.6703 ± 0.0654**
cleveland	0.5417 ± 0.0262	0.5703 ± 0.0264	**0.5942 ± 0.0213**	0.5730 ± 0.0271	0.5822 ± 0.0301	0.5845 ± 0.0362
image	0.9189 ± 0.0030	0.9688 ± 0.0023	0.9731 ± 0.0010	0.9270 ± 0.0026	0.9207 ± 0.0018	**0.9755 ± 0.0022**
waveform	0.7915 ± 0.0062	0.8878 ± 0.0040	0.8994 ± 0.0054	0.8005 ± 0.0103	0.7970 ± 0.0101	**0.9033 ± 0.0053**
seeds	0.9368 ± 0.0168	0.9410 ± 0.0238	**0.9508 ± 0.0235**	0.9282 ± 0.0156	0.9498 ± 0.0122	0.9483 ± 0.0229
wine_red	0.6056 ± 0.0152	0.6860 ± 0.0164	**0.7345 ± 0.0138**	0.6236 ± 0.0196	0.6241 ± 0.0162	0.7291 ± 0.0252
glass	0.7227 ± 0.0272	0.8332 ± 0.0574	0.8642 ± 0.0243	0.7219 ± 0.0329	0.7444 ± 0.0465	**0.8829 ± 0.0359**
penbased	0.9493 ± 0.0110	0.9216 ± 0.0021	0.9741 ± 0.0011	0.9748 ± 0.0024	0.9714 ± 0.0004	**0.9964 ± 0.0004**
texture	0.9592 ± 0.0015	0.989 ± 0.0007	0.9924 ± 0.0012	0.9600 ± 0.0022	0.9599 ± 0.0052	**0.9964 ± 0.0005**
wine_white	0.6288 ± 0.0052	0.718 ± 0.0381	0.7254 ± 0.0132	0.6365 ± 0.0076	0.6430 ± 0.0185	**0.7391 ± 0.0312**

strategy surpasses that of the MVDT in accuracy. It indicates that the performance of the method is contributed to further improve after dealing with the imbalanced problem. Besides, the results of the SPCM shows that the SPCM is obviously superior to the

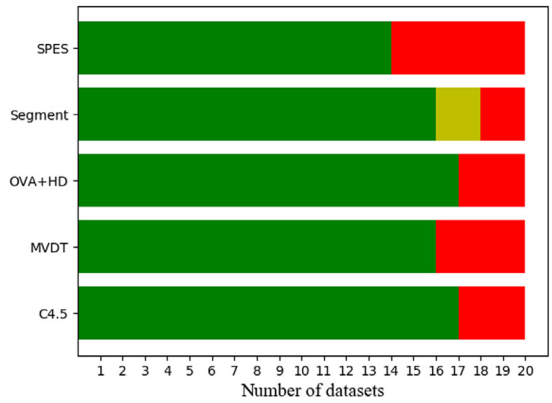


Fig. 4. Comparison of the SPCM with other methods in the number of win (green), tie (yellow) and loss (red) over twenty data sets.

OVA + HD from the accuracy point of view. It proves that the proposed approach can effectively correct the unreasonable splitting point by introducing additional information and enhance the classification performance. Finally, the SPCM method falls behind other methods in accuracy for five data sets, e.g., *wine, ecoli, automobile, tae* and *seeds*. All these data sets have a small number of examples and the maximal size is ecoli that has 327 examples. There are two possible reasons. First, the base classifiers are introduced by the half number of examples from data set, which may be under-fitting in the case of the OVA strategy. Second, Table 4 shows that the SPCM method wins against all other methods for some small size data sets, such as *glass* and *libras*. This means that distribution and characteristics of training data have a certain influence on classification performance of the introduced decision tree. Fig. 5 demonstrates the standard deviation of $\hat{K}$ candidate splitting points based on Hellinger distance and the sum of segment when splitting the root node for *glass, libras, seeds, tae, ecoli* and *wine*. This suggests that the candidate splitting points have similar Helliger distances but quite different segment values. In this case, choosing the one with minimal segment value not only maintains the discrimination capability but produces a compact decision tree. While the standard deviation is large based on Hellinger distance but small based on the segment for *seeds, tae, ecoli* and *wine*. It indicates that the candidate splitting points have quite different Hellinger distances but similar segment values. Therefore, the SPCM method chooses the splitting point with minimal segment and small Hellinger distance, which constructs a decision tree with poor generalization performance.

Fig. 4 shows comparison between the SPCM method and all other methods in the form of win/tie/loss graphs, where each row stands for a win/tie/loss. It is observed that the SPCM method is significantly prominent in the number of data sets. This indicates that the SPCM method is robust for different types of data sets.

Table 5 presents the scale of the decision tree in the number of nodes and depth. Supposing that d_1 and d_2 are the depths of the decision tree built by the SPCM and C4.5 methods, respectively; then the relative increase ratio of C4.5 compared to that of the SPCM is calculated by $(d_2 - d_1)/d_1$. Similar calculations are extended to other methods and the comparisons of the number of nodes. From the computation point of view, if the measure value is less than zero, the complexity of decision tree built by the C4.5 method is less than of the SPCM. On the contrary, if the measure value is larger than zero, it is inferred that the complexity of C4.5 is more than that of the SPCM. The complexity of decision trees are analyzed in terms of whether to use the OVA scheme or not. Fig. 6 shows the relative increase ratio of the decision tree scale, which takes the SPCM as the baseline

Table 8
The time comparisons of constructing decision tree for different methods.

DataSet	C4.5	MVDT	OVA + HD	Segment	SPES	SPCM
vowel	18.36	3.08	**2.61**	20.72	17.75	2.66
libras	15.68	**3.13**	3.17	20.17	17.47	3.14
yeast	4.95	0.91	**0.88**	6.26	6.27	0.90
wine	1.21	0.65	0.65	4.32	1.85	**0.64**
thyroid	1.11	0.61	**0.59**	4.29	1.81	0.69
iris	1.02	0.62	0.58	0.72	1.70	**0.37**
vehicle	5.40	2.25	**2.06**	6.60	5.75	2.65
ecoli	1.30	0.42	**0.39**	4.31	1.98	0.48
auto	1.42	0.45	**0.39**	4.41	1.99	0.42
segment	31.59	27.71	26.71	30.36	31.00	**25.16**
tea	1.10	0.61	**0.60**	4.30	1.96	0.63
cleveland	1.56	0.49	**0.41**	4.17	2.24	0.64
image	129.63	33.21	27.61	74.39	111.00	**23.88**
waveform	489.36	253.29	245.12	296.21	499.83	**236.16**
seeds	1.16	0.66	**0.63**	4.36	1.89	0.70
wine_red	28.28	12.89	6.79	16.79	28.05	**6.53**
glass	0.61	0.44	**0.39**	0.53	0.56	0.47
penbased	27.83	29.39	23.29	22.75	28.20	**21.87**
texture	639.28	120.12	98.72	585.21	425.20	**90.71**
wine_white	327.99	164.62	83.07	322.84	195.88	**69.08**

method and compares it with all the other methods. It is observed that the decision trees built by the methods combining with the OVA mechanism are smaller than those without combining the OVA scheme, regardless of the number of nodes or depth. Since it is easier to build a classifier to distinguish between two classes, the decision tree is simpler in binary classes. While the methods combining the OVA strategy, the decision tree generated by the OVA+C4.5 is more complicated than that of the OVA + HD and SPCM methods. This means that the Hellinger distance can produce a more compact decision tree than the C4.5 method in imbalanced scenarios. Moreover, although the OVA + HD method has less number of nodes than that of the SPCM method in most data sets, its depth shows the opposite trend. Besides, one can make a further analysis and find that the depth of decision tree built by the SPCM method is smaller than that of the OVA + HD for the relatively large data sets, except for *segment*. This interest discovery means that, with increasing the size of training set, the SPCM may build a more compact decision tree compared with the OVA + HD scheme. Because the SPCM method selects the splitting point by the *split ratio* that can reduces the depth of the decision tree. Fig. 7 shows a simple illustration. It is observed that although these two decision trees have the same number of nodes, the right one has the lower depth.

Tables 6 and 7 show the comparison of different methods in F1 score and MAUC value for all data sets, respectively. From Table 6, one can observe that, compared with all the other methods, the SPCM method has the highest F1 score in most of data sets. In addition, Table 7 shows that the AUC value of the SPCM is superior to that of other methods in fifteen out of twenty data sets. Especially, it is higher than the OVA + HD scheme in AUC for sixteen data sets. Furthermore, one can make further analysis and find an interest phenomenon that when the data sets are relatively large, the F1 score and AUC values of the SPCM are significantly higher than that of other methods. Therefore, it is concluded that the proposed method can effectively improve the successful predictions of examples from different categories of data sets.

Table 8 shows the training time of constructing decision tree for all the methods. From Table 8, one can observe that the methods, which have fewer training time, are the ones combining with OVA scheme. Among them, the OVA + HD is the best one in ten out of twenty data sets. Then, the SPCM lies on the second place, which has the fewest training time in nine data sets. However, one can further analysis and find an that the SPCM has fewer

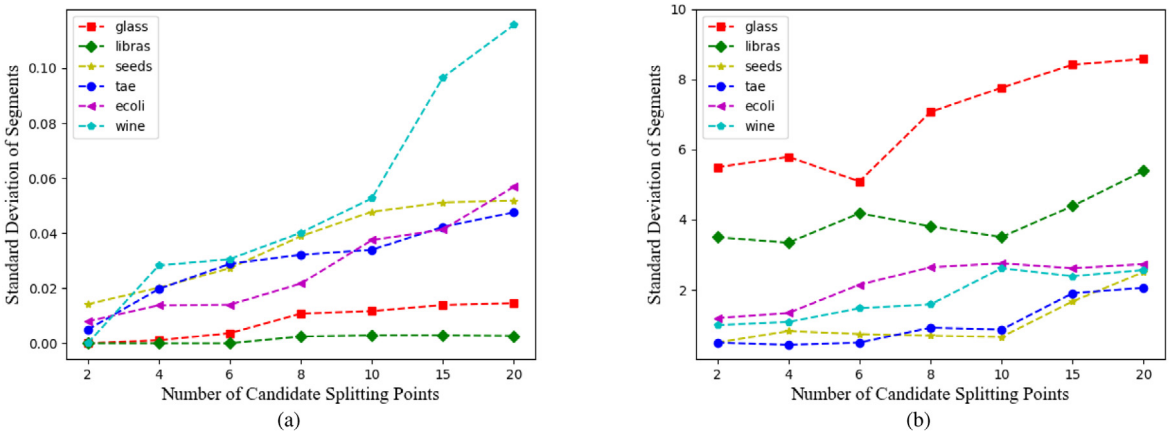


Fig. 5. Standard deviation based on Hellinger distance and the sum of segment of the $\hat{K}$ candidate splitting points when splitting the root node. (a) standard deviation based on Hellinger distance. (b) standard deviation based on the sum of segment.

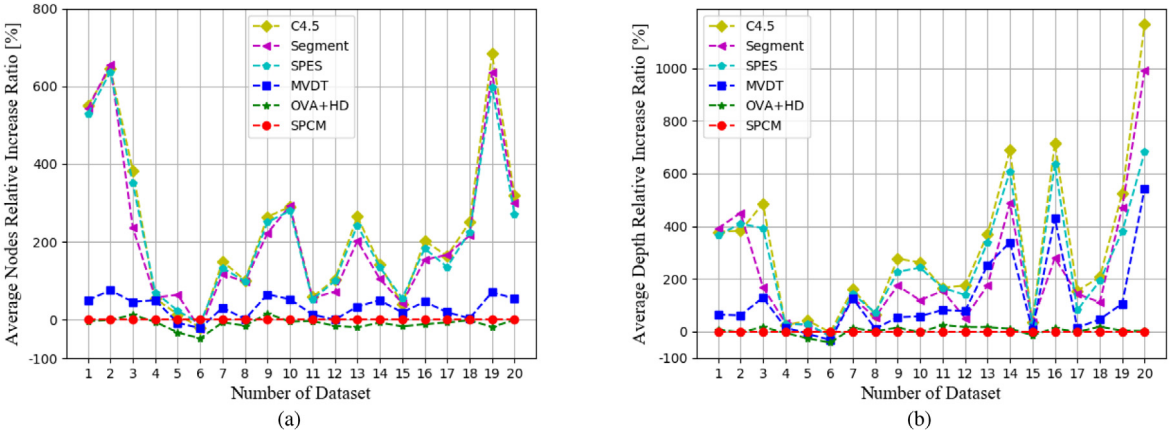


Fig. 6. Comparison of the average number of nodes and depth for different schemes and data sets. (a) comparison of the average number of nodes. (b) comparison of the average depth.

training time than OVA + HD for the relatively large data sets, such as *waveform* and *texture*. This interest discovery in turn has been proved by a fact that the depth of decision tree constructed by the SPCM is smaller than that of OVA + HD in these data sets. Therefore, it can draw a conclusion that, with increasing of training set size, the SPCM can effectively reduce time in training phase compared with other methods.

Fig. 8 shows the variations of different parameters in the accuracy, number of nodes and depth for all data sets. It is observed that as the $\hat{K}$ value increases, the average accuracy decreases. On the other hand, the average number of nodes and depth show an opposite trend, where the former presents an upward trend in Fig. 8(b), while the latter presents a downward trend in Fig. 8(c). Therefore, although the selected hyper-parameters of the SPCM are adaptive for different data sets, the average accuracy of the data sets show that when the parameter is small, the SPCM scheme yields better accuracy and smaller decision tree.

5.2. Experiment 2: Comparison with the OVO decomposition strategy

The OVO system has been commonly considered to be better classification than the OVA system [25]. The reason of this assertion is mainly attributed to the fact that the latter often produces the class imbalanced problems [55]. However, its classification performance can be comparable with that of the OVO system if the class imbalanced problem is solved. In order to verify this issue, a thorough experiment has been carried out between OVO

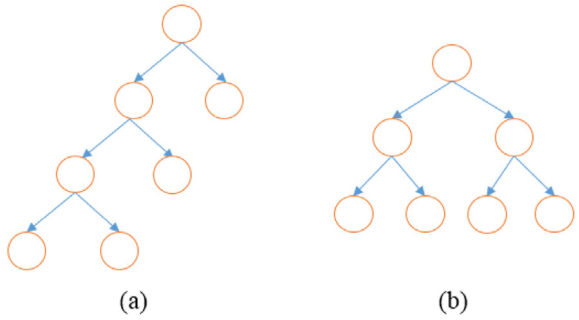


Fig. 7. Two decision trees with different depths. (a) a larger depth decision tree. (b) a compact decision tree.

scheme and OVA scheme on twenty multi-class data sets under the same conditions. Table 9 shows the experimental results of the OVO+C4.5 and the SPCM in accuracy for all the data sets. From Table 9, one can observe that the classification performance of the SPCM is higher than that of the OVO+C4.5 for sixteen out of twenty data sets. In order to illustrate more details of the SPCM in improving the classification performance, Fig. 9 shows comparison of the relative accuracy increase ratio of the OVO+C4.5 and the SPCM based on the MVDT. From Fig. 9, it is observed that the classification performance of the OVO+C4.5 is better than that of the MVDT in most data sets. However, the performance

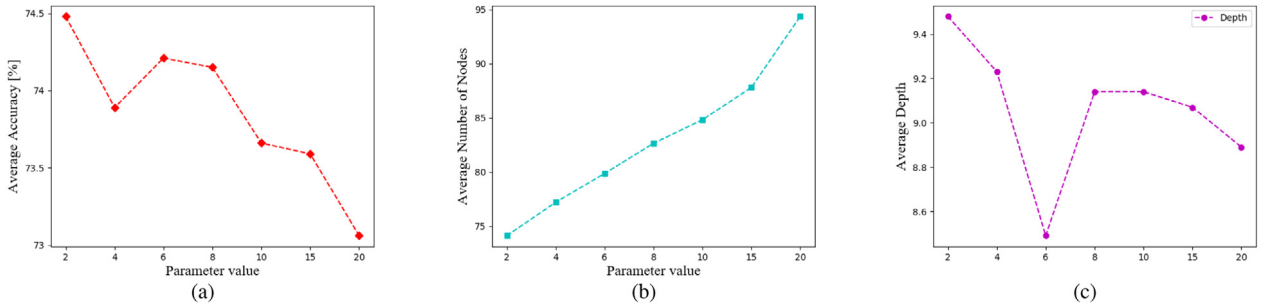


Fig. 8. Variations of different parameters in the average accuracy, number of nodes and the depth for twenty data sets.

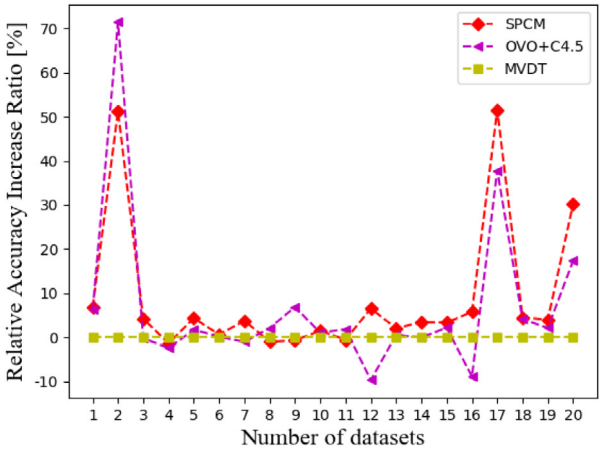


Fig. 9. Comparison of the relative accuracy ratio of the OVO+C4.5 and SPCM based on the MVDT .

Table 9
The average accuracy of the OVO + C4.5 and the SPCM schemes.

DataSet	OVO + C4.5	SPCM
vowel	0.7232 ± 0.0273	**0.7267 ± 0.0329**
libras	**0.6133 ± 0.0352**	0.5411 ± 0.0473
yeast	0.5411 ± 0.0197	**0.5647 ± 0.0221**
wine	0.8966 ± 0.0494	**0.9067 ± 0.0402**
thyroid	0.9196 ± 0.0290	**0.9430 ± 0.0184**
iris	0.9440 ± 0.0196	**0.9487 ± 0.0207**
vehicle	0.6846 ± 0.0178	**0.7175 ± 0.0224**
ecoli	**0.7969 ± 0.0301**	0.7729 ± 0.0335
auto	**0.6744 ± 0.0730**	0.6266 ± 0.0836
segment	0.9526 ± 0.0089	**0.9571 ± 0.0073**
tae	**0.5133 ± 0.0467**	0.5013 ± 0.0431
cleveland	0.4547 ± 0.0298	**0.5365 ± 0.0350**
image	0.8535 ± 0.0041	**0.8661 ± 0.0040**
waveform	0.7365 ± 0.0091	**0.7618 ± 0.0081**
seeds	0.9010 ± 0.0253	**0.9114 ± 0.0369**
wine_red	0.5505 ± 0.0154	**0.6390 ± 0.0106**
glass	0.6216 ± 0.0253	**0.6833 ± 0.0423**
penbased	0.9582 ± 0.0038	**0.9622 ± 0.0052**
texture	0.9195 ± 0.0044	**0.9364 ± 0.0075**
wine_white	0.5462 ± 0.0112	**0.6063 ± 0.0082**

of the SPCM surpasses the OVO+C4.5 scheme after dealing with the class imbalanced problem. This conclusion suggests that the classification accuracy of OVA system can be superior to the OVO system in the case of addressing the class imbalanced problem.

5.3. Experiment 3: comparison with the segment in the case of the OVA scheme

Similar to the SPCM, Segment [46] chooses the optimal splitting point by considering class distribution information and permutation information from training data. To verify the effectiveness of the proposed method that addressed the class imbalanced problem and the concept of split ratio, a comparative experiment between the SPCM method and the Segment method has been conducted on twenty data sets combining with the OVA strategy, respectively. The Multi-Segment and the SPCM schemes are evaluated from the scale of decision tree and the accuracy point of view.

Tables 10 and 11 show the comparison between the Multi-Segment and the SPCM in accuracy and tree scale for all the data sets. From Table 10, it is observed that the classification performance of the SPCM is superior to that of the Multi-Segment in fifteen out of twenty data sets. On the other hand, there is a clear difference between these two methods in the scale of decision tree. Table 11 shows that the scale of decision tree

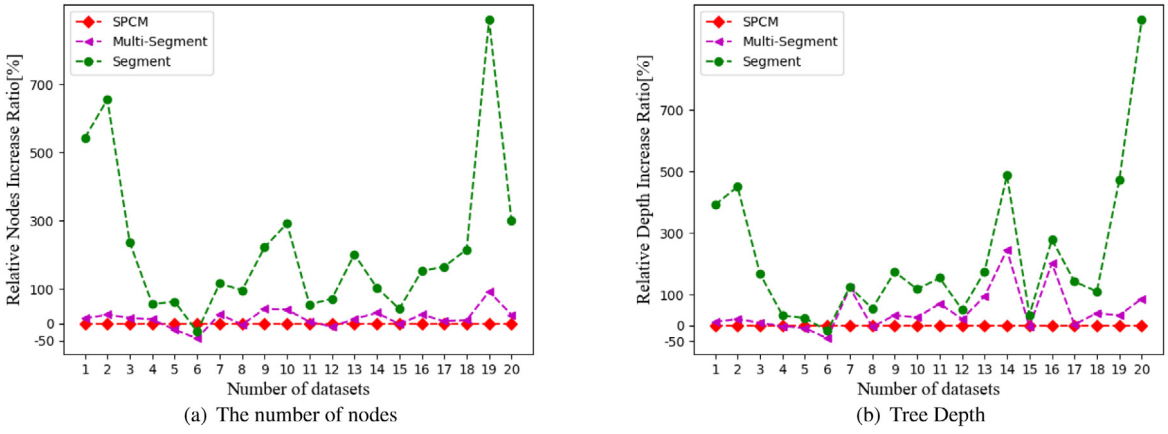


Fig. 10. Relative scale of decision tree increase ratio between SPCM and Multi-Segment.

Table 10
The comparison of the multi-segment and the SPCM in accuracy for all the data sets.

DataSet	Multi-Segment.	SPCM
vowel	0.7198 ± 0.0331	**0.7267 ± 0.0329**
libras	0.4778 ± 0.0328	**0.5411 ± 0.0473**
yeast	**0.5673 ± 0.0119**	0.5647 ± 0.0221
wine	0.9011 ± 0.0507	**0.9067 ± 0.0402**
thyroid	0.9374 ± 0.0304	**0.9430 ± 0.0184**
iris	**0.9547 ± 0.0208**	0.9487 ± 0.0207
vehicle	0.7083 ± 0.0133	**0.7175 ± 0.0224**
ecoli	**0.7982 ± 0.0376**	0.7729 ± 0.0335
auto	**0.6705 ± 0.0487**	0.6266 ± 0.0836
segment	0.9494 ± 0.0082	**0.9571 ± 0.0073**
tae	0.4947 ± 0.0570	**0.5013 ± 0.0431**
cleveland	**0.5426 ± 0.0256**	0.5365 ± 0.0350
image	0.8616 ± 0.0060	**0.8661 ± 0.0040**
waveform	0.7500 ± 0.0075	**0.7618 ± 0.0081**
seeds	0.8914 ± 0.0354	**0.9114 ± 0.0369**
wine_red	0.6180 ± 0.0196	**0.6390 ± 0.0106**
glass	0.4706 ± 0.0431	**0.6833 ± 0.0423**
penbased	0.9278 ± 0.0325	**0.9622 ± 0.0052**
texture	0.9010 ± 0.0211	**0.9364 ± 0.0075**
wine_white	0.5200 ± 0.0069	**0.6063 ± 0.0082**

Table 11
The average scale of decision tree constructed by the multi-segment and SPCM schemes.

DataSet	Multi-Segment		SPCM	
	Nodes	Depth	Nodes	Depth
vowel	33.35	9.84	**29.25**	**8.80**
libras	11.95	5.03	**9.53**	**4.14**
yeast	89.18	13.24	**77.02**	**12.18**
wine	8.53	**3.17**	**7.67**	3.23
thyroid	**10.33**	**4.23**	12.73	4.60
iris	**6.53**	**2.40**	11.67	4.10
vehicle	89.80	28.33	**71.00**	**12.73**
ecoli	**24.16**	**6.18**	25.10	6.46
auto	16.28	5.64	**11.47**	**4.22**
segment	33.57	8.24	**23.97**	**6.51**
tae	42.47	16.37	**40.67**	**9.57**
cleveland	**43.60**	11.78	48.28	**9.70**
image	203.47	28.65	**180.77**	**14.72**
waveform	433.60	59.43	**331.67**	**17.20**
seeds	**11.13**	**4.27**	11.40	4.47
wine_red	203.84	47.48	**162.32**	**15.68**
glass	19.16	6.12	**18.04**	**5.92**
penbased	82.73	12.49	**75.72**	**8.89**
texture	62.19	10.70	**43.22**	**8.03**
wine_white	482.43	44.05	**386.30**	**23.55**

constructed by the SPCM is significantly smaller than that of decision tree constructed by Multi-Segment in most data sets. Fig. 10 shows the relative scale increase ratio of decision tree based on the SPCM method. From Fig. 10(a), it shows that the number of nodes resulting from the SPCM is less than that of the Multi-Segment in most data sets. Fig. 10(b) shows that there is a significant difference in the depth of the decision tree between them depending on the size of the data sets. The larger the data sets, the greater difference between them. Specifically, the largest difference is more than 2.5 times in depth of Multi-Segment compared with the SPCM method for *waveform*.

Based on the above-mentioned analysis, the following conclusions are made: First, the Hellinger distance can effectively deal with the unbalanced problem compared with the information entropy. This is beneficial to improve the classification performance. Second, the Segment method chooses the candidate splitting point with the minimal number of expected segment as the optimal one. However, it may make an improper choice, because it uses the unreasonable initial information in the class imbalanced scenario. While the SPCM method determines the optimal splitting point by the sum of segment from two subsets and split ratio, which are insensitive to imbalanced problem. Therefore, it can be concluded that the proposed method not only maintains the outstanding classification accuracy through addressing the class imbalanced problem, but builds a more compact decision tree.

6. Conclusion and future works

The present study focuses on handling a multi-class classification problem by combining decision trees with the OVA decomposition strategy. Unlike previous literature, which has been mainly concentrated on aggregation strategies for outputs of multiple base classifiers, a great attention is paid in the present study to construct the well-identified base learners. Therefore, the splitting point correcting matrix (SPCM) is proposed to deal with the imbalanced problem caused by the OVA strategy. Full advantage is taken from information about training data at a node. Particularly, the concept of *split ratio* embedded in the SPCM is introduced, which is contributed to select the most appropriate splitting point. The performance of the proposed method is compared with different conventional methods. It is found that the SPCM scheme has remarkable advantageous, including the evaluating of the splitting point from the multi-angle. Therefore, it can achieve the purpose of correction.

An extensive comparison with different multi-class classification approaches is carried out. It is found that the proposed method is able to statistically outperform all of other methods in accuracy, as well as less complexity with respect to the built decision tree. Moreover, both well-known decomposition strategies, OVO and OVA, are analyzed in dealing with the multi-class problem. The results show that the OVA system is in defense of its reputation for good performance. Finally, the splitting points competition problem is studied in the process of building a decision tree. It is concluded that the SPCM not only gains superior accuracy, but also has a distinct advantage in tree depth compared to that of the Multi-Segment method.

Throughout this study, it is concluded that the SPCM is a simple method to deal with the imbalanced problem and it is a useful way to improve the classification accuracy and reduce the model complexity. However, there are many future works remained to be addressed. It is worth studying how to mine more information to help partition the node. It is also planed to investigate the possibility of combination ECOC and multi-information for handling multi-class imbalanced data sets.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jianjian Yan: Conceptualization, Methodology, Writing - review & editing, Writing - original draft, Investigation, Resources, Formal analysis, Validation. **Zhongnan Zhang:** Conceptualization, Methodology, Writing - review & editing, Writing - original draft. **Kunhui Lin:** Data curation, Supervision. **Fan Yang:** Funding acquisition. **Xiongbiao Luo:** Formal analysis, Validation.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant number 61801409).

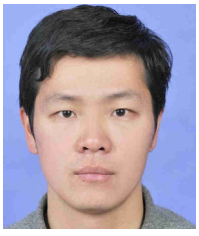
References

[1] X.X. Niu, C.Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits, Pattern Recognit. 45 (4) (2012) 1318–1325.
[2] A. Anand, P.N. Suganthan, Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates, J. Theoret. Biol. 259 (3) (2009) 533–540.
[3] G. Inan, U. Elif Derya, Multiclass support vector machines for EEG-signals classification, IEEE Trans. Inf. Technol. Biomed. 11 (2) (2007) 117–126.
[4] N. Khan, R. Ksantini, I. Ahmadb, B. Boufamab, A novel SVM + NDA model for classification with an application to face recognition, Pattern Recognit. 45 (1) (2012) 66–79.
[5] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, An ensemble of filters and classifiers for microarray data classification, Pattern Recognit. 45 (1) (2012) 531–539.
[6] E.B. Hunt, J. Marin, P.J. Stone, Experiments in induction, Am. J. Psychol. 80 (4) (1966) 17–19.
[7] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[8] J.R. Quinan, C4.5 Programs for Machine Learning, first ed., Morgan Kaufmann Publishers, San Mateo-California, 1993.
[9] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, CA, USA, 1984.
[10] M. Mehta, R. Agrawal, J. Rissanen, Sliq: a fast scalable classifier for data mining, in: International Conference on Extending Database Technology: Advances in Database Technology, 1996, pp. 18–32.
[11] J.C. Shafer, R. Agrawal, M. Mehta, SPRINT: A scalable parallel classifier for data mining, in: VLDB'96 Proceedings of the 22th International Conference on Very Large Data Bases, 1996, pp. 544–555.
[12] B. Chandra, P.P. Varghese, Moving towards efficient decision tree construction, Inform. Sci. 179 (8) (2009) 1059–1069.
[13] C.J. Mantas, J. Abellán, Analysis and extension of decision trees based on imprecise probabilites: Application on noisy data, Expert Syst. Appl. 41 (5) (2014) 2514–2525.
[14] Y. Wang, S.T. Xia, J. Wu, A less-greedy two-term Tsallis entropy information metric approach for decision tree classification, Knowl.-Based Syst. 120 (2017) 34–42.
[15] C.C. Wu, Y.L. Chen, Y.H. Liu, X.Y. Yang, Decision tree induction with a constrained number of leaf nodes, Appl. Intell. 45 (3) (2016) 1–13.
[16] A.C. Lorena, A.C.P.L.F. de Carvalho, J.a.M.P. Gama, A review on the combination of binary classifiers in multiclass problems, Artif. Intell. Rev. 30 (1–4) (2008) 19–37.
[17] Z. Zhang, B. Krawczyk, S. García, A. Rosales-Pérez, F. Herrera, Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data, Knowl.-Based Syst. 106 (2016) 251–263.
[18] P. Clark, R. Boswell, Rule induction with CN2: Some recent improvements, in: EWSL'91: Processing of the European Working Session on Learning, 1991, pp. 151–163.
[19] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error, J. Artificial Intelligence Res. 2 (1) (1995) 263–286.
[20] S. Wang, L. Jiang, C. Li, Adapting naive Bayes tree for text classification, Knowl. Inf. Syst. 44 (1) (2015) 77–89.
[21] X. Guan, J. Liang, Y. Qian, J. Pang, A multi-view OVA model based on decision tree for multi-classification tasks, Knowl.-Based Syst. 138 (2017) 208–219.
[22] M. Galar, A. Ferna'ndez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybern. C 42 (4) (2012) 463–484.
[23] J. Rnkranz, Round robin classification, J. Mach. Learn. Res. 2 (4) (2002) 721–747.
[24] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[25] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (8) (2011) 1761–1776.
[26] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[27] B. Chandra, R. Kothari, P. Paul, A new node splitting measure for decision tree construction, Pattern Recognit. 43 (8) (2010) 2725–2731.
[28] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[29] W.W. Cohen, Fast effective rule induction, in: Twelfth International Conference on International Conference on Machine Learning, 1995, pp. 115–123.
[30] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, in: ICML'00 Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 9–16.
[31] O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, Pattern Recognit. 41 (2) (2008) 713–725.
[32] P. Oriol, R. Petia, V. Jordi, Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1007–1012.
[33] M. Sun, K. Liu, Q. Wu, Q. Hong, B. Wang, H. Zhang, A novel ECOC algorithm for multiclass microarray data classification based on data complexity analysis, Pattern Recognit. 90 (2019) 346–362.
[34] J.H. Hong, J.K. Min, U.K. Cho, S.B. Cho, Fingerprint classification using one-vs-all support vector machines dynamically ordered with Naive Bayes classifiers, Pattern Recognit. 41 (2) (2008) 662–671.
[35] J.R. Quinlan, Improved use of continuous attributes in C4.5, J. Artif. Intell. (1996) 77–90.
[36] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (1) (2002) 321–357.
[37] H. He, B. Yang, E.A. Garcia, S. Li, Adasyn: adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Neural Networks, 2008, pp. 1322–1328.
[38] L. Wei, S. Chawla, D.A. Cieslak, N.V. Chawla, A robust decision tree algorithm for imbalanced data sets, in: SIAM International Conference on Data Mining, 2010, pp. 766–777.
[39] C.E. Shannon, A mathematical theory of communication, Bell Labs Tech. J. 27 (4) (1948) 379–423.
[40] D.A. Cieslak, N.V. Chawla, Learning decision trees for unbalanced data, in: European Conference on Machine Learning & Knowledge Discovery in Databases, 2008, pp. 241–256.
[41] D.A. Cieslak, T.R. Hoens, N.V. Chawla, Hellinger distance decision trees are robust and skew-insensitive, Data Min. Knowl. Discov. 24 (1) (2012) 136–158.
[42] K. Boonchuay, K. Sinapiromsaran, C. Lursinsap, Decision tree induction based on minority entropy for the class imbalance problem, Pattern Anal. Appl. 20 (3) (2017) 769–782.
[43] T. Kailath, The divergence and Bhattacharyya distance measures in signal selection, IEEE Trans. Commun. 15 (1) (1967) 52–60.
[44] Z. Daniels, D. Metaxas, Addressing imbalance in multi-label classification using structured hellinger forests, in: 31st AAAI Conference on Artificial Intelligence, 2017, pp. 1826–1832.
[45] K. Grabczewski, Techniques of decision tree induction, in: Meta-Learning in Decision Tree Induction, in: Studies in Computational Intelligence, vol. 498, Springer, Cham, 2014.
[46] R. Wang, S. Kwong, X.Z. Wang, Q. Jiang, Segment based decision tree induction with continuous valued attributes, IEEE Trans. Cybern. 45 (7) (2015) 1262–1275.
[47] J.J. Yan, Z.N. Zhang, L.W. Xie, Z.T. Zhu, A unified framework for decision tree on continuous attributes, IEEE Access 7 (1) (2019) 11924–11933.
[48] T. Elomaa, J. Rousu, On the Splitting Properties of Common Attribute Evaluation Functions, ReportC-2000-1, Department of Computer Science, University of Helsinki, 2000.
[49] B. Leo, Technical note: Some properties of splitting criteria, Mach. Learn. 24 (1) (1996) 41–47.
[50] T. Elomaa, J. Rousu, On the well-behavedness of important attribute evaluation functions, in: SCAI '97 Proceedings of the Sixth Scandinavian Conference on Artificial Intelligence, 1998, pp. 95–106.
[51] S.L. Salzberg, C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann publishers, Inc., 1993, Mach. Learn. 16 (3) (1994) 235–240.
[52] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: A stepwise procedure for building and training a neural network, in: F.F. Soulié, J. Hérault (Eds.), Neurocomputing, in: NATO ASI Series (Series F: Computer and Systems Sciences), 1990, pp. 41–50.
[53] I. Pillai, G. Fumera, F. Roli, Designing multi-label classifiers that maximize f measures: State of the art, Pattern Recognit. 61 (2017) 394–404.
[54] D.J. Hand, R.J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, Mach. Learn. 45 (2) (2001) 171–186.
[55] K. Bartosz, G. Mikel, W. Michał, B. Humberto, H. Francisco, Dynamic ensemble selection for multi-class classification with one-class classifiers, Pattern Recognit. 83 (2018) 34–51.

Jianjian Yan received his B.S. degree in computer science and technology from Jinggangshan University, Ji'an, China in 2008 and his M.S. degree in Central South University, Changsha, China in 2012. He is currently pursuing the Ph.D. degree in School of Informatics, Xiamen University, Xiamen, China. His current research interests including data mining, machine learning and computer vision and deep learning techniques.

Zhongnan Zhang received the B.E. and M.E. degrees in computer science and technology from Southeast University, Nanjing, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from The University of Texas at Dallas, TX, USA, in 2008. Since 2017, he has been a Full Professor with the Department of Software Engineering, Xiamen University, Xiamen, China, where he was an Assistant Professor from 2009 to 2012 and an Associate Professor from 2012 to 2017. His research interests include big data analysis, data mining, machine learning, and bioinformatics. He is an Editor of the journal PLoS ONE.

Kunhui Lin received the B.S. degree in computer science from Xiamen University, Xiamen, China, in 1983. Since 2008, he has been a Full Professor with the Software School, Xiamen University, Xiamen, China, where he was an Engineer from 1984 to 1997, an Assistant Professor from 1997 to 2000 and an Associate Professor from 2001 to 2008. His research interests include big data analysis, data mining, machine learning, and distributed computing.



Fan Yang received the B.S., M.S. and Ph.D. degree from Xiamen University, Xiamen, China, in 2007, 2010 and 2015 respectively, all in communication engineering. Since October 2015 he has been a faculty member in the School of Information Science and Engineering, Xiamen University, Xiamen, China. His current research interests include vehicular ad hoc networks, wireless communication and networking, computer vision and deep learning techniques.

Dr. Xiongbiao Luo received his PhD degree in Information Science from Nagoya University Japan in 2011. After a half-year postdoctoral fellowship at Nagoya University Japan, he worked as an assistant professor until March 2014. Next, he has been working as a research fellow at the University of Western Ontario Canada until September 2016. After that, he was a senior researcher at the French Institute of Health and Medical Research in France. He has extensive experience in image-guidance methods and external tracking devices for surgical navigation, clinical ultrasound navigation, computer vision and artificial intelligence techniques for clinical procedures, and information processing in computer assisted interventions with publications on these subjects in flagship journals and conferences including IEEE Transactions on Medical Imaging, Medical Image Analysis, IEEE Transactions on Biomedical Engineering, CVPR and MICCAI. Dr. Xiongbiao Luo is also actively involved in his research community. He is a reviewer for the top journals in his research field including IEEE Transactions on Medical Imaging, Medical Image Analysis, IEEE Transactions on Robotics, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, and IEEE Transactions on Biomedical Engineering. More recently he has been an Area Chair of MICCAI 2017 and the Chair of the Local Organizing Committee for MICCAI 2013, the premier conference in this field as well as Program Committee Member for several important conferences, and founded and continuously organized International Workshop on Computer Assisted and Robotic Endoscopy. Currently, he is a full Professor at the Department of Computer Science and Director of the XMU Center for Surgery and Engineering, Xiamen University (XMU) China.