Machine Learning (2019) 108:809–830
https://doi.org/10.1007/s10994-019-05786-2

N-ary decomposition for multi-class classification

Joey Tianyi Zhou[1] · Ivor W. Tsang[2] · Shen-Shyang Ho[3] · Klaus-Robert Müller[4,5]

Received: 27 February 2018 / Accepted: 9 January 2019 / Published online: 20 February 2019

Abstract
A common way of solving a multi-class classification problem is to decompose it into a collection of simpler two-class problems. One major disadvantage is that with such a binary decomposition scheme it may be difficult to represent subtle between-class differences in many-class classification problems due to limited choices of binary-value partitions. To overcome this challenge, we propose a new decomposition method called N-ary decomposition that decomposes the original multi-class problem into a set of simpler multi-class subproblems. We theoretically show that the proposed N-ary decomposition could be unified into the framework of error correcting output codes and give the generalization error bound of an N-ary decomposition for multi-class classification. Extensive experimental results demonstrate the state-of-the-art performance of our approach.

Keywords Ensemble learning · Multi-class classification · N-ary ECOC

1 Introduction

Many real-world problems are multi-class in nature. To handle multi-class problems, many approaches have been proposed. One research direction focuses on solving multi-class

Editors: Masashi Sugiyama, Yung-Kyun Noh.

✉ Joey Tianyi Zhou
zhouty@ihpc.a-star.edu.sg

Ivor W. Tsang
ivor.tsang@gmail.com

Shen-Shyang Ho
hos@rowan.edu

Klaus-Robert Müller
klaus-robert.mueller@tu-berlin.de

1 Institute of High Performance Computing, A*STAR, Singapore, Singapore

2 Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia

3 Rowan University, Camden, NJ, USA

4 Machine Learning Laboratory, Berlin Institute of Technology, Berlin, Germany

5 Department of Brain and Cognitive Engineering, Korea University, Seoul, Republic of Korea

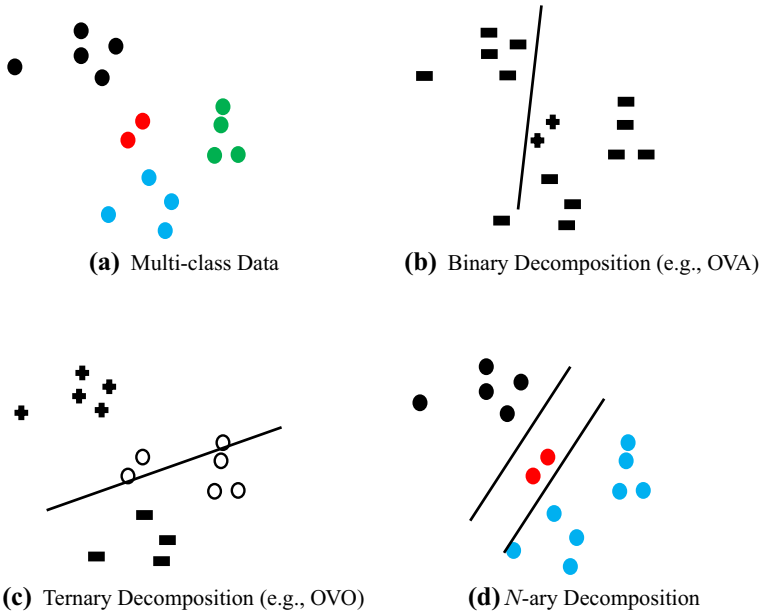
problems directly. These approaches include decision tree based methods (Quinlan 1986; Beygelzimer et al. 2009; Su and Zhang 2006; Bengio et al. 2010; Yang and Tsang 2011; Gao and Koller 2011; Deng et al. 2011). In particular, decision-tree based algorithms label each leaf of the decision tree with one of the N_C classes, and internal nodes can be selected to discriminate between these classes. The performance of decision-tree based algorithms heavily depends on the internal tree structure. Thus, these methods are usually vulnerable to outliers. To achieve better generalization, Yang and Tsang (2011) and Gao and Koller (2011) propose to learn the decision tree structure based on the large margin criterion. However, these algorithms usually involve solving sophisticated optimization problems and their training time increases dramatically with the increase of the number of classes. Contrary to these complicated methods, K-Nearest Neighbour (KNN) (Cover and Hart 1967) is a simple but effective and stable approach to handle multi-class problems. However, KNN is sensitive to noise features and can therefore suffer from the curse-of-dimensionality. Meanwhile, Crammer and Singer (2002) and Jenssen et al. (2012) propose a direct approach for learning multi-class support vector machines (M-SVM) by deriving the generalized notion of margins as well as separating hyperplanes.

Another research direction focuses on the framework of the ensemble of binary classifiers. It first decomposes a multi-class problem into multiple binary problems , and then one can reuse the well-studied binary classification algorithms for their simplicity and efficiency. To obtain base classifiers, different decomposition strategies can be found in the literature (Galar et al. 2011; Rocha and Goldenstein 2014). The most common strategies are called binary decomposition such as "one-vs-all" (OVA) (Knerr et al. 1990) and ternary decomposition such as "one-vs-one" (OVO) (Galar et al. 2011). To this end, more general decomposition strategies have been developed, for instance, error correcting output codes (ECOC) approaches (Dietterich and Bakiri 1991; Liu et al. 2013; Rocha and Goldenstein 2014; Zhao and Xing 2013; Übeyli 2007; García-Pedrajas and Ortiz-Boyer 2011) have been proposed in recent years to design codes to enable a good partition scheme. One of most popular ways is to use random dense/sparse binary matrix to represent class assignment in each subproblem and it is able to correct errors committed in some base classifiers through the final results aggregation. For aggregation strategy, there are a number of methods are developed to combine the outputs of the base classifiers, such as probability estimates (Wu et al. 2004), binary-tree based strategies (Fei and Liu 2006) and dynamic classification schemes (Hong et al. 2008).

Though all the above-mentioned approaches endeavor to enhance the partition strategy for classification tasks, their designs of base learners are confined to binary classification. In the more challenging real-world applications, there exists multi-class problems where some of the classes are very similar and difficult to differentiate with each other. The existing binary partition schemes cannot handle this challenge due to limited choices . It is highly possible that some classes are assigned with same or similar codes.

To address this issue, we investigate whether one can extend the existing binary decomposition scheme to an *N-ary* decomposition scheme to (i) allow users the flexibility to choose N to construct the subclass in order to (ii) improve the classification performance for a given dataset. The main idea of our scheme is to decompose the original multi-class problem into a series of smaller multi-class subproblems instead of binary classification problems. To make it clearer, we first define a *meta-class* as follows,

Definition 1 A meta-class is defined as a subset of classes such that the original classes are partitioned into different meta-classes.



(a) Multi-class Data **(b)** Binary Decomposition (e.g., OVA)

(c) Ternary Decomposition (e.g., OVO) **(d)** N-ary Decomposition

Fig. 1 Fail cases of existing decomposition scheme: Figure **a** illustrates the original multi-class data (i.e., black, red, blue and green). The binary decomposition scheme sometimes creates non-separate binary classification problems as shown in **b**. The ternary decomposition sometimes creates cases where the data from the same class is assigned to different classes as shown in **c**. N-ary decomposition scheme decomposes the difficult task into some smaller and easier tasks as shown in **d** (Color figure online)

Figure 1 is drawn to illustrate the meta-class. Suppose that all original classes (i.e., black, red, blue and green) can be merged into a series of large meta-classes (i.e., black, red, blue). So, in each level, there is a classifier to divide the data into two N smaller meta-classes ($N = 3$ in Fig. 1d). Here we only consider two levels. Based on the definition of meta-classes, this N-*ary* decomposition scheme is deemed as a divide-and-conquer method.

More interestingly, we revisit N-ary decomposition for multi-class classification from the perspective of ECOC. Each partition scheme corresponds to a specific coding matrix M. We find that the performance of ensemble based methods relies on the minimum distance, $\Delta_{\min}(M)$, between any distinct pair of rows in the coding matrix M. A larger $\Delta_{\min}(M)$ is more likely to rectify the errors committed by individual base classifiers (Allwein et al. 2001). We further theoretically investigate the impact of the error correcting capability of different decomposition strategies for multi-class classification and show that N-*ary* decomposition has more advantages in correcting errors than binary decomposition strategy for multi-class classification.

The main contributions of this paper are as follows.

– We propose a novel N-ary decomposition scheme that achieves a large expected distance between any pair of rows in M at a reasonable $N(> 3)$ for a multi-class problem (see Sect. 3). The two main advantages of such a decomposition scheme are as follows: (i) the ability to construct more discriminative codes and (ii) the flexibility for the user to select the best N for ensemble-based multi-class classification. In light of this approach, class binarization techniques are considered special cases of the N-ary decomposition.

– We provide theoretical insights on the dependency of the generalization error bound of N-ary decomposition on the average base classifier generalization error and the minimum distance between any two constructed codes (see Sect. 5). Furthermore, we conduct a series of empirical analysis to verify the validity of the theorem on the error bound (see Sect. 6).
– We show empirically that the optimal N (based on classification performance) lies in [3, 10] with a slight trade-off in computational cost (see Sect. 6).
– We show empirically the superiority of the proposed decomposition scheme over the state-of-the-art coding methods for multi-class prediction tasks on a set of benchmark datasets (see Sect. 6).

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the generalization from binary to N-ary decomposition scheme. In Sect. 4, we give the complexity analysis of N-ary decomposition and compare it with other schemes with the SVM classifier as a showcase. Section 5 gives the error bound analysis of N-ary decomposition. Finally, Sect. 6 discusses our empirical studies and Sect. 7 concludes this work.

2 Related work

Many decomposition strategy for multi-class classification (Allwein et al. 2001) have been proposed to design a good coding matrix M for partition assignment in recent years with $M_{ij} \in \{-1, 1, 0\}$, where $1/-1$ denotes the assigned positive/negative class, 0 denotes unselected class. Most of them fall into the following two categories, i.e., problem dependent decomposition, problem-independent decomposition (Rocha and Goldenstein 2014; Zhong and Cheriet 2013). Our proposed random decomposition belongs to the problem-independent decomposition. Here we mainly survey the problem-independent decomposition.

Problem-independent decomposition designs a good coding matrix for partition assignment independent of data, such as OVO, OVA, random binary/ternary decomposition (Dietterich 2000). However, the coding matrix design is not optimized for the training dataset or the instance labels. Therefore, these approaches usually require a large number of base classifiers generated by the pre-designed coding matrix. For example, the random ternary decomposition approach aims to construct the coding matrix $M \in \{-1, 1\}^{N_C \times N_L}$ where N_C is the number of classes, N_L is the code length, and its elements are randomly chosen as either 1 (positive class) or -1 (negative class) (Dietterich and Bakiri 1995). Allwein et al. (2001) extends this binary decomposition scheme to ternary decomposition by using a coding matrix $M \in \{-1, 0, 1\}^{N_C \times N_L}$ where the classes corresponding to 0 are not considered in the learning process. However, a random binary decomposition approach cannot guarantee that the created base binary classification task are always well designed and easily trained. Therefore, Allwein et al. (2001) suggest that binary/ternary decomposition approaches require at least $10 \log_2(N_C)$ and $15 \log_2(N_C)$ base classifiers, respectively, to achieve optimal results.

Problem-independent decomposition is the ensemble of binary classifiers, which has following advantages: (1) It is easy to use in practice without any ad-hoc design of coding matrix; (2) It can be parallelized due to independences of base tasks; (3) It enjoys a good theoretical guarantee on classification performance.

Due to the favorable properties and promising performance of problem-independent approaches for the classification task, they have been applied to real-world classification applications such as face verification (Kittler et al. 2003), ECG beats classification (Übeyli

2007), and even beyond multi-class problems, such as feature extraction (Zhong and Liu 2013) and fast similarity search (Yu et al. 2010).

Though all the above-mentioned variations of the binary decomposition endeavor to enhance the error correcting ability for the classification task, their designs are still based on aggregation of base binary classification results which lack some desirable properties available in their generalized form.

3 From a binary to *N*-ary decomposition

In this section, we discuss necessities and advantages of N-ary decomposition scheme from aspects of coding matrix and investigate the **column correlation of coding matrix** and **separation between codewords of different classes**.

Existing ensemble methods design the coding matrix and constrain the coding values either in $\{-1, 1\}$ (binary) or $\{-1, 0, 1\}$ (ternary) and train a number of different binary classifiers accordingly. A lot of studies show that when there are sufficient classifiers, ensemble-based multi-class classification algorithm can reach stable and reasonable performance (Rocha and Goldenstein 2014; Dietterich and Bakiri 1995). Nevertheless, binary and ternary codes can generate at most 2^{N_C} and 3^{N_C} binary classifiers, where N_C denotes the number of classes. On the other hand, due to limited choices of coding values, existing codes tend to create correlated and redundant classifiers and make them less effective "voters". Moreover, some studies show that binary and ternary codes usually require only $10 \log_2(N_C)$ and $15 \log_2(N_C)$ base classifiers, respectively, to achieve optimal results (Allwein et al. 2001). Furthermore, when the original multi-class problem is difficult, the existing coding schemes cannot handle well. For example, as shown in Fig. 1b, the binary decomposition that creates binary codes like OVA may result in difficult base binary classification tasks. The ternary decomposition (see Fig. 1c) may cause cases where the test data from the same class is assigned to different classes.

To address these issues, we extend the binary or ternary codes to N-ary codes. One example of the N-ary coding matrix to represent seven classes is shown in Table 1. Unlike the existing binary decomposition methods, a row of coding matrix M represents the code of each class and the code consists of N_L numbers in $\{1 \cdots N\}$, where $N > 3$; while a column M_s of M represents the N partitions of classes to be considered. To be specific, the N-ary ECOC approach consists of four main steps:

1. Generate an N-ary matrix M by uniformly random sampling from a range $\{1..N\}$ (e.g., Table 1).

Table 1 An example of N-ary coding matrix M with $N = 4$ and $N_L = 6$

	M_1	M_2	M_3	M_4	M_5	M_6
C_1	1	1	2	4	1	1
C_2	2	1	1	3	2	1
C_3	3	2	1	2	3	1
C_4	4	3	1	1	4	2
C_5	4	3	2	2	4	3
C_6	4	3	3	3	3	4
C_7	3	4	4	4	2	4

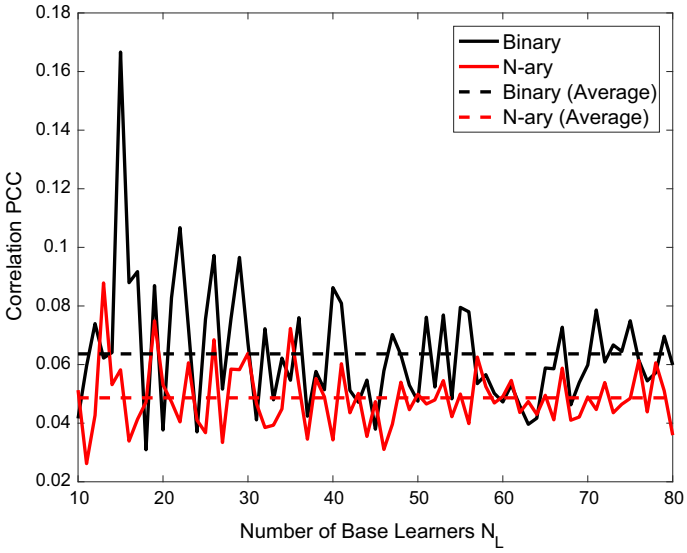


Fig. 2 Column correlation comparison (PCC vs. N_L)

2. For each of the N_L matrix columns, partition original training data into N groups based on the new class assignments and build an N-class classifier.
3. Given a test example $\mathbf{x}_t$, use the N_L classifiers to output N_L predicted labels for the testing output code (e.g., $f(\mathbf{x}_t) = [4, 3, 1, 2, 4, 2]$).
4. Final label prediction y_t for $\mathbf{x}_t$ is the nearest class based on minimum distance between the training and the testing output codes (e.g., $y_t = \arg\min_i d(f(\mathbf{x}_t), C_i) = 4$).

One notes that N-ary decomposition randomly breaks a large multi-class problem into a number of smaller multi-class subproblems. These subproblems are more complicated than binary problems and they incur additional computational cost. Hence, there is a trade-off between error correcting capability and computational cost.[1] Fortunately, our empirical studies indicate that N does not need to be too large to achieve good classification performance.

3.1 Column correlations of coding matrix

In the traditional binary decomposition strategy, it suggests longer codes, i.e., N_L is larger, however more binary base classifiers are likely to be more correlated. Thus, more base classifiers created by binary or ternary codes are not effective for final multi-class classification. To illustrate the advantage of N-ary decomposition in creating uncorrelated class assignment for base classifications, we conduct an experiment to investigate the column correlations of coding matrix M. The results are shown in Fig. 2. In the experiment, we set $N_C = 20$, $N = 5$, and N_L varies in [10, 80], and use Pearson's correlation (PCC) which is a normalized correlation measure that eliminates the scaling effect of the codes. From Fig. 2, we observe that N-ary decomposition achieves lower correlations for columns of coding matrix compared to conventional ternary codes. Especially, when the number of tasks is small, the correlations

[1] More complexity analyses can be found from Sect. 4.

over the created tasks for binary decomposition is significantly higher than that of the N-ary decomposition. Therefore, an N-ary decomposition not only provides more flexibility in creating a coding matrix, but also generates codes that are less correlated and less redundant, compared to traditional binary decomposition methods.

3.2 Separation between codewords of different classes

Apart from the column correlation, the row separation is another important measure to evaluate the error correcting ability of the coding matrix M (Dietterich and Bakiri 1995; Allwein et al. 2001). The codes for different classes are expected to be as dissimilar as possible. If codes (rows) for different classes are similar, it is easier to commit errors. Thus, the capability of error correction relies on the minimum distance, $\Delta_{\min}(M)$ or expected $\Delta(M)$ for any distinct pair of rows in the coding matrix $M \in \{1, 2, \ldots, N\}^{N_C \times N_L}$ where N_C is the number of classes, and N_L is the code length. Both the absolute distance and the Hamming distance can serve as the measure of row separation. The key difference between these two distances is that Hamming distance measures a scale-invariant difference. Specifically, the Hamming distance only cares about the number of different elements. It ignores the scale of the difference.

Hamming Distance One can use the *generalized Hamming distance* to calculate the $\Delta^{Ham}(M)$ for the existing coding schemes, which is defined as follows,

Definition 1 *(Generalized Hamming Distance)* Let $M(r_1, :)$, $M(r_2, :)$ denote row r_1, r_2 coding vectors in coding matrix M with length N_L, respectively. Then the generalized hamming distance can be expressed as

$$\Delta^{Ham}(M(r_1, :), M(r_2, :)) = \sum_{s=1}^{N_L} \begin{cases} 0 & \text{if } M(r_1, s) = M(r_2, s) \wedge M(r_1, s) \neq 0 \wedge M(r_2, s) \neq 0 \\ 1 & \text{if } M(r_1, s) \neq M(r_2, s) \wedge M(r_1, s) \neq 0 \wedge M(r_2, s) \neq 0 \\ 0.5 & \text{if } M(r_1, s) = 0 \vee M(r_2, s) = 0. \end{cases}$$

For the OVA coding, every two rows have exactly two entries with opposite signs, $\Delta_{min}^{Ham(OVA)}(M) = 2$. For the OVO coding, every two rows have exactly one entry with opposite signs, $\Delta_{min}^{Ham(OVO)}(M) = \left(\binom{N_C}{2} - 1\right)/2 + 1$, where N_C is the number of classes. Moreover, for a random coding matrix with its entries uniformly chosen, the expected value of any two different class codes is $\Delta^{Ham(RAND)}(M)$ is $N_L/2$, where N_L is the code length. A larger $\Delta^{Ham(RAND)}(M)$ is more likely to rectify the errors committed by individual base classifiers. Therefore, when $N_L \gg N_C$, a random coding matrix is expected to be more robust and rectify more errors than the OVO and OVA approaches (Allwein et al. 2001). However, the choice of only either binary or ternary codes hinders the construction of longer and more discriminative codes. For example, binary codes can only construct codes of length $N_L \leq 2^{N_C}$. Moreover, they lead to many redundant base learners. In contrast, for N-ary random matrix, the expected value of $\Delta^{Ham(N)}(M)$ is $N_L(1 - \frac{1}{N})$ (see Lemma 1 for proof). $\Delta^{Ham(N)}(M)$ is expected to be larger than $\Delta^{Ham(RAND)}(M)$ when $N \geq 3$ (Table 2).

Table 2 Comparison of distance of different codes

Decomposition strategies	Generalized Hamming distance	Absolute distance
OVA	2	4
OVO	$\left(\binom{N_C}{2} - 1\right)/2 + 1$	$2N_C - 2$
ECOC	$N_L/2$	N_L
N-ary	$N_L(1 - 1/N)$	$N_L(N^2 - 1)/3N$

Lemma 1 *The expected Hamming distance for any two distinct rows in a random N-ary coding matrix $M \in \{1, 2, \ldots, N\}^{N_C \times N_L}$ is*

$$\Delta^{Ham(N)}(M) = N_L \left(1 - \frac{1}{N}\right). \tag{1}$$

Proof Given a random matrix M with components chosen uniformly over $\{1, 2, \ldots, N\}$, for any distinct pair of entries in column s, i.e., $M(r_i, s)$ and $M(r_j, s)$, the probability of $M(r_i, s) = M(r_j, s)$ is $\frac{1}{N}$. Then the probability of $M(r_i, s) \neq M(r_j, s)$ is $1 - \frac{1}{N}$.

Therefore, according to Definition 1, the expected Hamming distance for M can be computed as follows,

$$\begin{aligned}\Delta^{Ham(N)}(M) &= N_L \left(1 \times \left(1 - \frac{1}{N}\right) + 0 \times \frac{1}{N}\right) \\ &= N_L \left(1 - \frac{1}{N}\right).\end{aligned}$$

□

Absolute distance Different from the Hamming distance, the absolute distance measures the difference scales. Thus, for a fair comparison, we assume that coding values are in the same scale for the absolute distance analysis. The definition of absolute distance is given as follows,

Definition 2 (*Absolute distance*) Let $M(r_1, :)$ and $M(r_2, :)$ denote row r_1 and r_2 coding vectors in coding matrix M with length N_L, respectively. Then the absolute distance can be expressed as

$$\Delta^{abs}(M(r_1, :), M(r_2, :)) = \sum_{s=1}^{N_L} |M(r_1, s) - M(r_2, s)|.$$

For the convenience of analysis, we first give the expected absolute distance for N-ary coding matrix in Lemma 2.

Lemma 2 *The expected absolute distance for any two distinct rows in a random N-ary coding matrix $M \in \{1, 2, \ldots, N\}^{N_C \times N_L}$ is*

$$\Delta^{abs(N)}(M) = N_L \frac{(N^2 - 1)}{3N}. \tag{2}$$

Proof Given a random matrix M with components chosen uniformly over $\{1, 2, \ldots, N\}$, for any distinct pair of entries in column s, i.e., $M(r_i, s)$, $M(r_j, s)$, we denote the corresponding expected absolute distance as $\Delta^{abs(N)}(M(:, s)) = \mathbb{E}\, d_{ij} = \mathbb{E}\, |M(r_i, s) - M(r_j, s)|$.

Table 3 All possible choices of d_{ij}

d_{ij}	$r_j = 1$	$r_j = 2$	$\cdots$	$r_j = N$
$r_i = 1$	0	1	$\cdots$	N − 1
$r_i = 2$	1	0	$\cdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	0	1
$r_i = N$	N − 1	$\cdots$	1	0

It can be calculated by averaging all the possible pairwise distances d_{ij} for $i, j \in \{1, 2, \ldots, N\}$. Since the two numbers r_i, r_j are chosen randomly from $\{1, ..., N\}$, $\Delta^N(M)$ can be expressed as follows:

$$\begin{aligned}\Delta^{abs(N)}(M(:,s)) &= \frac{1}{N^2}\sum_{i,j=1}^{N} d_{ij} \\ &= \frac{1}{N^2}\sum_{i,j=1}^{N} |M(r_i,s) - M(r_j,s)| \end{aligned} \tag{3}$$

First, we define the sequence a_n as follows:

$$a_n = (1 + 2 + \cdots + n) = \frac{n(n+1)}{2}. \tag{4}$$

Table 3 gives all the possible choices of d_{ij}. Thus the calculation of $\Delta^N(M)$ is equal to taking the average of all the entries in Table 3, which can be expressed as follows:

$$\begin{aligned}\Delta^{abs(N)}(M(:,s)) &= \frac{2}{N^2}(a_1 + a_2 + \cdots + a_{N-1}) \\ &= \frac{1}{N^2}(1 \times 2 + 2 \times 3 + \cdots + (N-1)N) \\ &= \frac{1}{N^2}\sum_{n=1}^{N}(n^2 - n) \\ &= \frac{1}{N^2}\left(\sum_{n=1}^{N} n^2 - \sum_{n=1}^{N} n\right) \\ &= \frac{1}{N^2}\left(\frac{N(N+1)(2N+1)}{6} - \frac{N(N+1)}{2}\right) \\ &= \frac{N^2-1}{3N}, \end{aligned} \tag{5}$$

where (5) comes from the symmetry of d_{ij}. Then

$$\Delta^{abs(N)}(M) = \sum_{s=1}^{N_L} \Delta^{abs(N)}(M(:,s)) = N_L \frac{(N^2-1)}{3N}.$$

□

For the OVA coding scheme, every two rows have exactly two entries with opposite signs, the minimum absolute distance $\Delta_{min}^{abs(OVA)}(M) = 4$; while for the OVO coding scheme, every two rows have exactly one entry with opposite signs and only $2N_C - 4$ entries with a difference of exactly one, $\Delta_{min}^{abs(OVO)}(M) = 2N_C - 2$. For binary random codes, the expected absolute distance between any two different rows is $\Delta^{abs(RAND)}(M) = N_L$. Thus, when N is large, $\Delta^{abs(N)}(M)$ is much larger than $\Delta^{abs(RAND)}(M)$, and N-ary coding is expected to be better.

The Hamming and absolute distance comparisons for different codes are summarized in Table 2. We can see that N-ary coding scheme has an advantage in creating more discriminative codes with larger distances for different classes in both two distance measures. This advantage is very important to analyze the generalization error analysis of N-ary ECOC.

4 Complexity comparison

As discussed in Sect. 3, N-ary codes have a better error correcting capability than the traditional random codes when N is larger than 3. However, one notes that the base classifier of each column is no longer solving a binary problem. Instead, we randomly break a large multi-class problem into a number of smaller multi-class subproblems. These subproblems are more complicated than binary problems and they incur additional computational cost. Hence, there is a trade-off between the error correcting capability and computational cost.

If the complexity of the algorithm employed to learn the small-size multi-class base problem is $\mathcal{O}(g(N, N_{tr}, d))$ with N classes, N_{tr} training examples, d predictive features and $g(N, N_{tr}, d)$ is the complexity function w.r.t N, N_{tr}, d, then the computational complexity of N-ary codes is $\mathcal{O}(N_L g(N, N_{tr}, D))$ for codes of length N_L.

Taking SVM as the base learner for example, one can learn each binary classification task created by binary coding matrix with training complexity of $\mathcal{O}(N_{tr}^3)$ for traditional SVM solvers that build on the quadratic programming (QP) problems. However, a major stumbling block for these traditional methods is in scaling up these QPs to large data sets, such as those commonly encountered in data mining applications. Thus, some state-of-the-art SVM implementations, e.g., LIBSVM (Chang and Lin 2011), Core Vector Machines (Tsang et al. 2005), have been proposed to reduce training time complexity from $\mathcal{O}(N_{tr}^3)$ to $\mathcal{O}(N_{tr}^2)$ and $\mathcal{O}(N_{tr})$, respectively. Nevertheless, how to efficiently train SVM is not the focus of our paper. For the convenience of complexity analysis, we use the time complexity of the traditional SVM solvers as the complexity of the base learners. Then, the complexity of binary codes is $\mathcal{O}(N_L N_{tr}^3)$. Different from existing decomposition method, one can directly address the multi-class problem in one single optimization process, e.g., multi-class SVM (Crammer and Singer 2002). This kind of model combines multiple binary-class optimization problems into one single objective function and simultaneously achieves the classification of multiple classes. In this way, the correlations across multiple binary classification tasks are captured in the learning model. The resulting QP optimization requires a complexity of $\mathcal{O}((N_C N_{tr})^3)$. However, it causes high computational complexity for a relatively large number of classes. In contrast, N-ary codes are in the complexity of $\mathcal{O}(N_L(N N_{tr})^3)$, where $N < N_C$. In this case, it achieves better trade-off between the error correcting capability and computational cost, especially for large class size N_C.

We summarize the time complexity of different codes in Table 4. In Sect. 6.1.4, our empirical studies indicate that N does not need to be too large to achieve optimal classification performance.

Table 4 Complexity analysis

Classifier	SVM
Binary decomposition	$\mathcal{O}(N_L N_{tr}^3)$
Direct multi-class	$\mathcal{O}((N_C N_{tr})^3)$
N-ary decomposition	$\mathcal{O}(N_L (N N_{tr})^3)$

5 Generalization analysis of *N*-ary decomposition for multi-class classification

In Sect. 5.1, we study the error correcting ability of an N-ary decomposition. In Sect. 5.2, we derive the generalization error bound for N-ary decomposition independent of the base classifier.

5.1 Analysis of error correcting on *N*-ary decomposition

To study the error correcting ability of N-ary decomposition, we first define the distance between the codes in any distinct pair of rows, $M(r_i)$ and $M(r_j)$, in an N-ary coding matrix M as $\Delta^N(M(r_i), M(r_j))$. It is the sum of the N_L distances between two entries, $M(r_i, s)$ and $M(r_j, s)$ in the same column s at two different rows, r_i and r_j, i.e., $\Delta^N(M(r_i), M(r_j)) = \sum_{s=1}^{N_L} \Delta^N(M(r_i, s), M(r_j, s))$.

We further define $\rho = \min_{r_i \neq r_j} \Delta^N(M(r_i), M(r_j))$ as the minimum distance between any two rows in M.

Proposition 1 *Given an N-ary coding matrix M and a vector of predicted labels $f(\mathbf{x}) = [f_1(\mathbf{x})), \ldots, f_{N_L}(\mathbf{x}))]$ by N_L base classifiers for a test instance $\mathbf{x}$. If $\mathbf{x}$ is misclassified by the N-ary ECOC decoding, then the distance between the correct label in $M(y)$ and $f(\mathbf{x})$ is greater than one half of ρ, i.e.,*

$$\Delta^N(M(y), f(\mathbf{x})) \geq \frac{1}{2}\rho. \tag{6}$$

Proof Suppose that the distance-based decoding incorrectly classifies a test instance $\mathbf{x}$ with known label y. In other words, there exists a label $r \neq y$ for which

$$\Delta^N(M(y), f(\mathbf{x})) \geq \Delta^N(M(r), f(\mathbf{x})).$$

Here, $\Delta^N(M(y), f(\mathbf{x}))$ and $\Delta^N(M(r), f(\mathbf{x}))$ can be expanded as the element-wise summation. Then, we have

$$\sum_{s=1}^{N_L} \Delta^N(M(y, s), f_s(\mathbf{x})) \geq \sum_{s=1}^{N_L} \Delta^N(M(r, s), f_s(\mathbf{x})). \tag{7}$$

Based on the above inequality, we obtain:

$$\begin{aligned} \Delta^N(M(y), f(\mathbf{x})) &= \frac{1}{2} \sum_{s=1}^{N_L} \left\{ \Delta^N(M(y, s), f_s(\mathbf{x})) + \Delta^N(M(y, s), f_s(\mathbf{x})) \right\} \\ &\geq \frac{1}{2} \sum_{s=1}^{N_L} \{ \Delta^N(M(y, s), f_s(\mathbf{x})) + \Delta^N(M(r, s), f_s(\mathbf{x})) \} \end{aligned} \tag{8}$$

$$\geq \frac{1}{2}\sum_{s=1}^{N_L}\{\Delta^N(M(y,s), M(r,s))\}$$
$$= \frac{1}{2}\Delta^N(M(r), M(y))$$
$$\geq \frac{1}{2}\rho, \tag{9}$$

where Inequality (8) uses Inequality (7) and Inequality (9) follows from the triangle inequality. □

Remark 1 From Proposition 1, one notes that a mistake on a test instance $(\mathbf{x}, y)$ implies that $\Delta^N(M(y), f(\mathbf{x})) \geq \frac{1}{2}\rho$. In other words, the prediction codes are not required to be exactly the same as the ground-truth codes for all the base classifications. As long as the distance is no larger than $\frac{1}{2}\rho$, N-ary coding can rectify the error committed by some base classifiers, and is still able to make an accurate prediction. This error correcting ability is very important especially when the labeled data is insufficient. Moreover, a larger minimum distance, i.e., ρ, leads to a stronger capability of error correcting. Note that this proposition holds for all the distance measures and traditional coding schemes due to the fact that only the triangle inequality is required in the proof.

5.2 Generalization error of *N*-ary decomposition

The next result provides a generalization error bound for *any* type of base classifier, such as the SVM classifier and decision tree, used in the N-nary decomposition for multi-class classification.

Theorem 1 (N-ary decomposition error bound) *Given N_L base classifiers, $f_1, \ldots, f_{N_L}$, trained on N_L subsets $\{(\mathbf{x}_i, M(y_i, s))_{i=1,\ldots,N_{tr}}\}_{s=1,\ldots,N_L}$ of the dataset with N_{tr} instances for coding matrix $M \in \{1, 2, \ldots, N\}^{N_C \times N_L}$. The generalized error rate for the N-ary ECOC approach using distance-based decoding is upper bounded by*

$$\frac{2N_L\bar{B}}{\rho}, \tag{10}$$

where $\bar{B} = \frac{1}{N_L}\sum_{s=1}^{N_L} B_s$ and B_s is the upper bound of the distance-based loss for the s^{th} base classifier.

Proof According to Proposition 1, for any misclassified data instance, the distance between its incorrect label vector $f(\mathbf{x})$ and the true label vector $M(y)$ should satisfy the minimal distance $\frac{\rho}{2}$, i.e., $\Delta^N(M(y), f(\mathbf{x})) = \sum_{s=1}^{N_L}\Delta^N(M(y,s), f_s(\mathbf{x})) \geq \frac{\rho}{2}$.

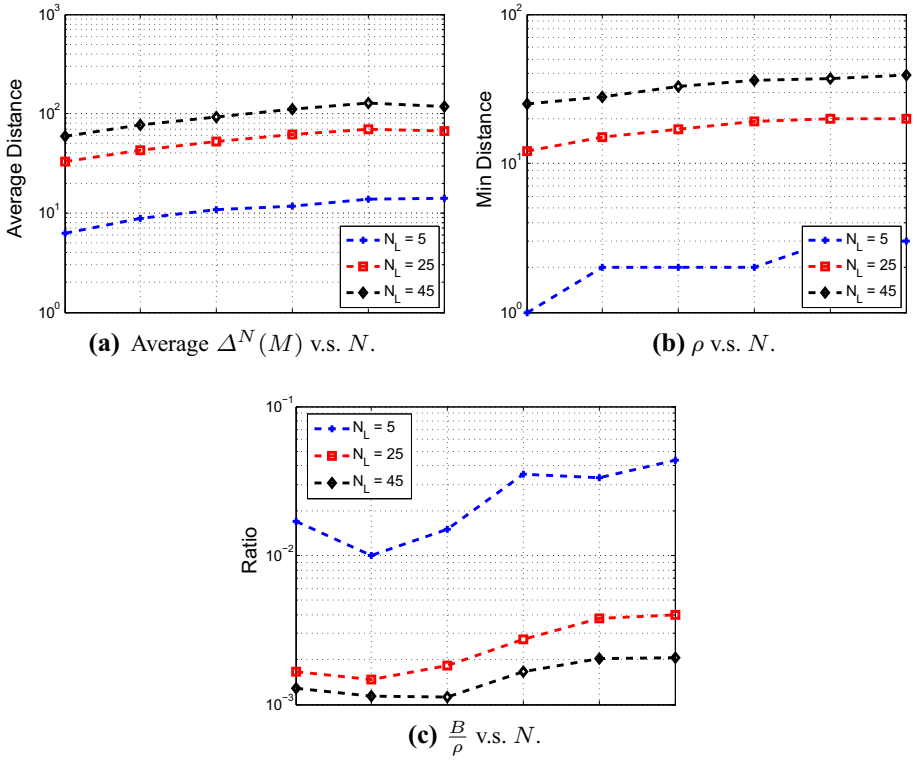
Let a be the number of incorrect label predictions for a set of test instances of size N_{te}. One obtains

$$a\frac{\rho}{2} \leq \sum_{i=1}^{N_{te}}\sum_{s=1}^{N_L}\Delta^N(M(y_i,s), f(\mathbf{x}_i)). \tag{11}$$

Then,

$$a \leq \frac{2N_{te}\sum_{s=1}^{N_L} B_s}{\rho} = \frac{2N_{te}N_L\bar{B}}{\rho}, \tag{12}$$

where $\bar{B} = \frac{1}{N_L}\sum_{s=1}^{N_L} B_s$.



(a) Average $\Delta^N(M)$ v.s. N. **(b)** ρ v.s. N.

(c) $\frac{B}{\rho}$ v.s. N.

Fig. 3 Experimental results to study error bound (Theorem 1) w.r.t. N

Hence, the testing error rate is bounded by $\frac{2N_L\bar{B}}{\rho}$. □

Remark 2 From Theorem 1, one notes that for the fixed N_L, the generalization error bound of the N-ary decomposition depends on the two following factors:

1. The averaged loss $\bar{B}$ for all the base classifiers. In practice, some base tasks may be badly designed due to the randomness. As long as the averaged loss $\bar{B}$ over all the tasks is small, the resulting ensemble classifier is still able to make a precise prediction.
2. The minimum distance ρ for coding matrix M. As we discussed in Proposition 1, the larger ρ, the stronger capability of error correcting N-ary code has.

Both two factors are affected by the choice of N. In particular, $\bar{B}$ increases as N increases since the base classification tasks become more difficult. On the other hand, from experimental results in Fig. 3b, it is observed that ρ becomes larger when N increases. Therefore, there is a tradeoff between these two factors.

6 Experimental results

We present experimental results on 11 well-known UCI multi-class datasets from a wide range of application domains. The statistics of these datasets are summarized in Table 5. The parameter N is chosen by cross-validation procedure. With the tuned parameters, all

Table 5 Summary of the datasets used in the experiments

Dataset	#Train	#Test	#Features	#Classes
Pendigits	3498	7494	16	10
Vowel	462	528	10	10
News20	3993	15935	62061	20
Letters	5000	15000	16	26
Auslan	1000	1565	128	95
Sector	3207	6412	55197	105
Aloi	50000	58000	128	1000
Glass	100	114	9	10
Satimage	3435	3000	36	7
Usps	4298	5000	256	10
Segment	1310	1000	19	7

methods are run ten realizations. Each has different random splittings with fixed training and testing size as given in Table 5. Our experimental results focus on the comparison of different encoding schemes rather than decoding schemes. Therefore, we fix generalized hamming distance as the decoding strategy for all the coding designs for a fair comparison.

To investigate the effectiveness of the proposed N-ary coding scheme, we compare it with data-independent coding schemes including OVO, OVA, and random binary encoding as well as the direct multi-class methods multi-class SVM (M-SVM) and decision tree. For the random binary encoding scheme, or ECOC in short, and the N-ary strategy, we select the matrix with the largest minimum absolute distance from 1000 randomly generated matrices.

To ensure a fair comparison and easy replication of results, the base learners decision tree CART (Breiman et al. 1984) and linear SVM are implemented with the CART decision tree MATLAB toolbox and the LIBSVM (Chang and Lin 2011) with the linear kernel in default settings, respectively.

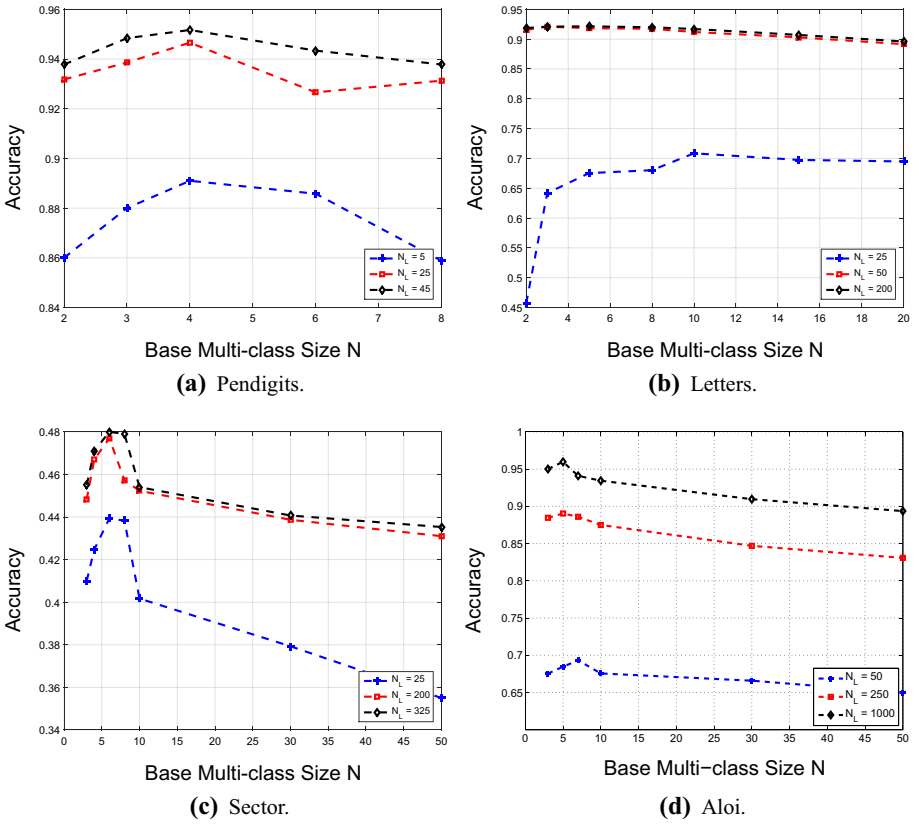
6.1 Error bound analysis on *N*-ary decomposition

In the bound analysis, we choose hamming distance 1 to measure the row separation as a showcase. According to Theorem 1, the generalization error bound depends on the minimum distance ρ between any two distinct rows in the N-ary coding matrix M as well as the average loss of base classifiers $\bar{B}$. In particular, the expected value of $\Delta^N(M)$ scales with $O(N)$.

In this subsection, we investigate the effect of the number of classes N using the Pendigits dataset with CART as the base classifier to illustrate the following aspects: (i) $\Delta^N(M)$ between any two distinct rows of codes (see Fig. 3a), (ii) ρ (see Fig. 3b), (iii) $\frac{\bar{B}}{\rho}$ (see Fig. 3c), and (iv) the classification performance (see Fig. 4). The empirical results corroborate the proposed error bounds in Theorem 1.

6.1.1 Average distance $\Delta^N(M)$ versus N

Recall that the hamming distance for different coding matrices discussed in Sect. 3 are: $\Delta^N(M) = N_L(1 - \frac{1}{N})$, $\Delta^{rand}(M) = N_L/2$, $\Delta^{ova}_{\min}(M) = 2$ and $\Delta^{ovo}_{\min}(M) = \left(\binom{N_C}{2} - 1\right)/2 + 1$.



(a) Pendigits. **(b)** Letters. **(c)** Sector. **(d)** Aloi.

Fig. 4 Accuracy versus N

From Fig. 3a, we observe that the empirical average hamming distances of the constructed N-ary coding matrices for random N-ary schemes are close to $N_L(1 - \frac{1}{N})$. Furthermore, when there are 45 base classifiers, the average distance for N-ary coding matrices is larger than 30, which is larger than that of the binary random codes with an average absolute distance of 22.5. Moreover, a higher N leads to a larger average distance. Comparing Fig. 3a, b, the large average distance $\Delta^N(M)$ also correlates with the large minimum distance ρ.

6.1.2 Minimum distance ρ versus N

For the Pendigits dataset with 10 classes, ρ for OVA and OVO are 4 and 18, respectively. From Fig. 3b, we observe that with a fixed number of base classifiers, ρ increases with the number of multi-class subproblems of class-size N, meanwhile ρ also increases with respect to the code length N_L. Furthermore, in comparison to the other coding schemes, our proposed method usually creates a coding matrix with a large ρ. For example, in Fig. 3b, one observes that when there are 25 and 45 base classifiers, the corresponding ρ for binary random codes are 0. On the other hand, N-ary decomposition, given a sufficiently large N, creates a coding matrix with ρ to be larger than 10 and 20, respectively. Although N-ary decomposition creates an N-ary coding matrix with a large ρ when N is larger, in real-world applications, it is preferred that N is not too large to ensure reasonable computational cost and difficulty of

subproblems. In short, N-ary decomposition provides a better alternative to creating a coding matrix with a large class separation compared to traditional coding schemes.

6.1.3 Ratio $\bar{B}/\rho$ versus N

Both $\bar{B}$ and ρ are dependent on N. Moreover, from the generalization error bound, we observe that $\bar{B}/\rho$ directly affects classification performance.

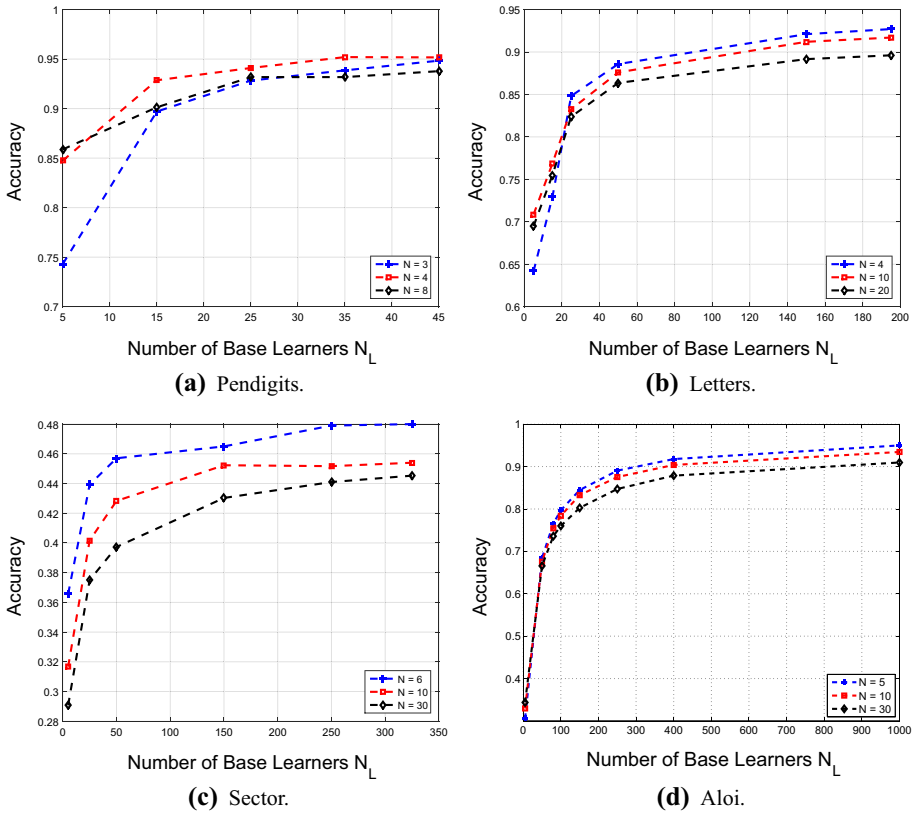
Hence, this ratio, which bounds the classification error, requires further investigation. Figure 3c shows that when $N = 4$, the ratio $\bar{B}/\rho$ is lowest. This observation suggests that the more the row and column separation of the coding matrix, the stronger the capability of error correction (Dietterich and Bakiri 1995). Therefore, N-ary decomposition is a better way to creating the coding matrix with large separation among the classes as well as more diversity, compared to the binary and ternary coding schemes. One notes that $\bar{B}/\rho$ starts to increase when $N \geq 5$. This means that the increase of the average base classifier loss $\bar{B}$ overwhelms the increase in ρ. The reason for this phenomena is the increase in difficulty of the subproblem classification with more classes.

6.1.4 Classification accuracy versus N

Next, we study the impact of N on the multi-class classification accuracy. We use datasets Pendigits, Letters, Sectors, Aloi with 10 classes, 26 classes, 105 classes, 1000 classes respectively as showcase. In order to a obtain meaningful analysis, we choose a suitable classifier for different datasets. In particular, we apply the CART to datasets Pendigits, Letters and Aloi and linear SVM to Sectors. One observes from Fig. 4 that the N-ary decomposition achieves competitive prediction performance when $3 \leq N \leq 10$. However, given sufficient base learners, the classification error starts increasing when N is large (e.g. $N > 4$ for Pendigits, $N > 5$ for Letters and $N > 8$ for Sector). This is because the base tasks are more challenging to solve when N is large and it indicates the influence of $\bar{B}$ outweighs that of ρ. Furthermore, one observes that the performance curves in Figs.3c and 4a roughly correlate to each other. Hence, one can estimate the trend in the empirical error using the ratio $\bar{B}/\rho$. This verifies the validity of the generalized error bound in Theorem 1. To investigate the choice of N on multi-class classification more comprehensively, we further conduct experiments on the other datasets. The results of datasets Pendigits, Letters, Sectors and Aloi are summarized in Fig. 4a–d, respectively. For the rest of the datasets, we have the similar observations. In general, smaller values of N ($N \in [3, 10]$) usually lead to reasonably competitive performance. In other words, the complexity of base learners for N-ary codes does not need to significantly increase above 3 for the performance to be better than existing binary or ternary coding approaches.

6.1.5 Classification accuracy versus N_L

From Fig. 5, we observe that high accuracy can be achieved with a small number of base learners. Another important observation is that given fewer base learners, it is better to choose a large value of N rather than a small N. This may be due to the fact that a larger N leads to stronger discrimination among codes as well as base learners. However, neither a large nor small N can reach optimal results given a sufficiently large N_L.



(a) Pendigits. **(b)** Letters. **(c)** Sector. **(d)** Aloi.

Fig. 5 Accuracy versus N_L

6.2 Comparison to state-of-the-art decomposition strategies

We compare our proposed N-ary decomposition to other popular decomposition strategies for with different base classifiers including decision tree (DT) (Breiman et al. 1984) and support vector machine (SVM) (Chang and Lin 2011).[2] The two binary classifiers can be easily extended to a multi-class setting. In particular, we use the multi-class SVM (M-SVM) (Crammer and Singer 2002) implemented with the MSVMpack (Lauer and Guermeur 2011). In addition to the multi-class extension of the two classifiers, we also compare N-ary decomposition strategies to OVO, OVA, random ternary decomposition strategies with the two binary classifiers. For random ternary and N-ary decomposition strategies, we report the best results with $N_L \leq N_C(N_C - 1)/2$, which is sufficient for conventional random decomposition methods to reach optimal performance (Allwein et al. 2001). But for Aloi dataset with 1000 classes, we only report the results for all the decomposition strategies within $N_L = 1000$ due to its large class size.

[2] Note that coding design is independent from base learners. It is fair to fix the base learners for decomposition strategy comparison.

Table 6 Classification accuracy and standard deviation obtained by SVM classifiers for UCI datasets

Dataset	OVO	OVA	ECOC	M-SVM	Nary
Pendigits	**93.71 ± 2.03**	81.75 ± 1.07	87.64 ± 1.08	89.85 ± 1.65	93.22 ± 1.71
Vowel	**48.67 ± 2.63**	30.30 ± 1.42	34.28 ± 1.55	41.67 ± 1.42	39.96 ± 2.07
News20	68.36 ± 1.70	72.65 ± 2.24	70.61 ± 1.67	70.78 ± 0.85	**72.79 ± 1.08**
Letters	81.85 ± 1.26	66.93 ± 1.37	77.78 ± 1.26	80.16 ± 2.06	**82.27 ± 1.09**
Auslan	89.94 ± 2.23	41.12 ± 1.28	83.15 ± 0.84	90.06 ± 1.58	**91.57 ± 0.96**
Sector	85.06 ± 1.54	89.06 ± 1.35	88.01 ± 1.47	88.75 ± 1.55	**91.05 ± 1.32**
Aloi	91.49 ± 1.68	84.26 ± 2.53	85.48 ± 1.17	86.69 ± 1.02	**92.77 ± 1.86**
Glass	**61.84 ± 2.24**	55.00 ± 2.23	56.84 ± 1.12	58.84 ± 2.42	60.25 ± 1.60
Satimage	85.69 ± 2.57	83.11 ± 1.86	81.19 ± 1.62	83.28 ± 0.36	**86.50 ± 0.93**
Usps	94.30 ± 1.14	92.37 ± 1.65	90.76 ± 1.45	92.16 ± 1.27	**96.15 ± 2.47**
Segment	92.30 ± 1.46	92.00 ± 1.89	87.80 ± 1.61	89.96 ± 1.49	**93.60 ± 1.53**

The best performance on each dataset is marked in bold

6.2.1 Comparison to state-of-the-art baselines with SVM classifiers

The classification accuracy of different coding schemes as well as proposed N-ary coding with SVM classifiers are presented in Table 6. We observe that OVO has the best and most stable performance on most datasets of all the encoding schemes except for N-ary coding. This is because all the information between any two classes is used during classification and the OVO coding strategy has no redundancy among different base classifiers. However, it sacrifices efficiency for better performance. It is very expensive for both training and testing when there are many classes in the datasets such as the Auslan, Sector and Aloi. Especially, for Aloi with 1000 classes, it is often not viable to calculate the entire OVO classifications in the real-world application as it would require 499 500 base learners in the pool of possible combinations for training and testing. The performance of OVA is unstable. For the datasets News20 and Sector, OVA even significantly outperforms OVO. However, the performances of OVA on the datasets Vowel, Letters, and Glass are much worse than other encoding schemes. Note that ECOCONE is initialized with OVA. We observe that M-SVM achieves better results than random binary decomposition because it considers relationship among classes. However, the training complexity of M-SVM is very high. In contrast to M-SVM, random binary decomposition is ensemble of binary classifiers, which can be parallelized due to independences of base tasks. N-ary decomposition combines the advantages of both M-SVM and ensemble to achieve better performance.

6.2.2 Comparison to state-of-the-art baselines with decision tree classifiers

Next, we compare N-ary decomposition with other state-of-the-art coding schemes using binary decision tree classifiers CART (Breiman et al. 1984) as well as its multi-class extension M-CART. We implement it with the CART toolbox with a default setting and the results are reported in Table 7. We observe that binary decision tree classifiers with traditional decomposition strategies are worse than the direct multi-class extension of the decision tree. The decision tree classifiers show better performances than SVM on the Pendigits, Vowel, and Letters datasets. However, it shows very poor performances on high dimensional datasets such as News20 and Sector. This is due to the fact that high-dimensional features

Table 7 Classification accuracy and standard deviation obtained by CART classifiers for UCI datasets

Dataset	OVO	OVA	ECOC	M-CART	Nary
Pendigits	93.84 ± 2.33	78.12 ± 1.24	83.54 ± 1.30	87.64 ± 1.12	**95.84 ± 1.08**
Vowel	44.45 ± 1.74	33.57 ± 2.31	43.65 ± 2.35	45.45 ± 2.25	**48.50 ± 1.20**
News20	50.60 ± 1.17	45.23 ± 1.15	51.29 ± 1.26	50.83 ± 1.37	**53.71 ± 1.70**
Letters	81.56 ± 1.30	74.69 ± 1.50	89.75 ± 1.55	77.35 ± 1.26	**92.15 ± 1.92**
Auslan	79.84 ± 2.23	72.86 ± 2.04	83.15 ± 2.84	78.89 ± 1.18	**85.17 ± 1.26**
Sector	39.49 ± 1.33	41.89 ± 1.26	43.60 ± 1.17	45.89 ± 2.15	**47.05 ± 1.27**
Aloi	89.26 ± 1.49	72.10 ± 2.60	79.41 ± 1.08	73.00 ± 2.07	**95.13 ± 1.89**
Glass	52.84 ± 1.15	50.12 ± 1.24	54.65 ± 1.35	**64.00 ± 2.12**	56.00 ± 1.14
Satimage	85.70 ± 1.27	84.15 ± 1.08	85.86 ± 2.75	83.47 ± 2.36	**86.47 ± 2.23**
Usps	90.94 ± 2.16	80.89 ± 1.47	91.95 ± 1.57	83.54 ± 1.16	**92.77 ± 1.15**
Segment	93.68 ± 1.25	86.45 ± 2.22	96.44 ± 2.52	92.70 ± 1.34	**97.10 ± 1.28**

The best performance on each dataset is marked in bold

often lead to complex tree structure construction. Nevertheless, N-ary decomposition still can significantly improve the performance on either traditional coding schemes with binary decision tree learner as well as the multi-class decision tree.

In summary, our proposed N-ary decomposition is superior to traditional decomposition schemes and direct multi-class algorithms on most tasks, and provides a flexible strategy to decompose many classes into many smaller multi-class problems, each of which can be independently solved by either M-SVM or M-CART in parallelization.

6.2.3 Discussion on many class situation

From the experiments results, we observe that the N-ary decomposition shows significant improvement on the Aloi dataset with 1000 classes over other existing coding schemes as well as direct multi-class classification algorithms, especially decision tree classifiers. For the binary or ternary codes, it is highly possible to assign the same codes to different classes. From the experimental results, we observe the minimum distance ρ for binary and ternary coding are small or even tends to be 0. In other words, the existing coding cannot help the classification algorithms to differentiate some classes. In contrast, N-ary with $N_L = 1000$ and $N = 5$, the minimum distance ρ is 741. Thus, it creates codes with larger margins for different classes, which explains the superior On the other hand, the direct multi-class algorithms cannot work well when the class size is large. Furthermore, the computation cost for direct multi-class algorithms is in $O(N_C^3)$. When the class size N_C is large, the algorithms are expensive to train. On the contrary, random binary codes can be easily parallelized due to the independence among the subproblems.

6.2.4 Discussion on performance on each individual class

To understand the performances of different codes for each individual class, we show the confusion matrix on the Pendigits dataset in Fig. 6. First, we observe that binary code (i.e., OVA) has very poor performances on some classes in terms of recall or precision. For example, recall on class 2, 6 and precision on class 10 are below 50%. It can be explained by that as illustrated in Fig. 1b, binary codes may lead to nonseparable cases. Nevertheless, it achieves

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	Pre
1	90.2	0	0	0	0	0.1	0	0	0.7	0.3	88.9
2	0	96.3	0.1	0	0	0	0	0	0	0.1	94.8
3	0	0.5	89.8	0.1	0	0	0	0	0.3	0.3	89.2
4	0	0.1	0	92.1	0	0	0	0	0	0	98.5
5	0.1	0.1	0	0	89.8	0	0.3	0	0	0.1	94.7
6	0	0.3	0	0	0	95.2	0	0	0.4	0	86.1
7	0	0	0	0	0	0	90.7	0	0.1	0	98.7
8	0.2	1.1	0.1	0.2	0.1	0.9	0	89.8	0.7	0	75.9
9	0.2	0	0	0	0	0.5	0	0.1	93.1	0	89.9
10	0.1	4.6	0	1.4	0.1	3.2	0	0.1	0.5	91.2	46.9
Rec	94.4	35.4	97.8	82.1	98.2	49.9	96.4	98.6	71.8	91.8	81.8

(a) Binary Code.

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	Pre
1	90.2	0	0	0	0	0.2	0	0	0.2	0.3	92.9
2	0	90.7	0.3	0	0	0.1	0	0.2	0	0.2	90.5
3	0	0.1	89.9	0	0	0	0	0	0	0	98.3
4	0	0.6	0	90.6	0	0	0	0	0	0.3	90.7
5	0.3	0.1	0	0	89.7	0	0	0	0	0.1	94.6
6	0	0.1	0	0	0	91.4	0	0	0.3	0.2	92.9
7	0	0	0	0	0	0	90.4	0	0	0	99.4
8	0	0.1	0	0.1	0	0	0	89.9	0.3	0	95.4
9	0.2	0	0	0	0	0.4	0	0	91.8	0	92.1
10	0	0.1	0	0.1	0	0.2	0	0.1	0.5	91.7	89.8
Rec	94	89.6	97.4	97.9	99.4	89.6	99.6	96.9	85.4	85.6	93.7

(b) Ternary Code.

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	Pre
1	89.9	0	0.2	0.1	0.1	0.1	0.1	0	0	0.3	92.9
2	0	90.5	0.2	0.1	0.1	0.1	0.1	0.1	0.2	0.1	90.2
3	0	0.3	90.4	0.1	0.1	0	0.1	0	0	0	94.7
4	0	0.3	0.1	91	0.1	0.1	0	0	0	0.3	91.1
5	0.1	0	0	0	90.3	0	0	0	0	0.1	97.2
6	0	0.1	0	0.1	0	90.9	0	0	0.2	0.2	92.8
7	0	0.1	0	0	0	0.1	90.8	0	0.1	0	95.9
8	0	0	0.1	0.1	0	0	0.1	89.9	0.5	0	92.1
9	0.1	0	0	0.1	0.1	0.1	0.1	0	91.7	0	94.8
10	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0.2	91.4	91.1
Rec	97.2	91.7	91.9	93.7	93.6	95	95.3	97.6	86.5	89.2	93.2

(c) N-ary Code.

Fig. 6 Confusion matrix on Pendigits: in confusion matrix, the entry in the ith row and jth column is the percentage of images from class i that are misidentified as class j. Average classification rates for individual classes are listed along the diagonal. The last column and last row are precision (Pre) and recall (Rec) respectively

best classification results on the class 2, 4, 6 and class 9. Compared to the binary code, ternary code (i.e., OVO) largely reduces the bias and improve precision and recall scores on most classes. What is more interesting, when the ternary code and N-ary decomposition achieves comparable overall performances, N-ary decomposition achieves smaller maximal errors. It may be benefited from simpler subtasks created by N-ary decomposition, as shown in Fig. 1d.

7 Conclusions

In this paper, we investigate whether one can relax binary decomposition to N-ary decomposition to achieve better multi-class classification performance. In particular, we present an N-ary decomposition strategy that decomposes the original multi-class problem into simpler multi-class subproblems. The advantages of such decomposition are as follows: (i) the ability to construct more discriminative codes and (ii) the flexibility for the user to select the best N for random decomposition-based classification. We derive a base classifier independent generalization error bound for the N-ary decomposition classification problem. We show empirically that the optimal N (based on classification performance) lies in [3, 10] with some tradeoff in computational cost. Experimental results on benchmark multi-class datasets show that the proposed decomposition achieves superior prediction performance over the state-of-the-art multi-class baselines. In the future, we will investigate a more efficient realization of N-ary decomposition to improve the prediction speed.

Acknowledgements Joey Tianyi Zhou is supported by Programmatic Grant No. A1687b0033 from the Singapore government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain). Ivor Tsang is supported by Australian Research Council grants DP180100106, LP150100671, and FT130100746.

References

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, *1*, 113–141.

Bengio, S., Weston, J., & Grangier, D. (2010). Label embedding trees for large multi-class tasks. In *NIPS* (pp. 163–171).

Beygelzimer, A., Langford, J., Lifshits, Y., Sorkin, G., & Strehl, A. (2009). Conditional probability tree estimation analysis and algorithms. In *UAI* (pp. 51–58).

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*., Statistics/probability series Belmont, California: Wadsworth Publishing Company.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27:1–27:27.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Crammer, K., & Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265–292.

Deng, J., Satheesh, S., Berg, A., & Fei-Fei, L. (2011). Fast and balanced: Efficient label tree learning for large scale object recognition. In *NIPS*.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *MCS* (pp. 1–15). Springer.

Dietterich, T. G., & Bakiri, G. (1991). Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *AAAI* (pp. 572–577). AAAI Press.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Fei, B., & Liu, J. (2006). Binary tree of SVM: A new fast multiclass training and classification algorithm. *IEEE Transactions on Neural Networks*, *17*(3), 696–704.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, *44*(8), 1761–1776.

Gao, T., & Koller, D. (2011). Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV* (pp. 2072–2079).

García-Pedrajas, N., & Ortiz-Boyer, D. (2011). An empirical study of binary classifier fusion methods for multiclass classification. *Information Fusion*, *12*(2), 111–130.

Hong, J.-H., Min, J.-K., Cho, U.-K., & Cho, S.-B. (2008). Fingerprint classification using one-vs-all support vector machines dynamically ordered with Naïve Bayes classifiers. *Pattern Recognition*, *41*(2), 662–671.

Jenssen, R., Kloft, M., Zien, A., Sonnenburg, S., & Müller, K.-R. (2012). A scatter-based prototype framework and multi-class extension of support vector machines. *PLoS ONE*, *7*(10), e42947.

Kittler, J., Ghaderi, R., Windeatt, T., & Matas, J. (2003). Face verification via error correcting output codes. *Image and Vision Computing*, *21*(13–14), 1163–1169.

Knerr, S., Personnaz, L., & Dreyfus, G. (1990). Single-layer learning revisited: A stepwise procedure for building and training a neural network. In F. F. Soulié & J. Hérault (Eds.), *Neurocomputing*. NATO ASI Series (Series F: Computer and Systems Sciences), Vol. 68. Berlin, Heidelberg: Springer.

Lauer, F., & Guermeur, Y. (2011). MSVMpack: A multi-class support vector machine package. *Journal of Machine Learning Research*, *12*, 2269–2272.

Liu, X.-Y., Li, Q.-Q., & Zhou, Z.-H. (2013). Learning imbalanced multi-class data with optimal dichotomy weights. In *IEEE 13th international conference on data mining (ICDM)* (pp. 478–487). IEEE.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Rocha, A., & Goldenstein, S. (2014). Multiclass from binary: Expanding one-vs-all, one-vs-one and ECOC-based approaches. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(2), 289–302.

Su, J., & Zhang, H. (2006). A fast decision tree learning algorithm. In *Proceedings of the 21st national conference on artificial intelligence, AAAI'06* (Vol. 1, pp. 500–505). AAAI Press.

Tsang, I. W., Kwok, J. T., & Cheung, P. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, *6*, 363–392.

Übeyli, E. D. (2007). Ecg beats classification using multiclass support vector machines with error correcting output codes. *Digital Signal Processing*, *17*(3), 675–684.

Wu, T.-F., Lin, C.-J., & Weng, R. C. (2004). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, *5*(Aug), 975–1005.

Yang, J.-B., & Tsang, I. W. (2011). Hierarchical maximum margin learning for multi-class classification. In *UAI*.

Yu, Z., Cai, D., & He, X. (2010). Error-correcting output hashing in fast similarity search. In *Proceedings of the second international conference on internet multimedia computing and service, ICIMCS '10* (pp. 7–10). New York, NY, USA: ACM.

Zhao, B., & Xing, E. P. (2013). Sparse output coding for large-scale visual recognition. In *CVPR* (pp. 3350–3357). IEEE.

Zhong, G., & Cheriet, M. (2013). Adaptive error-correcting output codes. In *IJCAI* (pp. 1932–1938).

Zhong, G., & Liu, C.-L. (2013). Error-correcting output codes based ensemble feature extraction. *Pattern Recognition*, *46*(4), 1091–1100.