Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Learning label-specific features for decomposition-based multi-class classification

Bin-Bin JIA [1], **Jun-Ying LIU** [1], **Jun-Yi HANG** [2,3], **Min-Ling ZHANG** (✉)[2,3]

1 College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
2 School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
3 Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China

Abstract Multi-class classification can be solved by decomposing it into a set of binary classification problems according to some encoding rules, e.g., one-vs-one, one-vs-rest, error-correcting output codes. Existing works solve these binary classification problems in the original feature space, while it might be suboptimal as different binary classification problems correspond to different positive and negative examples. In this paper, we propose to learn label-specific features for each decomposed binary classification problem to consider the specific characteristics containing in its positive and negative examples. Specifically, to generate the label-specific features, clustering analysis is respectively conducted on the positive and negative examples in each decomposed binary data set to discover their inherent information and then label-specific features for one example are obtained by measuring the similarity between it and all cluster centers. Experiments clearly validate the effectiveness of learning label-specific features for decomposition-based multi-class classification.

Keywords Machine learning, multi-class classification, error-correcting output codes, label-specific features.

1 Introduction

In machine learning and data mining [1, 2], binary classification is one of the most widely studied and commonly used learning tasks, which simply tries to answer the question "YES/NO". However, it is more general to answer the question "Which one is it" from a number of candidate classes which yields the multi-class classification (MCC) task. In fact, there are many real-world applications from various domains which can be formalized as a MCC problem to be solved, e.g., computer vision [3, 4], text mining [5, 6], bioinformatics [7, 8]. Moreover, there are also some more complicated classification tasks which can be reduced into a MCC problem to be solved, e.g., multi-label classification [9, 10], multi-dimensional classification [11, 12].

To deal with the MCC task, existing works can be roughly categorized into direct strategy and indirect strategy. Specifically, the direct strategy aims at designing classification algorithms which can directly learn from multi-class data, e.g., multi-class support vector machine, softmax regression. On the other hand, the indirect strategy aims at decomposing the MCC problem into a set of binary classification problems and then any off-the-shelf binary classification algorithms can be used to solve the MCC problem [13]. Compared with the direct strategy, the indirect strategy is more flexible, and some empirical studies also show its superior performance [14, 15].

There are three popular strategies to decompose the MCC problem into a set of binary classification problems, including one-vs-one (OvO), one-vs-rest (OvR) and error-correcting output codes (ECOC) [16], where both the first two strategies can be regarded as special cases of the last one [17]. Generally, ECOC consists of two steps, namely encoding and decoding, where the former aims at transforming the MCC problem into a set of binary classification problems while the latter aims at obtaining the final prediction according to the predictive outputs of learned binary classifiers.

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *zhangml@seu.edu.cn*

Thus, existing works mainly explore from two corresponding perspectives, i.e., how to consider the specific characteristics of practical problems for better encoding [18, 19] and how to make full use of predictive outputs of learned binary classifiers for better decoding [20, 21].

As for the learning procedure of binary classifiers, the common strategy in existing studies is to train binary classifiers based on the original features of corresponding examples. In other words, identical feature space is used for all decomposed binary classification problems. However, the positive and negative examples in different decomposed binary classification problems are related to different classes according to the decoding rule, which leads to different specific characteristics for different problems. Take the handwritten digits classification [22] as an example, for learning the binary classifier to classify '1' and '6', features related to vertical edges might be more discriminative, while for learning the binary classifier to classify '1' and '7', features related to horizontal edges might be more discriminative. Therefore, if *label-specific features* can be learned to solve each decomposed binary classification problem, it can be expected to obtain more effective binary classifiers and then improve the performance of decomposition-based MCC solutions with the same encoding and decoding strategy.

Here, it is worth noting that the label-specific features refer to the specific features learned for the positive and negative class labels in the decomposed binary classification problem. The class labels do not necessarily correspond to one of the class labels in the original multi-class classification problem. For example, in ECOC decomposition strategy, the positive and negative class labels in each decomposed binary classification problem might correspond to multiple class labels in the original multi-class classification problem. Besides, even in the OvO decomposition strategy, the label-specific features for the same class label in different decomposed binary classification problems are not necessarily the same due to their another different class labels.

In this paper, we investigate the feasibility of learning label-specific features for decomposition-based MCC solutions. Specifically, for each decomposed binary classification problem, we respectively perform clustering analysis on its positive examples and negative examples to discover the inherent characteristics residing in this specific classification problem. Based on the clustering results, label-specific features for each example are constructed by measuring the similarity between the example and all cluster centers. Experiments show the superiority of learning binary classifiers based on the generated label-specific features against the original features.

The rest of the paper is organized as follows. Firstly, related works are briefly discussed in Section 2. Then, the technical details of learning label-specific features for decomposition-based MCC are presented in Section 3. After that, comparative studies are conducted in Section 4. Finally, this paper is concluded in Section 5.

2 Related Work

The most basic classification task is binary classification and many popular classification algorithms are initially designed for this task, e.g., support vector machine [23]. As a generalized case of binary classification, the MCC problem can be solved via either adapting existing binary classification algorithms (i.e., direct strategy) or transforming it into binary classification problems (i.e., indirect strategy). Besides, the recently proposed approach named CODIL aims at bridging the gap between the two strategies to combine their respective strengths [24]. For the direct strategy, some of binary classification algorithms can be directly generalized to solve the MCC problem, e.g., *k*NN classifier and decision trees, while some of them need special adaptions for multi-class data, e.g., support vector machine [25] and adaptive boosting [26]. For the indirect strategy, the key challenges are how to decompose the MCC problem into binary classification problems (i.e., encoding) and how to combine the predictive outputs of these binary classifiers to obtain the final prediction (i.e., decoding).

For the encoding phase, the basic OvR and OvO result in fixed binary decomposition for one MCC problem while the vanilla ECOC simply generates random coding matrix and then results in random binary decomposition [16]. In other words, they cannot consider the specific characteristics of practical applications and then might lead to suboptimal solutions. The DECOC method [18] hierarchically generates each column of ECOC coding matrix by maximizing the quadratic mutual information among different classes in a top-to-down manner, and SECOC [19] further partitions some complicated classes into several subclasses based on DECOC. Different from DECOC and SECOC, M2ECOC [27] and SM2ECOC [28] generate the ECOC coding matrix in a bottom-to-up manner via maximum margin criterion. Besides, given an ECOC coding matrix, the ECOC-ONE method [21] will iteratively generate new columns to the coding matrix based on well-established discriminability criterion which is related to the specific MCC application.

For the decoding phase, OvO decoding usually corresponds to majority voting based on binary-valued predictions of binary classifiers while OvR decoding usually corresponds to maximizing real-valued decision values of binary classifiers. For the general ECOC, the decoding phase corresponds to measuring the distance between the predictive outputs from binary classifiers and the corresponding ground-truth class in each decomposed binary classification problem [16]. Commonly used distance metrics include Hamming distance and Euclidean distance, which only utilize the binary-valued predictions of binary classifiers. The loss-based decoding strategy [17] introduces some loss function (e.g., exponential loss) for distance measuring by utilizing the predictive confidences of binary classifiers. To pay more attentions on the predictive outputs from binary classifiers with better performance, the weighted loss-based decoding strategy [20] further introduces the empirical performance to weight the corresponding loss of each binary classifier.

There are also several works which focus on the binary classifier learning procedure for the decomposed binary classification problems. As different classification problems are inclined to different classification algorithms due to their specific characteristics, the DOAO method [29] solves the decomposed binary classification problems with different binary classification algorithms. Besides, there might be some inherent relationships among the decomposed binary classification problems as they are obtained from the same MCC problem, the JCL method [30] trains all decomposed binary classifier in a joint manner by introducing covariance regularization. However, to the best of our knowledge, there have not been existing works which focus on learning label-specific features for each decomposed binary classification problem.

3 Methodology

In this section, we firstly give the technical details of binary decomposition strategies, including the decomposition procedure and the corresponding prediction procedure. After this, we present the technical details of learning label-specific features for each decomposed binary classification problem.

Formally speaking, for a multi-class classification problem with N classes, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional input space, and $\mathcal{Y} = \{c_1, c_2, \ldots, c_N\}$ be the output space with N possible classes. Given a MCC training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \mid 1 \leq i \leq m\}$ with m examples, where $\boldsymbol{x}_i \in \mathcal{X}$ denotes the d-dimensional instance vector and $y_i \in \mathcal{Y}$ denotes the corresponding class associated with $\boldsymbol{x}_i$. The task of multi-class classification aims at learning a mapping function $g : \mathcal{X} \mapsto \mathcal{Y}$ which can assign a proper class for unseen instance.

3.1 Binary Decomposition

For convenience, let $C_j = \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, y_i) \in \mathcal{D}, y_i = c_j\}$ be the set of instances associated with the j-th class in $\mathcal{D}$ $(1 \leq j \leq N)$, it is easy to know that $\sum_{j=1}^{N} |C_j| = m$, where $|\cdot|$ computes the cardinality of one set.

3.1.1 The One-vs-One Strategy

In OvO strategy, each binary classification data set is constructed by putting one class as positive and one class as negative. It is easy to know that there are a total of $L_{\text{ovo}} = \frac{N(N-1)}{2}$ decomposed binary classification tasks for a MCC task with N classes.

Specifically, let $\mathcal{D}_{\text{ovo}}^l = \{(\boldsymbol{x}_i^l, y_i^l) \mid 1 \leq i \leq n^l\}$ be the l-th decomposed binary classification data set $(1 \leq l \leq L_{\text{ovo}})$, suppose $\mathcal{D}_{\text{ovo}}^l$ takes class c_u as positive and class c_v as negative $(1 \leq u < v \leq N)$, i.e., $y_i^l = +1$ if $\boldsymbol{x}_i^l \in C_u$ while $y_i^l = -1$ if $\boldsymbol{x}_i^l \in C_v$, we have $n^l = |C_u| + |C_v|$. Based on each data set $\mathcal{D}_{\text{ovo}}^l$, we can learn one binary classifier f_{ovo}^l with some binary classification algorithm $\mathfrak{L}$ (e.g., logistic regression). For unseen instance $\boldsymbol{x}_*$, each classifier f_{ovo}^l can return one binary-valued predicted result (i.e., $+1$ or -1). If the result is $+1$ (or -1), then one vote will be recorded for the corresponding positive class c_u (or negative class c_v). Let s_j denote the number of recorded votes for the j-th class, then the final prediction $\hat{y}_*$ for $\boldsymbol{x}_*$ is determined by:

$$\hat{y}_* = c_{\hat{j}}, \text{ where } \hat{j} = \underset{1 \leq j \leq N}{\arg\max} \ s_j \tag{1}$$

3.1.2 The One-vs-Rest Strategy

In OvR strategy, each binary classification data set is constructed by putting one class as positive and the rest classes as negative. It is easy to know that there are a total of $L_{\text{ovr}} = N$ decomposed binary classification tasks for a MCC task with N classes.

Specifically, let $\mathcal{D}_{\text{ovr}}^l = \{(\boldsymbol{x}_i^l, y_i^l) \mid 1 \leq i \leq n^l\}$ be the l-th decomposed binary classification data set $(1 \leq l \leq L_{\text{ovr}})$, where $\mathcal{D}_{\text{ovr}}^l$ takes the l-th class c_l as positive and the rest classes as negative, i.e., $y_i^l = +1$ if $\boldsymbol{x}_i^l \in C_l$ while $y_i^l = -1$ otherwise, we always have $n^l = m$. Based on each data set $\mathcal{D}_{\text{ovr}}^l$, we can learn one binary classifier f_{ovr}^l with some binary classification algorithm $\mathfrak{L}$ (e.g., logistic regression). For unseen instance $\boldsymbol{x}_*$, each classifier f_{ovr}^l can return one real-valued predicted

Algorithm 1 Learning label-specific features for decomposition-based MCC.

```
Input: The MCC training set $\mathcal{D}$, the employed binary classifier $\mathfrak{L}$, and the unseen instance $\boldsymbol{x}_*$
Output: The predicted class $\hat{y}_*$ for $\boldsymbol{x}_*$
Decompose $\mathcal{D}$ into a set of binary data sets $\mathcal{D}^l = \{(\boldsymbol{x}_i^l, y_i^l) \mid 1 \le i \le n^l\}$ $(1 \le l \le L)$ via OvO, OvR or ECOC;
for $l = 1$ to $L$ do
    Divide $\mathcal{D}^l$ into two subsets $\mathcal{D}_+^l$ and $\mathcal{D}_-^l$ according to Eq.(4);
    Determine the number of clusters $m^l$ for $\mathcal{D}_+^l$ and $\mathcal{D}_-^l$ according to Eq.(5);
    Group $\mathcal{D}_+^l$ and $\mathcal{D}_-^l$ into $m^l$ disjoint clusters;
    Initialize $\tilde{\mathcal{D}}^l = \varnothing$;
    for $i = 1$ to $n^l$ do
        For the example $\boldsymbol{x}_i^l \in \mathcal{D}^l$, compute its label-specific features $\phi^l(\boldsymbol{x}_i^l)$ according to Eq.(6);
        Add $\left(\phi^l(\boldsymbol{x}_i^l), y_i^l\right)$ into $\tilde{\mathcal{D}}^l$, i.e., $\tilde{\mathcal{D}}^l = \tilde{\mathcal{D}}^l \cup \left(\phi^l(\boldsymbol{x}_i^l), y_i^l\right)$;
    end for
    Train binary classifier $g^l$ over $\tilde{\mathcal{D}}^l$, i.e., $g^l \leftarrow \mathfrak{L}(\tilde{\mathcal{D}}^l)$;
end for
for $l = 1$ to $L$ do
    For unseen instance $\boldsymbol{x}_*$, compute its label-specific features $\phi^l(\boldsymbol{x}_*)$ according to Eq.(6);
    Obtain the prediction of $\boldsymbol{x}_*$ based on its label-specific features $\phi^l(\boldsymbol{x}_*)$ and binary classifier $g^l$;
end for
Return $y_*$ via Eq.(1), Eq.(2) or Eq.(3) according to the corresponding decomposition strategy.
```

result denoted as $f_{\mathrm{ovr}}^l(\boldsymbol{x}_*)$, then the final prediction $\hat{y}_*$ for $\boldsymbol{x}_*$ is determined by:

$$\hat{y}_* = c_{\hat{j}}, \text{ where } \hat{j} = \mathop{\arg\max}_{1 \le j \le N} \ f_{\mathrm{ovr}}^j(\boldsymbol{x}_*) \tag{2}$$

3.1.3 The Error Correcting Output Codes Strategy

In ECOC strategy [16], each binary classification data set is constructed by putting several classes as positive and several classes as negative. Thus, compared to OvO and OvR, ECOC is also known as the most widely used implementation of Many-vs-Many (MvM) technique. Generally, we need to preset the parameter L_{ecoc} to determine the number of decomposed binary classification tasks in ECOC.

Specifically, ECOC generates a coding matrix $\mathbf{M} \in \{+1, -1\}^{N \times L_{\mathrm{ecoc}}}$ or $\{+1, 0, -1\}^{N \times L_{\mathrm{ecoc}}}$ to determine which classes are taken as positive and which classes are taken as negative. let $\mathcal{D}_{\mathrm{ecoc}}^l = \{(\boldsymbol{x}_i^l, y_i^l) \mid 1 \le i \le n^l\}$ be the l-th decomposed binary classification data set which is constructed according to the l-th column $\mathbf{M}$, if the (j, l)-th item $\mathbf{M}(j, l)$ of $\mathbf{M}$ is equal to $+1$ (or -1), then the j-th class will be taken as positive (or negative) in $\mathcal{D}_{\mathrm{ecoc}}^j$. Moreover, if $\mathbf{M}(j, l) = 0$, then the j-th class will not be used to construct $\mathcal{D}_{\mathrm{ecoc}}^l$. It is easy to know $n^l = \sum_{j=1, \mathbf{M}(j,l) \neq 0}^N |C_j|$. Based on each data set $\mathcal{D}_{\mathrm{ecoc}}^l$, we can learn one binary classifier f_{ecoc}^l with some binary classification algorithm $\mathfrak{L}$ (e.g., logistic regression). For unseen instance $\boldsymbol{x}_*$, each classifier f_{ecoc}^l can return one predicted result, the final prediction $\hat{y}_*$ for $\boldsymbol{x}_*$ is determined by:

$$\hat{y}_* = c_{\hat{j}}, \text{ where } \hat{j} = \mathop{\arg\max}_{1 \le j \le N} \ dist\,(f_{\mathrm{ecoc}}(\boldsymbol{x}_*), \mathbf{M}(j, :)) \tag{3}$$

where $f_{\mathrm{ecoc}}(\boldsymbol{x}_*) = [f_{\mathrm{ecoc}}^1(\boldsymbol{x}_*), \ldots, f_{\mathrm{ecoc}}^{L_{\mathrm{ecoc}}}(\boldsymbol{x}_*)]$ and $\mathbf{M}(j, :)$ denotes the j-th row of $\mathbf{M}$. Take the Hamming decoding as an example, $f_{\mathrm{ecoc}}^l(\boldsymbol{x}_*)$ corresponds to binary-valued prediction (i.e., $+1$ or -1) and $dist\,(\cdot, \cdot)$ computes the Hamming distance between its two binary-valued input vectors.

3.2 Label-Specific Features Construction

For simplicity, suppose the MCC problem is transformed into L binary classification problems via some decomposition strategy (e.g., OvO, OvR, ECOC), let $\mathcal{D}^l = \{(\boldsymbol{x}_i^l, y_i^l) \mid 1 \le i \le n^l\}$ be the l-th decomposed binary data set, we aim at constructing some label-specific features which can provide more discriminative information for model induction. Following the idea of learning label-specific features for multi-label classification [31], we choose to characterize specific information w.r.t. each decomposed binary data set via clustering techniques. Specifically, $\mathcal{D}^l$ is firstly divided into two subsets:

$$\begin{aligned} \mathcal{D}_+^l &= \{\boldsymbol{x}_i^l \mid (\boldsymbol{x}_i^l, y_i^l) \in \mathcal{D}^l, y_i^l = +1\} \\ \mathcal{D}_-^l &= \{\boldsymbol{x}_i^l \mid (\boldsymbol{x}_i^l, y_i^l) \in \mathcal{D}^l, y_i^l = -1\} \end{aligned} \tag{4}$$

In other words, $\mathcal{D}_+^l$ and $\mathcal{D}_-^l$ consist of the training instances which are associated with the positive class and negative class in $\mathcal{D}^l$, respectively. To capture the inherent property w.r.t. two parts of training instances, we group $\mathcal{D}_+^l$ into m_+^l disjoint clusters, where the corresponding m_+^l centers are denoted as $\{\boldsymbol{p}_1^l, \boldsymbol{p}_2^l, \ldots, \boldsymbol{p}_{m_+^l}^l\}$. Likewise, $\mathcal{D}_-^l$ is also grouped into m_-^l disjoint clusters, where the corresponding m_-^l centers are denoted as $\{\boldsymbol{n}_1^l, \boldsymbol{n}_2^l, \ldots, \boldsymbol{n}_{m_-^l}^l\}$. In this paper, the clustering procedure is simply done by the traditional k-means algorithm [32]. Moreover, to treat the contribution of $\mathcal{D}_+^l$ and $\mathcal{D}_-^l$ equally for the subsequent model induction, the number of clusters for $\mathcal{D}_+^l$ and $\mathcal{D}_-^l$ is set to be the same, i.e., $m_-^l = m_-^l = m^l$. Specifically, m^l is set as follows:

$$m^l = \lceil r \cdot \min(|\mathcal{D}_+^l|, |\mathcal{D}_-^l|)\rceil \tag{5}$$

where $r \in (0, 1)$ is a hyper-parameter to be tuned.

Generally speaking, the clusters identified by clustering algorithm reveal the underlying structure of the instance set. Thus, the obtained cluster centers can be utilized to construct label-specific features for each decomposed binary classification problem. Here, we define a mapping function $\phi^l : \mathcal{X} \rightarrow \mathcal{Z}^l$ where $\mathcal{Z}^l$ denotes the $2m^l$-dimensional label-specific feature space:

$$\phi^l(\boldsymbol{x}) = \left[\kappa(\boldsymbol{x}, \boldsymbol{p}_1^l), \ldots, \kappa(\boldsymbol{x}, \boldsymbol{p}_{m^j}^l), \kappa(\boldsymbol{x}, \boldsymbol{n}_1^l), \ldots, \kappa(\boldsymbol{x}, \boldsymbol{n}_{m^j}^l)\right]^\top \tag{6}$$

Here, $\kappa(\cdot, \cdot)$ is some similarity measure between two vectors. In this paper, we simply use the Euclidean distance:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \tag{7}$$

With the identified cluster centers and the defined mapping function $\phi^l(\cdot)$, $\mathcal{D}^l$ can be converted into a new binary classification data set:

$$\tilde{\mathcal{D}}^l = \{(\phi^j(\boldsymbol{x}_i^l), y_i^l) \mid 1 \leq i \leq n^l\} \tag{8}$$

Based on $\tilde{\mathcal{D}}^l$, we can use some binary classification algorithm $\mathfrak{L}$ (e.g., logistic regression) to learn a classification model g^l, i.e., $g^l \leftarrow \mathfrak{L}(\tilde{\mathcal{D}}^l)$. With the predictive outputs of all the L binary classifiers $g^1, \ldots, g^L$, we can further obtain the final prediction for unseen instance according to the corresponding decoding rule of the decomposition strategy.

Algorithm 1 summarizes the complete procedure of learning label-specific features for MCC. Specifically, step 1 corresponds to the binary decomposition procedure via one of decomposition strategies. For the l-th binary classification problem, steps 3-10 correspond to specific-features learning procedure and step 11 corresponds to the model induction procedure. For unseen instance $\boldsymbol{x}_*$, steps 13-16 obtain its predictions in all decomposed binary classification problems, and step 17 returns its final multi-class prediction.

4 Experiments

In this section, comparative studies are conducted to evaluate the effectiveness of learning label-specific features for the decomposed binary classification problems in decomposition-based MCC solutions. Firstly, Subsection 4.1 introduces the experimental setup, including benchmark data sets, evaluation metrics and compare schema. Secondly, Subsection 4.2 reports the experimental results with some discussions. Finally, Subsection 4.3 further analyzes the sensitivity and stability for the method of learning label-specific features.

4.1 Experimental Setup

4.1.1 Benchmark Data Sets

In this paper, a total of ten publicly available multi-class data sets are collected for comparative studies, where their detailed characteristics, including the number of examples (#Example), the number of class labels (#Label) and the number of features (#Feature), are summarized in Table 1.

Table 1 Detailed characteristics of the employed MCC data sets.

Data Set	#Example	#Label	#Feature
iris	150	3	4
wine	178	3	13
glass	214	6	9
vowel	528	11	10
dna	2000	3	180
satimage	4435	6	36
usps	7291	10	256
pendigits	7494	10	16
letter	15000	26	16
shuttle	43500	7	9

4.1.2 Evaluation Metrics

In this paper, the two popular multi-class evaluation metrics *Accuracy* and *Average*-F1 are used to evaluate the generalization performance of different multi-class classifiers. Specifically, given a set of MCC test examples $\mathcal{S} = \{(\boldsymbol{x}_i, y_i) \mid 1 \leq i \leq p\}$ where $y_i \in \{c_1, \cdots, c_N\}$, let f be the multi-class classifier to be evaluated and $\hat{y}_i = f(\boldsymbol{x}_i)$ be the predicted class for $\boldsymbol{x}_i$, then the two metrics can be defined as follows:

- *Accuracy*:

$$Acc_{\mathcal{S}}(f) = \frac{1}{p}\sum_{i=1}^{p}[\![\hat{y}_i, y_i]\!] \tag{9}$$

Table 2 Detailed experimental results (mean±std) where the employed binary classifier is SVM.

(a) *Accuracy*

	OvO		OvR		ECOC		
Data Set	Specific	Original	Specific	Original	Specific	Original	MSVM
iris	**0.947±0.053**	0.820±0.077	**0.940±0.049**	0.813±0.093	**0.960±0.056**	0.767±0.047	0.947±0.053
wine	**0.983±0.027**	0.966±0.029	**0.989±0.024**	0.983±0.027	**0.977±0.029**	0.949±0.032	0.966±0.039
glass	**0.640±0.093**	0.620±0.087	**0.677±0.120**	0.472±0.128	**0.706±0.122**	0.579±0.081	0.594±0.096
vowel	**0.915±0.029**	0.750±0.048	**0.723±0.062**	0.398±0.083	**0.852±0.059**	0.441±0.060	0.703±0.071
dna	**0.902±0.060**	0.863±0.036	**0.883±0.045**	0.879±0.032	**0.855±0.124**	0.653±0.041	0.916±0.023
satimage	0.851±0.064	**0.864±0.018**	**0.884±0.028**	0.778±0.025	**0.894±0.009**	0.795±0.035	0.858±0.021
usps	**0.960±0.006**	0.958±0.010	**0.942±0.018**	0.788±0.016	**0.961±0.005**	0.910±0.007	0.944±0.011
pendigits	**0.993±0.002**	0.983±0.004	**0.983±0.020**	0.797±0.013	**0.993±0.003**	0.864±0.017	0.965±0.008
letter	**0.908±0.009**	0.829±0.014	**0.843±0.027**	0.527±0.010	**0.838±0.015**	0.475±0.022	0.781±0.016
shuttle	**0.993±0.013**	0.882±0.007	**0.997±0.001**	0.789±0.037	**0.997±0.001**	0.764±0.049	0.980±0.001

(b) *Average*-F1

	OvO		OvR		ECOC		
Data Set	Specific	Original	Specific	Original	Specific	Original	MSVM
iris	**0.942±0.058**	0.807±0.068	**0.935±0.054**	0.811±0.076	**0.956±0.064**	0.738±0.066	0.941±0.058
wine	**0.979±0.035**	0.960±0.036	**0.985±0.032**	0.983±0.027	**0.975±0.034**	0.947±0.032	0.960±0.045
glass	0.550±0.121	**0.577±0.117**	**0.563±0.149**	0.473±0.095	**0.628±0.123**	0.528±0.082	0.557±0.114
vowel	**0.916±0.028**	0.727±0.054	**0.707±0.064**	0.356±0.075	**0.853±0.054**	0.399±0.051	0.664±0.077
dna	**0.884±0.075**	0.856±0.037	0.862±0.065	**0.873±0.034**	**0.842±0.125**	0.652±0.041	0.906±0.025
satimage	**0.824±0.070**	0.823±0.023	**0.851±0.033**	0.649±0.024	**0.861±0.019**	0.665±0.049	0.782±0.020
usps	**0.955±0.006**	0.952±0.012	**0.938±0.020**	0.762±0.018	**0.956±0.005**	0.900±0.008	0.937±0.013
pendigits	**0.993±0.002**	0.983±0.004	**0.982±0.022**	0.788±0.010	**0.993±0.003**	0.860±0.018	0.965±0.008
letter	**0.906±0.009**	0.826±0.014	**0.846±0.027**	0.510±0.009	**0.839±0.014**	0.442±0.025	0.775±0.015
shuttle	**0.799±0.080**	0.478±0.060	**0.704±0.114**	0.305±0.069	**0.716±0.055**	0.280±0.066	0.660±0.067

Here, $[\![\pi_1, \pi_2]\!]$ returns 1 if π_1 is equal to π_2 and 0 otherwise.

- *Average*-F1:

$$AvgF1_S(f) = \frac{1}{N}\sum_{j=1}^{N}\frac{2P_j \cdot R_j}{P_j + R_j} \tag{10}$$

Here, P_j and R_j denote the precision and recall for the j-th class whose definitions are given as follows:

$$P_j = \frac{\sum_{i=1}^{p}[\![\hat{y}_i, c_j]\!] \wedge [\![y_i, c_j]\!]}{\sum_{i=1}^{p}[\![\hat{y}_i, c_j]\!]}, R_j = \frac{\sum_{i=1}^{p}[\![\hat{y}_i, c_j]\!] \wedge [\![y_i, c_j]\!]}{\sum_{i=1}^{p}[\![y_i, c_j]\!]}$$

For the two metrics, it is easy to know that the *larger* the values, the *better* the performance. Ten-fold cross validation is conducted over each data set for each compared method, where both the mean value and the standard deviation for each evaluation metric are recorded for comparative studies.

4.1.3 Compare Schema

In this paper, to validate whether learning label-specific features for decomposition-based MCC solution is effective or not, given one MCC problem which is solved via decomposition strategy, we compare the experimental results where the binary classifier for each decomposed binary classification problem is trained based on the learned label-specific features and original features. For simplicity, given one decomposition strategy, we use “Specific” and “Original” to respectively denote that the decomposed binary classification problems are solved based on the learned label-specific and original features in the following parts of this paper.

We investigate the three popular decomposition strategies, including OvR, OvO and ECOC. As implementing decomposition MCC methods necessitates a binary classifier, both support vector machine (SVM) and logistic regression (LR) are investigated in experiments and both of them are implemented by LIBLINEAR [33]. We also include multi-class SVM (MSVM) [34] and softmax regression [35] in experiments. They are the two direct MCC methods which generalize SVM and LR to MCC scenarios and serve as the basic reference for decomposition-based MCC solutions. Besides, the ECOC matrix is generated by the Matlab built-in function `designecoc` with parameter setting ‘*denserandom*’.

Table 3 Detailed experimental results (mean±std) where the employed binary classifier is LR.

(a) *Accuracy*

	OvO		OvR		ECOC		
Data Set	Specific	Original	Specific	Original	Specific	Original	Softmax
iris	0.947±0.053	**0.953±0.055**	**0.933±0.070**	0.893±0.084	**0.907±0.105**	0.707±0.064	0.933±0.070
wine	0.983±0.027	0.983±0.027	0.983±0.027	**0.989±0.023**	0.972±0.040	0.972±0.040	0.983±0.027
glass	**0.659±0.093**	0.607±0.117	**0.696±0.127**	0.575±0.093	**0.668±0.127**	0.552±0.093	0.603±0.108
vowel	**0.849±0.048**	0.797±0.062	**0.776±0.042**	0.532±0.068	**0.803±0.034**	0.356±0.070	0.635±0.061
dna	**0.933±0.018**	0.931±0.014	0.930±0.018	**0.944±0.022**	0.906±0.023	**0.912±0.023**	0.937±0.016
satimage	**0.900±0.013**	0.868±0.020	**0.890±0.013**	0.839±0.019	**0.893±0.015**	0.805±0.025	0.860±0.021
usps	0.965±0.006	0.965±0.007	0.948±0.007	**0.951±0.010**	**0.960±0.007**	0.914±0.012	0.953±0.008
pendigits	**0.990±0.004**	0.979±0.005	**0.987±0.004**	0.943±0.008	**0.989±0.003**	0.869±0.015	0.959±0.008
letter	**0.911±0.012**	0.835±0.013	**0.862±0.013**	0.718±0.015	**0.883±0.011**	0.445±0.022	0.767±0.014
shuttle	**0.996±0.001**	0.966±0.003	**0.993±0.002**	0.929±0.003	**0.989±0.003**	0.854±0.069	0.966±0.002

(b) *Average*-F1

	OvO		OvR		ECOC		
Data Set	Specific	Original	Specific	Original	Specific	Original	Softmax
iris	0.943±0.058	**0.951±0.059**	**0.934±0.067**	0.891±0.077	**0.901±0.116**	0.681±0.071	0.930±0.070
wine	0.979±0.035	**0.980±0.032**	0.979±0.035	**0.989±0.024**	**0.970±0.041**	0.969±0.042	0.977±0.037
glass	**0.540±0.130**	0.532±0.132	**0.592±0.164**	0.483±0.110	**0.524±0.124**	0.422±0.097	0.528±0.123
vowel	**0.849±0.042**	0.771±0.066	**0.771±0.051**	0.495±0.078	**0.796±0.051**	0.313±0.074	0.599±0.061
dna	**0.923±0.022**	0.921±0.016	0.920±0.023	**0.936±0.024**	0.892±0.029	**0.904±0.024**	0.928±0.019
satimage	**0.873±0.017**	0.827±0.021	**0.862±0.015**	0.753±0.021	**0.864±0.018**	0.688±0.033	0.812±0.025
usps	0.961±0.007	0.961±0.009	0.942±0.008	**0.945±0.012**	**0.955±0.007**	0.906±0.012	0.947±0.009
pendigits	**0.990±0.004**	0.979±0.004	**0.987±0.004**	0.942±0.008	**0.988±0.003**	0.866±0.015	0.959±0.007
letter	**0.909±0.012**	0.832±0.014	**0.859±0.013**	0.712±0.015	**0.882±0.011**	0.406±0.023	0.762±0.014
shuttle	**0.761±0.053**	0.625±0.078	**0.650±0.083**	0.511±0.064	**0.562±0.093**	0.374±0.124	0.602±0.068

Table 4 Wilcoxon signed-ranks test for "Specific" against "Original" (at 0.05 significance level; p-values shown in the brackets).

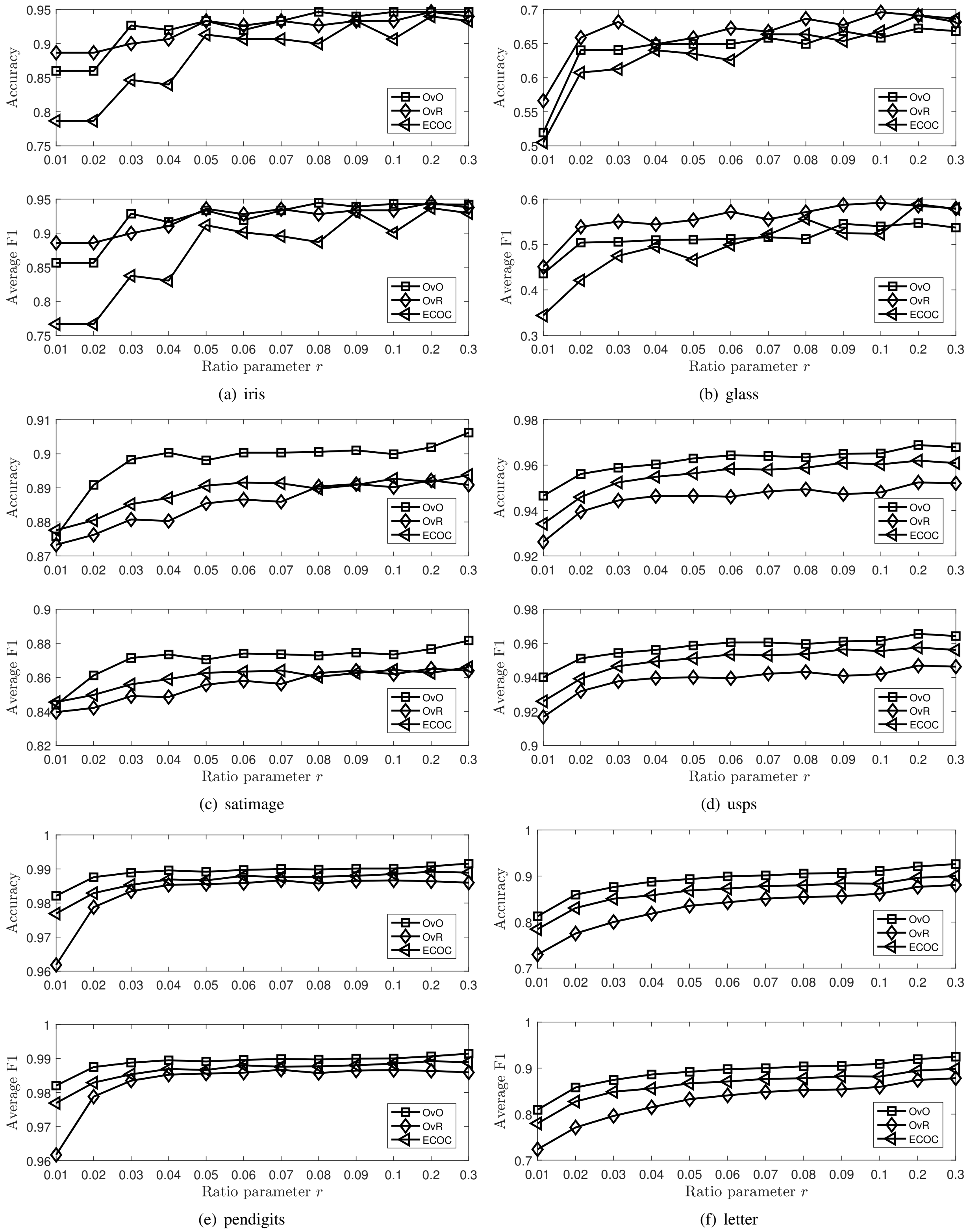
Decomposition	SVM		LR	
Strategy	*Accuracy*	*Average*-F1	*Accuracy*	*Average*-F1
OvO	**win**[9.77e-03]	**win**[1.95e-02]	**win**[1.95e-02]	**win**[3.71e-02]
OvR	**win**[1.95e-03]	**win**[5.86e-03]	**win**[2.73e-02]	**win**[2.73e-02]
ECOC	**win**[1.95e-03]	**win**[1.95e-03]	**win**[7.81e-03]	**win**[5.86e-03]

4.2 Experimental Results

The detailed experimental results are reported in Tables 2-3. For convenient comparison, the better experimental result of "Specific" and "Original" over each data set for each decomposition strategy is shown in bold face. Moreover, for each decomposition strategy in terms of each evaluation metric, to further show whether "Specific" is statistically better than "Original" over the whole benchmark data sets, *Wilcoxon signed-ranks test* [36] is used as the statistical test tool (the significance level is set to 0.05). Accordingly, the test results are summarized in Table 4 where the p-values are also shown in the brackets.

According to the statistical test results in Table 4, it is shown that no matter SVM or LR is employed as the binary classifier, "Specific" always achieves statistically better performance than "Original" over the whole benchmark data sets in terms of both the two evaluation metrics for all the three decomposition strategies. These experimental results clearly validate the effectiveness of learning label-specific features for decomposed binary classification problems in decomposition-based MCC solutions.

According to the detailed experimental results in Tables 2-3, it can be observed that "Specific" achieves superior performance against "Original" over most data sets for all the three strategies with either SVM or LR being the binary classifier. However, it is shown that the performance relationships between "Specific" and "Original" are related to bi-



(a) iris

(b) glass

(c) satimage

(d) usps

(e) pendigits

(f) letter

Fig. 1 Performance of the three decomposition-based MCC methods change when the value of *r* increases from 0.01 to 0.3.

nary classifier, data set, decomposition strategy and evaluation metrics. For example, on data set `wine`, when SVM is employed, “Specific” achieves superior performance against “Original” in all cases, while when LR is employed, “Specific” only achieves superior performance against “Original” for ECOC in terms of *Average*-F1. For another example,

Table 5 The experimental results (minimum/mean±std/maximum) for stability analysis where the employed binary classifier is SVM.

(a) *Accuracy*

Data Set	OvO	OvR	ECOC	MSVM
iris	0.940/0.943±0.003/0.947	0.940/0.945±0.005/0.953	0.933/0.945±0.007/0.953	0.947±0.053
wine	0.972/0.974±0.003/0.977	0.972/0.978±0.004/0.983	0.966/0.975±0.005/0.983	0.966±0.039
glass	0.636/0.649±0.009/0.664	0.640/0.672±0.015/0.692	0.668/0.675±0.007/0.692	0.594±0.096
vowel	0.869/0.889±0.013/0.911	0.701/0.714±0.012/0.742	0.852/0.862±0.008/0.879	0.703±0.071
dna	0.875/0.905±0.016/0.924	0.896/0.910±0.008/0.926	0.782/0.858±0.030/0.887	0.916±0.023
satimage	0.849/0.864±0.013/0.886	0.836/0.877±0.016/0.889	0.888/0.894±0.003/0.898	0.858±0.021
usps	0.952/0.956±0.003/0.960	0.920/0.934±0.008/0.945	0.959/0.961±0.001/0.963	0.944±0.011
pendigits	0.983/0.988±0.002/0.991	0.974/0.985±0.006/0.992	0.992/0.992±0.000/0.993	0.965±0.008
letter	0.898/0.903±0.003/0.909	0.816/0.837±0.010/0.850	0.825/0.835±0.005/0.841	0.781±0.016
shuttle	0.992/0.996±0.002/0.998	0.997/0.997±0.000/0.997	0.997/0.997±0.000/0.997	0.980±0.001

(b) *Average*-F1

Data Set	OvO	OvR	ECOC	MSVM
iris	0.933/0.938±0.005/0.944	0.935/0.942±0.007/0.952	0.928/0.941±0.008/0.952	0.941±0.058
wine	0.968/0.970±0.003/0.974	0.969/0.974±0.004/0.979	0.963/0.972±0.005/0.980	0.960±0.045
glass	0.539/0.556±0.009/0.568	0.529/0.562±0.023/0.598	0.553/0.573±0.014/0.598	0.557±0.114
vowel	0.875/0.889±0.013/0.911	0.681/0.698±0.016/0.735	0.852/0.861±0.008/0.877	0.664±0.077
dna	0.861/0.890±0.018/0.912	0.871/0.894±0.012/0.914	0.758/0.842±0.033/0.875	0.906±0.025
satimage	0.816/0.832±0.012/0.854	0.802/0.845±0.018/0.861	0.858/0.866±0.005/0.873	0.782±0.020
usps	0.946/0.951±0.003/0.956	0.916/0.930±0.008/0.941	0.954/0.956±0.001/0.958	0.937±0.013
pendigits	0.983/0.988±0.002/0.991	0.973/0.985±0.006/0.992	0.992/0.992±0.000/0.993	0.965±0.008
letter	0.895/0.902±0.004/0.907	0.825/0.842±0.008/0.853	0.827/0.836±0.004/0.841	0.775±0.015
shuttle	0.783/0.815±0.016/0.845	0.718/0.745±0.019/0.778	0.666/0.697±0.019/0.722	0.660±0.067

Table 6 The experimental results (minimum/mean±std/maximum) for stability analysis where the employed binary classifier is LR.

(a) *Accuracy*

Data Set	OvO	OvR	ECOC	Softmax
iris	0.933/0.942±0.005/0.953	0.920/0.930±0.007/0.940	0.900/0.916±0.010/0.933	0.933±0.070
wine	0.972/0.979±0.006/0.983	0.977/0.982±0.002/0.983	0.972/0.977±0.003/0.983	0.983±0.027
glass	0.650/0.661±0.008/0.673	0.663/0.682±0.012/0.705	0.640/0.656±0.012/0.673	0.603±0.108
vowel	0.850/0.859±0.005/0.869	0.763/0.785±0.011/0.801	0.782/0.791±0.009/0.807	0.635±0.061
dna	0.931/0.936±0.003/0.941	0.926/0.929±0.002/0.932	0.903/0.908±0.003/0.912	0.937±0.016
satimage	0.900/0.901±0.001/0.903	0.888/0.890±0.001/0.892	0.886/0.889±0.001/0.891	0.860±0.021
usps	0.965/0.966±0.001/0.968	0.948/0.949±0.001/0.950	0.960/0.961±0.001/0.962	0.953±0.008
pendigits	0.990/0.990±0.000/0.991	0.986/0.987±0.001/0.988	0.988/0.988±0.000/0.989	0.959±0.008
letter	0.908/0.909±0.001/0.910	0.860/0.862±0.002/0.864	0.882/0.883±0.001/0.885	0.767±0.014
shuttle	0.995/0.995±0.000/0.996	0.992/0.993±0.000/0.994	0.989/0.990±0.001/0.991	0.966±0.002

(b) *Average*-F1

Data Set	OvO	OvR	ECOC	Softmax
iris	0.933/0.939±0.006/0.952	0.918/0.929±0.008/0.941	0.897/0.912±0.010/0.931	0.930±0.070
wine	0.967/0.974±0.006/0.981	0.974/0.978±0.002/0.979	0.969/0.974±0.003/0.980	0.977±0.037
glass	0.537/0.547±0.008/0.558	0.553/0.584±0.019/0.624	0.530/0.552±0.017/0.576	0.528±0.123
vowel	0.845/0.856±0.006/0.866	0.759/0.779±0.011/0.793	0.772/0.786±0.010/0.800	0.599±0.061
dna	0.920/0.926±0.003/0.931	0.916/0.919±0.002/0.923	0.890/0.894±0.003/0.899	0.928±0.019
satimage	0.873/0.875±0.001/0.877	0.860/0.862±0.001/0.865	0.856/0.860±0.002/0.862	0.812±0.025
usps	0.961/0.963±0.001/0.965	0.942/0.943±0.001/0.944	0.955/0.957±0.001/0.958	0.947±0.009
pendigits	0.990/0.990±0.000/0.990	0.986/0.987±0.001/0.988	0.988/0.988±0.000/0.989	0.959±0.007
letter	0.907/0.908±0.001/0.909	0.857/0.860±0.002/0.862	0.880/0.882±0.001/0.883	0.762±0.014
shuttle	0.746/0.753±0.007/0.763	0.622/0.650±0.014/0.665	0.529/0.543±0.010/0.561	0.602±0.068

on data set `usps` and when LR is employed, “Specific” achieves superior/equal/inferior performance against “Original” for ECOC/OvO/OvR strategy. Therefore, in real-world applications, different label-specific features generation techniques should be developed with considering the MCC solution’s specific characteristics, including the binary classifier, decomposition strategy, the evaluation metric, etc.

Besides, as shown in Tables 2-3, with the same decomposition strategy, there are many cases that "Specific" achieves superior performance against the corresponding direct multi-class classifier while "Original" achieves inferior performance against the corresponding direct multi-class classifier (e.g., the experimental results over data set `shuttle` when the employed binary classifier is SVM). These experimental results also validate the effectiveness of learning label-specific features for decomposed binary classification problems in decomposition-based MCC solutions.

4.3 Further Analysis

4.3.1 Parameter Sensitivity Analysis

As stated in Subsection 3.2, there is one parameter in the procedure of learning label-specific features, i.e., the ratio r to determine the number of clusters. To investigate how this parameter affects the performance, sensitivity analysis is conducted w.r.t. the ratio r. Specifically, Fig. 1 illustrates how the performance of the three decomposition-based MCC methods fluctuate when the value of r increases from 0.01 to 0.3. It can be observed that the performance of these methods will be improved with increasing r in general. Besides, it can also be observed that the performance will become stable or be improved very slowly with large r. However, the larger the value of r, the higher the dimension of label-specific features, which means higher computational complexity of model induction. In previous comparative studies, the value of r is fixed as 0.1 to make a compromise between performance and computational complexity.

4.3.2 Algorithmic Stability Analysis

As stated in Subsection 3.2, the specific information about each decomposed binary data set is characterized by clustering techniques, and the clustering procedure is simply done by the traditional k-means algorithm. As the clusters identified by k-means are unstable and then the label-specific features will be different for the same decomposed binary data set. To show this influence, we repeat the experiments ten times with different random seeds to obtain different clustering results. As mentioned before, ten-fold cross validation is conducted over each data set for each compared method. When the experiments are repeated, we only record the mean value for each evaluation metric and then obtain ten mean values over each data set in terms of each evaluation metric. Tables 5-6 report the minimum, mean and standard deviation, maximum of the ten mean values. The experimental results of multi-class SVM and softmax regression are also given for reference purpose. It is shown that, in general, the larger the data set, the more stable the experimental results. One exception is the data set `shuttle`, which has a relatively large standard derivation. The possible reason is that this data set suffers severe class imbalance [37]. Although it has a total of 43500 samples, the second, third, sixth, seventh classes only have 37, 132, 6, 11 samples, respectively.

5 Conclusion

The main contribution of this paper is proposing to learn label-specific features for each decomposed binary classification problem in decomposition-based MCC solutions, which suggests a novel perspective to learn decomposition-based multi-class classifier. To put this idea into practice, we generate label-specific features via clustering analysis for each binary classification problem. Experimental results clearly validate the effectiveness of such strategy.

This paper only represents a preliminary attempt towards learning label-specific features for decomposition-based MCC solutions. In the future, it is interesting to explore other feasible techniques to obtain label-specific features, e.g., feature selection. Besides, it is also interesting to further utilize the specific information residing in different decomposition strategies.

References

1. Zhou Z H. Machine Learning. Singapore: Springer Nature, 2021
2. Han J, Pei J, Tong H. Data Mining: Concepts and Techniques. 4th ed. Cambridge: Morgan kaufmann, 2022
3. Zhao L, Song Y, Zhu Y, Zhang C, Zheng Y. Face recognition based on multi-class SVM. In: Proceedings of the Chinese Control and Decision Conference. 2009, 5871–5873
4. Wu K, Jia F, Han Y. Domain-specific feature elimination: multi-source domain adaptation for image classification. Frontiers of Computer Science, 2023, 17(4):174705
5. Wang T Y, Chiang H M. Fuzzy support vector machine for multi-class text categorization. Information Processing and Management, 2007, 43(4):914–929
6. Moreo A, Esuli A, Sebastiani F. Word-class embeddings for multi-class text classification. Data Mining and Knowledge Discovery, 2021, 35(3):911–963
7. Frid A, Manevitz L, Mosafi O. Multi-class classification in parkinson's disease by leveraging internal topological structure of the data and of

the label space. In: Proceedings of the International Joint Conference on Neural Networks. 2019, 1–9

8. Wei K, Li T, Huang F, Chen J, He Z. Cancer classification with data augmentation based on generative adversarial networks. Frontiers of Computer Science, 2022, 16(2):162601
9. Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(7):1079–1089
10. Zhang M L, Li Y K, Yang H, Liu X Y. Towards class-imbalance aware multi-label learning. IEEE Transactions on Cybernetics, 2022, 52(6):4459–4471
11. Read J, Martino L, Luengo D. Efficient monte carlo methods for multi-dimensional learning with classifier chains. Pattern Recognition, 2014, 47(3):1535–1546
12. Jia B B, Zhang M L. Multi-dimensional classification via stacked dependency exploitation. Science China Information Sciences, 2020, 63(12):222102
13. Lorena A C, Carvalho A, Gama J. A review on the combination of binary classifiers in multiclass problems. Artificial Intelligence Review, 2008, 30(1-4):19–37
14. Hsu C W, Lin C J. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 2002, 13(2):415–425
15. Duan K, Keerthi S. Which is the best multiclass SVM method? an empirical study. In: Proceedings of the 6th International Workshop on Multiple Classifier Systems. 2005, 278–285
16. Dietterich T, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 1995, 2:263–286
17. Allwein E, Schapire R, Singer Y. Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research, 2000, 1:113–141
18. Pujol O, Radeva P, Vitrià J. Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(6):1007–1012
19. Escalera S, Tax D, Pujol O, Radeva P, Duin R. Subclass problem-dependent design for error-correcting output codes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(6):1041–1054
20. Escalera S, Pujol O, Radeva P. On the decoding process in ternary error-correcting output codes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(1):120–134
21. Pujol O, Escalera S, Radeva P. An incremental node embedding technique for error correcting output codes. Pattern Recognition, 2008, 41(2):713–725
22. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11):2278–2324
23. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20(3):273–297
24. Liu J Y, Jia B B. Combining one-vs-one decomposition and instance-based learning for multi-class classification. IEEE Access, 2020, 8:197499–197507
25. Wang Z, Xue X. Multi-class support vector machine. In: Ma Y, Guo G, eds. Support Vector Machines Applications, Springer, Cham, 2014, 23–48
26. Hastie T, Rosset S, Zhu J, Zou H. Multi-class adaboost. Statistics and Its Interface, 2009, 2(3):349–360
27. Zheng F, Xue H, Chen X, Wang Y. Maximum margin tree error correcting output codes. In: Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence. 2016, 681–691
28. Zheng F, Xue H. Subclass maximum margin tree error correcting output codes. In Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence. 2018, 454–462
29. Kang S, Cho S, Kang P. Constructing a multiclass classifier using one-against-one approach with different binary classifiers. Neurocomputing, 2015, 149:677–682.
30. Liu M, Zhang D, Chen S, Xue H. Joint binary classifier learning for ECOC-based multi-class classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(11):2335–2341.
31. Zhang M L, Wu L. LIFT: multi-label learning with label-specific features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(1):107–120.
32. Jain A, Murty M N, Flynn P. Data clustering: a review. ACM Computing Surveys, 1999, 31(3):264–323.
33. Fan R E, Chang K W, Hsieh C J, Wang X R, Lin C J. LIBLINEAR: a library for large linear classification. Journal of Machine Learning Research, 2008, 9:1871–1874.
34. Crammer K, Singer Y. On the algorithmic implementation of multi-class kernel-based vector machines. Journal of Machine Learning Research, 2001, 2:265–292.
35. Dobson A, Barnett A. An Introduction to Generalized Linear Models. 4th ed. Boca Raton: Chapman & Hall/CRC, 2018
36. Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7:1–30, 2006.
37. Wang S,Yao X. Multiclass imbalance problems: analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 2012, 42(4):1119–1130

Bin-Bin Jia received the bachelor's degree from North China Electric Power University in 2010, and the master's degree from Beihang University in 2013. He joined Lanzhou University of Technology in 2013 and is an assistant professor currently. From Sept. 2017 to Mar. 2022, he studied in Southeast University where he obtained the Ph.D. degree. His main research interests include machine learning and data mining.

Jun-Ying Liu received the bachelor's degree from North China Electric Power University, China, in 2010, and the master's degree from Beijing Jiaotong University, China, in 2012. Currently, she is an assistant professor at the College of Electrical and Information Engineering, Lanzhou University of Technology. Her main research interests include machine learning and data mining.

Jun-Yi Hang received the BSc and MSc degrees from Beihang University, China, in 2017 and 2020 respectively. Currently, he is a PhD student at the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining, especially in learning from multi-label data.

Min-Ling Zhang received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 2001, 2004 and 2007, respectively. Currently, he is a Professor at the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. In recent years, Dr. Zhang has served as the General Co-Chairs of ACML'18, Program Co-Chairs of PAKDD'19, CCF-ICAI'19, ACML'17, CCFAI'17, PRICAI'16, Senior PC member or Area Chair of AAAI 2017-2020, IJCAI 2017-2021, KDD 2021, ICDM 2015-2020, etc. He is also on the editorial board of ACM Transactions on Intelligent Systems and Technology, Neural Networks, Science China Information Sciences, Frontiers of Computer Science, etc. Dr. Zhang is the Steering Committee Member of ACML and PAKDD, secretary-general of the CAAI Machine Learning Society, standing committee member of the CCF Artificial Intelligence & Pattern Recognition Society. He is a Distinguished Member of CCF, CAAI, and Senior Member of ACM, IEEE.