LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy

Chen Lin, Wenqiang Chen, Cheng Qiu, Yunfeng Wu, Sridhar Krishnan, Quan Zou

Abstract

Selective ensemble is a learning paradigm that follows an “overproduce and choose” strategy, where a number of candidate classifiers are trained, and a set of several classifiers that are accurate and diverse are selected to solve a problem. In this paper, the hybrid approach called D3C is presented; this approach is a hybrid model of ensemble pruning that is based on k-means clustering and the framework of dynamic selection and circulating in combination with a sequential search method. Additionally, a multi-label D3C is derived from D3C through employing a problem transformation for multi-label classification. Empirical study shows that D3C exhibits competitive performance against other high-performance methods, and experiments in multi-label datasets verify the feasibility of multi-label D3C.

Keywords: Selective ensemble learning, Clustering, Dynamic selection, Circulating combination, Multi-label classification, Machine learning

1. Introduction

Recently, ensemble techniques have gained widespread interest among researchers in the machine learning community. Bagging [1] and boosting [2] methods for team member generation, and variants thereof, such as arcing [3], have been shown to be very successful. In most of the experiments, the ensemble algorithms were shown to be more effective than each independent classifier. In addition, the ensemble techniques have been shown to be very accurate classification techniques and have been employed for a broad range of applications and under a variety of scenarios, such as bioinformatics [4,5], software reuse [6], face recognition [7], and lung cancer cell identification [8].

Although the ensemble systems can provide considerable impact, there are drawbacks or weaknesses at the same time. Most of the previous work has either focused on combining the outputs of multiple trained networks or has only indirectly addressed the necessity for generating a good set of networks. However, in some real-life applications, the number of independent classifiers that are required to achieve a reasonable accuracy is enormously large and, hence, very space consuming [9].

Margineantu and Dietterich [10] observed that, in some domains, the boosting algorithm, although effective at generating a diverse set of classifiers, requires a large amount of memory for storing all of the independent classifiers, to lower the prediction error. More specifically, the AdaBoost algorithm [2] requires approximately 200 C4.5 trees [11] to achieve very good generalisation accuracy.

Should we ensemble all of the available classifiers? At present, most of the approaches ensemble all of the available classifiers for prediction. However, Zhou revealed that it was better to ensemble many classifiers instead of all of the classifiers at hand [12]. In [12], an approach called genetic algorithm-based selective ensemble (GASEN) was presented, which could select appropriate classifiers for composing an ensemble from a set of available classifiers, and a large empirical study showed that GASEN generated neural network ensembles that had far smaller sizes but stronger generalisation ability.

Why can selective ensemble learning generate stronger generalisation ability and save more time? Selective ensemble learning is a useful strategy for reducing the ensemble size by selecting only a fraction of the classifiers from the original ensemble. Additionally, the basic ability of selective ensemble learning is a reduction in the complexity of the classification. In addition, not only accuracy but also diversity is required from most of the combination methods for classifiers. Diversity among the members of a team of classifiers is deemed to be a key issue in classifier combination. Selective ensemble learning considers the diversity, obtains a more diverse subset of classifiers and performs better...
than the original ensemble most of the time. Selective ensemble learning can be viewed as searching optimal solutions to obtain satisfactory prediction accuracy. Many searching strategies have been proposed to find a near-optimal sub-ensemble, such as GASEN, which is mentioned above.

Selective ensemble is also a learning paradigm; it trains a number of base classifiers and selects some of them to ensemble. Generally speaking, selective ensemble techniques can be divided into four categories: Clustering based, Ranking based, Optimisation based and other methods. Lazarevic and Obradovic proposed a method that was based on the approach that applied k-means clustering [13] to an entire set of classifiers for the purpose of identifying the groups that had similar classifiers and then eliminated redundant classifiers that were in each cluster [9]. Giacinto and Roli presented an approach to the automatic design of multiple classifier systems based on hierarchical agglomerative clustering algorithms and gave an optimality proof [14]. In [15], Bryll utilised attribute bagging (AB), which is a technique for improving the accuracy and stability of classifier ensembles that are induced by using random subsets of features, and they demonstrated that ranking the attribute subsets by their classification accuracy and voting using only the best subsets further improved the resulting performance of the ensemble. Croux ordered different classifiers by increasing the error rate that resulted from the bootstrapped training set [16]. The methods based on optimisation mainly trained a number of independent classifiers first, then assigned random weights to those classifiers, and finally, employed optimisation algorithms to evolve the weights so that they could characterise, to some extent, the fitness of the classifiers in constituting an ensemble for the purpose of selecting a subset of classifiers, such as in GASEN.

There is no prior reason to assume that any of the four strategies in this question must be used. None of the aforementioned strategies can apply universally. Every method has its demerits. The instability of the methods that are based on clustering is considered to be a bad influence on the effect. The methods based on ranking waste much time when training the base classifiers and require a large amount of storage. For selective ensemble learning based on optimisation, it is not easy to find the global optimum using a heuristic search algorithm; as a result, most of them waste time on parameter optimisation.

In machine learning, the hybrid approach has been an active research area for improving classification or prediction performance over a single learning approach. In general, hybridisation is based on combining two different machine learning techniques. The rationale behind the hybrid model is that a hybrid classification model can be composed of one unsupervised learner to pre-process the training data and one supervised learner to learn the clustering result. In [17], Soto et al. proposed to apply dynamical (instance-based) pruning to a subset of classifiers that were selected using standard ensemble pruning techniques. The pruned sub-ensemble was built by first modifying the order in which the classifiers were aggregated in the ensemble and then selecting the first classifiers in the ordered sequence. The double pruning algorithm was used to reduce the storage requirements, speed up the classification process and improve the performance of the parallel ensembles. Most of the previous work focused on the improvement of selective ensemble techniques, while little attention has been devoted to the development of the hybrid approach of selective ensemble techniques. Motivated by the performance of the hybrid model, an approach that integrates two types of selective ensemble techniques is proposed and called dynamic selection and circulating combination-based clustering (D3C) in this paper, and we give an effectiveness proof. In addition, the proposed method is transformed to handle multi-label classification data.

Multi-label learning tasks are ubiquitous in real-world problems; multi-label learning problems have attracted a substantial amount of attention from machine learning researchers, and multi-label classification methods are increasingly required by modern applications, such as protein function classification, music categorisation, and semantic scene classification [18]. The ensemble learning methods can bring robustness to multi-label classification. Bagging, boosting and random forests are popular ensemble learning methods for classification problems. The main motivation is the production of multi-label models that can be understood by humans. To reduce the information redundancy within multi-label learning, a model-shared subspace boosting algorithm was developed that automatically finds shared subspace models, where each model was learned from the random feature subspace and bootstrap data and combined a number of base models through multiple labels [19]. One famous approach for solving the difficulty of the concept ambiguity that is encountered in text categorisation, where each document might belong to several labels simultaneously, was called BOOSTEXTER, which was proposed by Schapire and Singer [20] and was, in fact, extended from the popular ensemble learning method ADABOOST [21]. AdaBoost.MH and AdaBoost.MR are also two extensions of AdaBoost for multi-label data. In 2003, Comité et al. [22] extended alternating decision tree to handle multi-label data, where the AdaBoost.MH was employed to train a multi-label alternating decision tree. Ensembles of pruned sets (EPS) was an ensemble method that used pruning to reduce the computational complexity of the label power-set (LP) method (by up to an order of magnitude in practice) and that used a label-set subsampling method to preserve the high predictive performance. Empirical evidence indicated that this method was competitive in terms of its efficiency [23]. It is worth mentioning that there are few approaches that use selective ensemble methods to accomplish a multi-label classification task. Therefore, we design D3C as an approach for multi-label classification, and we detail the comparative experimental results.

The remainder of this paper is organised as follows. In Section 2, ensemble pruning based on k-means clustering, dynamic selection and circulating combination are introduced; D3C is proposed and D3C for multi-label classification is described. In Section 3, the optimisation methods are provided to improve the performance of the proposed model. In Section 4, a large empirical study is reported for analysing the competitive performance of the hybrid model. Finally, in Section 5, the contributions of our work are concluded, and several interesting issues for future work are indicated.

2. The design of the methodology

This section introduces the process of the hybrid model proposed in this study. The hybrid model employs two types of selective ensemble techniques, which are a combination of the ensemble pruning based on k-means clustering and dynamic selection and circulating combination. The hybrid model follows the so-called "overproduce and choose" strategy [14]. The strategy allows one to exploit effectively all of the available methods for the creation of a set of "candidate" classifiers. Different classifier types that make different errors are used in the overproduction phase to generate the initial ensemble of candidate members. The choice phase based on k-means clustering is aimed at eliminating redundant classifiers. The subsequent choice phase is the framework of dynamic selection and circulating combination, which is aimed at making up the classifiers that have a high degree of diversity and improving the ensemble performance without the exhaustive enumeration of all of the possible subsets. Fig. 1 shows
the flowchart of the proposed model. Each part of the hybrid approach is described in the following subsections.

2.1. Ensemble pruning based on k-means clustering

In this section, ensemble pruning that is based on k-means clustering is presented. This procedure acts as the pre-process for constituent classifiers. There are two reasons for pruning the ensemble: first, the performance of an ensemble that is composed of many classifiers could be better than any of the classifiers [12] and second, the ensemble methods require that a large number of memories be stored in all of the base classifiers, and the ensemble pruning can greatly decrease the memory that is used in real-world applications [24]. The objective of the approach is to determine the classifiers that should be excluded from ensembles and to keep the prediction performance. The basic idea of this method is to use an unsupervised approach.

Let $\Omega = \{1, \ldots, c\}$ be a set of class labels; let $\mathbf{x} \in \mathbb{R}^{n}$ be a vector that has $n$ features to be labelled in $\Omega$; and let $S = \{ (\mathbf{x}_1, o_1), \ldots, (\mathbf{x}_m, o_m)\}$, where $o_j \in \Omega$ is a training set. To create classifiers that produce different errors, an initial set of classifiers that vary the type $H = \{ h_1, \ldots, h_T\}$ is given. Denote by $h_i(\mathbf{x}, o_j)$ the output that the classifier $h_i$ makes for the instance $(\mathbf{x}_j, o_j) \in S$. The classifier output represents a correct/incorrect decision. Moreover, the output $h_i(\mathbf{x}_j, o_j)$ is 1 if $(\mathbf{x}_j, o_j)$ is recognised correctly by $h_i$ and 0 otherwise. This arrangement is an “oracle” type of output because it assumes that we know the correct label $o_j$ of $\mathbf{x}_j$. It is apparent that the output that the classifiers $h_i$ make for the entire training set $S$ can be represented as a vector $\mathbf{y}_i$. The vector $\mathbf{y}_i$ contains $m$ correct/incorrect decisions, one for each of the data examples from the training set $S$ to express the correctness of the classification.

The standard $k$-means algorithm is applied to partition the set of classifiers into subsets that contain similar classifiers. Next, suppose that there are $m$ instances, and the actual output of the $i$-th classifier $h_i$ and the $j$-th classifier $h_j$ on those instances satisfies that $\mathbf{y}_i = \{d_{i1}, \ldots, d_{im}\}^T$ and $\mathbf{y}_j = \{d_{j1}, \ldots, d_{jm}\}^T$, respectively, where $d_{ij}$ denotes the output of the $a$-th classifier on the $b$-th instance. It is obvious that if the actual output of $h_i$ on the $k$-th instance is 0, and it is the same as the actual output of $h_j$, then $d_{ij}d_{jk} = 1$; otherwise, $d_{ij}d_{jk} = 0$. The possibility of misclassifying the intersection of $h_i$ and $h_j$ is

$$
\text{Prob}(h_i \text{fails}, h_j \text{fails}) = \frac{1}{m} \sum_{k=1}^{m} d_{ik}d_{jk}
$$

Additionally, $D(h_i, h_j)$ denotes the distance between $h_i$ and $h_j$ and is defined as

$$
D(h_i, h_j) = 1 - \text{Prob}(h_i \text{fails}, h_j \text{fails})
$$

Here, the set of classifiers $H = \{ h_1, \ldots, h_T\}$ are partitioned into $k$ clusters by finding $k$ points $\{M_j\}_{j=1}^{k}$ such that $\sum_{j=1}^{k} \min D(h_i, M_j)$ is minimised, where the points $\{M_j\}_{j=1}^{k}$ are known as cluster centroids or cluster means.

To identify the optimal number of clusters, the number of clusters $k$ is increased gradually until $\min \sum_{j=1}^{k} \min D(h_i, M_j)$ starts to deteriorate. When the optimal number $k$ of the clusters is obtained, according to the assumption in [9], the agreement among the classifiers from the same cluster is large. Therefore, the majority of the classifiers can be eliminated according to their accuracy on the validation.

2.2. The diversity measure for dynamic selection and circulating combination

Diversity has been recognised as a very important characteristic in classifier combination. Additionally, many statistics can measure the diversity among the classifiers, such as the Q statistics, the correlation and the entropy of the votes. These measures of diversity can be grouped into two categories: (i) pairwise diversity measures, and (ii) non-pairwise diversity measures. The first group of the measures of diversity considers only two classifiers. The second group of the measures of diversity works mostly on the whole group of classifiers. Although a substantial number of the measures of diversity is available for the framework of dynamic selection and circulating combination, this paper concerns only the measurement of the interrater agreement $\kappa$ [25].

The interrater agreement $\kappa$ has links to the intraclass correlation coefficient, and its non-pairwise diversity measure is
In addition, majority voting (MV), which only works with nominal class labels, is the most common selection criterion and is a measure for the majority voting error rate. Here, how majority voting works is described as follows. Assuming that \( t \) classifiers vote for the instance \((\vec{x}_j, o_j)\), each classifier classifies the instance and outputs the probability distribution for the instance. Suppose that there is a set of class labels \( \Omega = \{1, ..., c\} \) and the probability distribution of the instance is a vector \( \bar{p} = (p_1, ..., p_c) \), where \( p_i \) denotes the probability of the \( j \)-th label. Then, the label with the highest probability obtains a vote. If multiple labels have the same probability, then all of those labels receive a vote. Once all of the classifiers cast the vote, the label with the most votes is selected as the label for the test instance. If multiple labels have the same number of votes, then one of those labels will be selected at random. The MVE can be formulated as

\[
MVE = 1 - \frac{1}{m} \sum_{j=1}^{m} MV(\vec{x}_j, o_j)
\]

where \( MV(\vec{x}_j, o_j) \) represents the correct/incorrect decision of the majority voting and is an oracle type of output.

The sequential search method is embedded into the framework. This method can be seen as a multilayer sequential search method. In addition, the multilayer classifier subset selection can take full advantage of each base classifier and preserve more useful information than a selective ensemble that obtains only one optimal selection as an ensemble result. The whole framework is composed of many layers. In this framework, the selective ensemble method of one layer can be seen as a multimodal optimisation problem; each layer will generate an oracle output, and the search method of the current layer is based on the oracle output of the previous layer. In this situation, each classifier will have the opportunity to participate in one ensemble. A schematic diagram of a multilayer search method is shown in Fig. 2.

At each step of the sequential search method, the algorithm should decide whether to update the selected subset. If a small range change occurs on the interrater agreement \( \kappa \) and MVE becomes

\[
\kappa = 1 - \frac{1}{p(1-p)} D_{S_{xy}}
\]

where \( p \) is described as the average accuracy of \( t \) classifiers for the training set \( S = \{(\vec{x}_1, o_1), ..., (\vec{x}_m, o_m)\} \) and is expressed as follows:

\[
p = \frac{1}{m} \sum_{j=1}^{m} \sum_{t=1}^{t} h(\vec{x}_j, o_j)
\]

where \( h(\vec{x}_j, o_j) \) represents a correct/incorrect decision and is the "oracle" type of output mentioned above.

The averaged disagreement measure \( D_{S_{xy}} \) is

\[
D_{S_{xy}} = \frac{1}{t(2-x)} \sum_{i=1}^{t} \sum_{j=1}^{t} D_{S_{ij}}
\]

where \( D_{S_{ij}} \) is the ratio between the number of observations on which one classifier is correct and the other is incorrect to the total number of observations and is defined by

\[
D_{S_{ij}} = \frac{N_{ij}^{01} + N_{ij}^{10}}{N_{ij}^{11} + N_{ij}^{10} + N_{ij}^{01} + N_{ij}^{00}}
\]

where \( N_{ij}^{ab} \) shows the relationship between a pair of classifiers \( h_i \) and \( h_j \) on the instance \((\vec{x}_j, o_j)\) and is defined by

\[
N_{ij}^{ab} = \begin{cases} 
N_{ij}^{00} & \text{if } h_i(\vec{x}_j, o_j) = 0 \text{ and } h_j(\vec{x}_j, o_j) = 0 \\
N_{ij}^{01} & \text{if } h_i(\vec{x}_j, o_j) = 0 \text{ and } h_j(\vec{x}_j, o_j) = 1 \\
N_{ij}^{10} & \text{if } h_i(\vec{x}_j, o_j) = 1 \text{ and } h_j(\vec{x}_j, o_j) = 0 \\
N_{ij}^{11} & \text{if } h_i(\vec{x}_j, o_j) = 1 \text{ and } h_j(\vec{x}_j, o_j) = 1 
\end{cases}
\]

The interrater agreement \( \kappa \) is considered to be a measure of the similarity; in other words, the higher the value is, the less diversity. Furthermore, the interrater agreement \( \kappa \) not only focuses on the diversity of the whole system but also considers the prediction performance of the entire system.

2.3. Dynamic selection and circulating combination

In this section, we introduce dynamic selection and circulating combination. This approach is presented in terms of the adaptive framework of selective ensemble learning. In [26], a framework with selective ensemble learning based on a complementarity factor was proposed, and the experimental results of handwritten digit recognition showed that it was more flexible and efficient compared to other classifier selection methods. In this paper, the framework with the sequential methods is proposed, and the sequential methods include sequential forward search (SFS) and sequential backward search (SBS). Sequential search methods start from an empty set or the set of all candidates as the initial selected subset. In the beginning, the performance in the overproduction phase is used to rank all of the classifiers that were obtained by ensemble pruning based on \( k \)-means clustering. Additionally, SFS adds the most accurate classifier and SBS deletes the least accurate classifier from the selected subset iteratively, with the aim of improving the final performance. At each step, only one or a small number of classifiers is added or deleted; as a result, the complexity of the search is not high. In addition, this approach is efficient even for large-scale problems [27].

The selection criterion is very important for sequential search. Furthermore, the interrater agreement \( \kappa \) mentioned above can be the selection criterion. However, there is no explicit theory that shows a measure of diversity that is as straightforward as accuracy. In addition, majority voting (MV), which only works with nominal classes, is utilised as the combination rule. Therefore, the majority voting error (MVE) in conjunction with the interrater agreement \( \kappa \) acts as a selection criterion.
smaller, then the algorithm makes all of the changes visible to the next step. Furthermore, the algorithm implements a rollback to the previous state and then moves on to the next step. To control the procedure dynamically and make the algorithm more flexible, the expected accuracy $\lambda$, interval $\Delta\lambda$, threshold $N_{\text{threshold}}$, range $\alpha$, and selected subset $V = \Phi$ for SFS ($V = H'$ for SBS);

Output: selected subset $V, MVE(V)$

Procedure:
1. begin
2. while $\lambda \geq 0$ do
3. if $1 - MVE(V) > \lambda$
4. return $V$;
5. end if
6. if Num($V$) > $N_{\text{threshold}}$
7. $V_{\text{suboptimal}} = V$;
8. end if
9. for $h'_1$ to $h'_k$ do
10. change $V$ to $V'$;
11. if $|\kappa(V) - \kappa(V')| < \alpha$ $\&$ $MVE(V) > MVE(V')$
12. commit the changes;
13. if $1 - MVE(V') > \lambda$
14. return $V'$;
15. else
16. if $\kappa(V') < \kappa(V)$
17. $V_{\text{suboptimal}} = V'$;
18. end if
19. end if
20. else
21. rollback to $V$;
22. end if
23. end for
24. $\lambda = \lambda - \Delta\lambda$;
25. end while
26. End

Fig. 3. The framework of dynamic selection and circulating combination.

2.4. D3C for multi-label classification

As stated in the above sections, although there have been selective ensemble learning methods that optimise the performance of the single-label classification in terms of general accuracy, these methods are not suitable for multi-label classification problems. In this section, the proposed algorithm called D3C, which is a method for multi-label classification, is presented.

Traditional two-class and multi-class problems can both be cast into multi-label problems by restricting each instance to having only one label. However, the generality of the multi-label problem makes it more difficult. Currently, we can group existing methods for multi-label classification into two main categories: (i) problem transformation methods and (ii) algorithm adaption methods. The methods of the first group are defined as those methods that transform the multi-label classification problem into either one or more single-label classification problems. There exist several transformations that can be used to convert a multi-label dataset
into multiple single-label datasets. Binary relevance (BR) is a popular problem transformation method that learns a binary classifier for each different label. LP is a simple but effective problem transformation method that works as follows: it considers each unique set of labels that exists in a multi-label training set to be one of the classes of a new single classification task. Given a new instance, a single-label classifier of LP outputs the most probable class, which actually represents a set of labels. The methods of the second group extend specific learning algorithms to handle multi-label data directly [28]. This paper focuses only on D3C with the binary relevance multi-label learning method. The following paragraphs introduce several new notations and the rationale behind D3C with binary relevance.

In multi-label classification, each instance in the training set can be associated with multiple labels, and the task is concerned with learning a model that outputs a bipartition of the set of labels into relevant and irrelevant with respect to a query instance. Let $L = \{l_1, \ldots, l_Q\}$ be the finite set of labels, let $\mathcal{X}_j \in \mathbb{R}^n$ still be the $j$-th instance with $n$ features and let the $Q$-dimensional vector $\vec{T}_i \in \mathbb{R}^Q$ be the associated labels of the instance $\mathcal{X}_i$. The associated label set $\vec{T}_i = [t_{i1}, \ldots, t_{iQ}]^T$, $t_{ij}$ is specified as the nominal attributes with values “0” and “1” and indicates the absence and presence of the $j$-th label of the $i$-th instance. D3C with binary relevance decomposes multi-label learning into multiple independent selective ensembles. The algorithm transforms the original multi-label dataset $\mathcal{S}_{\text{multi-label}} = \{(\mathcal{X}_1, \vec{T}_1), \ldots, (\mathcal{X}_m, \vec{T}_m)\}$ into multiple single-label data sets $\mathcal{S} = \{(\mathcal{X}_1, l_{11}), \ldots, (\mathcal{X}_m, l_{m1})\}, \ldots, \{(\mathcal{X}_1, l_{1Q}), \ldots, (\mathcal{X}_m, l_{mQ})\}$ that contain all examples of the original dataset. Subsequently, D3C generates a different subset of classifiers $H_j$ for a different dataset $\{(\mathcal{X}_1, l_{1j}), \ldots, (\mathcal{X}_m, l_{mj})\}$ associated with each different label. For the classification of a new instance $\mathcal{X}_i$, this method outputs a set of labels, the union of the labels that are output by the Q subset of classifiers

$$\vec{T}_i = [H_{i1}(\mathcal{X}_i), \ldots, H_{iQ}(\mathcal{X}_i)]^T$$

where $H_{ij}(\mathcal{X}_i)$ represents the $j$-th label of the instance $\mathcal{X}_i$, which is produced by the $j$-th subset of classifiers $H_j$. Fig. 4 shows the procedure of D3C with binary relevance for multi-label classification.

Multi-label classification requires different measures compared with those used in the case of traditional single-label classification. Some of the measures that evaluate bipartitions are calculated based on the average differences of the actual and the predicted sets of labels over all examples of the evaluation data set. Others decompose the evaluation process into separate evaluations for
each label, which they subsequently average over all of the labels. The former and the latter are called example-based and label-based evaluation measures, respectively [29]. Additionally, it is convenient to define the evaluation measures by the above notations.

The first evaluation criterion to be introduced is an example-based evaluation measure, and the so-called Hamming loss is defined as

\[
\text{Hamming loss} (H_1, ..., H_q) = \frac{1}{m} \sum_{l=1}^{m} \frac{|\hat{I}_l \Delta \bar{I}_l|}{Q}
\]

where \(\Delta\) denotes the symmetric difference of two sets, which is the set of elements that are in either of the sets and not in their intersection. Additionally, \(|\hat{I}_l \Delta \bar{I}_l|\) is the number of elements in the symmetric difference between \(\hat{I}_l\) and \(\bar{I}_l\). The smaller value of the Hamming loss implies a better performance for the multi-label classifier.

The label-based evaluation measure is the micro-average F-measure using the measures for binary evaluation to evaluate the average performance of multi-label classification. It is defined as follows:

\[
F\text{- measure}_{\text{micro}} = \frac{2 \sum_{i=1}^{Q} tp_i}{\sum_{i=1}^{Q} \left( fp_i + \sum_{i=1}^{Q} fp_i \right) + 2 \sum_{i=1}^{Q} \left( fn_i + \sum_{i=1}^{Q} fn_i \right)}
\]

where \(tp_i, fp_i,\) and \(fn_i\) are the number of true positives, false positives and false negatives after binary evaluation for the label \(\hat{I}_i = [H_1(\hat{x}_1), ..., H_q(\hat{x}_m)]^T\). The larger value of the micro-average F-measure implies a better performance of the multi-label classifier.

The Hamming loss and micro-average F-measure are popular and indicative. They evaluate the performance of each multi-label learning algorithm in the remainder of this paper.

### 3. The optimisation methods

In the aforementioned sections, it is known that D3C follows the “overproduce and choose” strategy. This strategy is based on the creation of a large set of classifiers in the overproduce phase. To achieve a good initial set of candidate members, the error diversity and accuracy are two fundamental needs that should be satisfied. Varying the classifier type is one of the best ways of creating error-diverse classifiers. Accordingly, different classifier types used in the overproduce phase are aimed at making different errors. One issue in creating accurate classifiers is that the algorithms require parameter tuning to achieve an optimal or near optimal performance. The parameter tuning for a classifier is a rather tedious process. To address this issue, cross validation is used as the evaluation component of an exhaustive search to automate the process and generate the effective parameters. Although cross validation is an example that involves automatic setting of the parameters in a small dataset, it is time-consuming in a large dataset due to the key problem of the substantial computational cost of the training of different classifier types. The parallel optimisation is designed so that the problem can be solved.

#### 3.1. Cross validation for parameter tuning in D3C

There are some parameters for the base classifiers of D3C that can be set in the training process, such as the penalty parameter and the kernel parameters of the support vector machine. Parameter tuning is a widely studied problem in statistics. The goal is to identify a good parameter combination so that the classifier can accurately predict unknown data. Cross validation is one way of automating parameter tuning to obtain the best values for the parameters. The strategy behind cross validation is to separate the dataset into two parts, one of which is considered to be unknown. The prediction accuracy obtained from the “unknown” set more precisely reflects the performance on classifying an independent dataset. In k-fold cross-validation, we first divide the training set into k subsets of equal size. In a sequential fashion, one subset is tested using the classifier trained on the remaining k – 1 subsets. Thus, each instance of the whole training set is predicted once, which makes the cross-validation accuracy be the percentage of data that is correctly classified [30]. Cross validation is straightforward but appears naive. It can optimise over an arbitrary number of parameters, but we usually focus on only the parameters that have influence over the classifiers. It should be noted that cross validation has one drawback, which is that it cannot optimise over nested parameters. Hence, it ignores the obvious explosion of possible parameter combinations. Fig. 5 describes the method for using cross validation to search the optimal parameter combinations of a decision tree. For convenience, Fig. 5 shows only the process of a decision tree. However, note that every classifier should optimise the parameters by cross validation.

#### 3.2. The parallel optimisation for independent training of D3C

The training of candidate classifiers is an important process for D3C. However, training a large set of the classifiers sequentially on a large dataset is still a bottleneck, as is cross validation for parameter tuning. Therefore, it is important to develop parallel optimisation for the process when solving certain large-scale classification problems. In this section, we propose efficient solutions to the above problem.

Given a dataset \(S = ((\hat{x}_1, \omega_1), ..., (\hat{x}_m, \omega_m))\) and an initial set of classifiers that vary the type \(H = \{h_1, ..., h_T\}\), training the set of classifiers on the dataset is to obtain the matrix \(M_{\text{oracle}}\), which comprises the oracle output and is written as follows:

\[
M_{\text{oracle}} = \begin{bmatrix}
    h_1(\hat{x}_1, \omega_1) & \ldots & h_T(\hat{x}_1, \omega_m) \\
    \vdots & \ddots & \vdots \\
    h_1(\hat{x}_m, \omega_1) & \ldots & h_T(\hat{x}_m, \omega_m)
\end{bmatrix}
\]
The parallelisation can allow a process to overlap computational operations with one or more training sessions. First, the parallelisation can be realised by a multi-thread technique. The parallel training is efficient due to two aspects. First, the parallelisation can allow a process to overlap computational operations with one or more training sessions. Second, different parameter tunings can be conducted with no effort. Additionally, computations on different columns are independent.

The parallel optimisation diagram is depicted in Fig. 6. We have built $T$ classifiers in the first row, and parameter optimisations from $PC(i, 1)$ to $PC(i, n)$ on the $i$-th column are independent. Additionally, computations on different columns are independent. As a result, this architecture of the computation is suitable for parallel optimisation.

### 4. Experimental analysis

To know how well D3C works, a large empirical study is performed. This section presents the information on the datasets and, then, describes the experimental methodology, and finally, it reports on the experimental results.

#### 4.1. Datasets

Our experiments were conducted on eight datasets, which had different sizes. Among those datasets, four are used for single-label classification while the remaining are used for multi-label classification. The information on all of the datasets is tabulated in Table 1.

We consider three real and one artificial dataset for single-label classification. Three real datasets that represent two-class problems are taken from the UCI machine learning repository [31], which has been extensively used in testing the performance of diverse types of classifiers. Liver and Ionosphere have been used by Zhou in comparing C4.5 and neural ensemble-based C4.5 (NeC4.5) [32]. Pima-diabetes has been chosen to study the relationship between the diversity of ensembles in bagging and boosting and the efficiency of these combining techniques when applied to the nearest mean classifier [33]. One artificial dataset, called Homburg, is chosen for multi-class classification and is a freely available benchmark dataset for audio classification and clustering, which comprises 10-s samples of 1886 songs obtained from the Garageband site [34]. The songs are classified into nine genres: Funk/Soul, Alternative, Folk/Country, Rock, Rap/HipHop, Pop, Jazz, Electronic, and Blues, and the number of songs in each genre vary. Fig. 7 provides an overview.

For multi-label classification, we experiment with three datasets from three different applications domains: emotion detection, proteomics and semantic annotation. The emotions dataset contains 593 annotated songs, which are labelled with six main emotional clusters from the Tellegen–Watson–Clark model of mood [35,36]. The yeast dataset is a biological dataset that is concerned with gene function classification. Additionally, each gene is described by the concatenation of micro-array expression data and a phylogenetic profile and is associated with a set of functional classes from the comprehensive Yeast Genome Database of the Munich Information Center for protein sequences [37]. The scene dataset relates to a scene classification problem. A photograph can belong to more than one conceptual class, such as sunset and beach at the same time. The described datasets are widely used as benchmark datasets for the evaluation of multi-label classification algorithms. They can be obtained from the knowledge discovery and machine learning website [18]. The details of the datasets, such as the number of examples, the number of attributes, and the number of labels, are given in Table 1.

#### 4.2. Experimental methodology

For single-label classification, D3C is comparable against other methods, such as ensemble pruning based on $k$-means clustering, SFS, and SBS, and the experiment is conducted in the datasets of liver, Ionosphere and pima-diabetes, to show the effectiveness of the proposed model. We implement the D3C and other methods in Java, within the framework of the WEKA library of machine learning.

### Table 1

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attribute</th>
<th>Class</th>
<th>Label</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liver</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>345</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>35</td>
<td>2</td>
<td>1</td>
<td>351</td>
</tr>
<tr>
<td>Pima-diabetes</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>768</td>
</tr>
<tr>
<td>Homburg</td>
<td>50</td>
<td>8</td>
<td>1</td>
<td>1886</td>
</tr>
<tr>
<td>Emotions</td>
<td>78</td>
<td>–</td>
<td>6</td>
<td>593</td>
</tr>
<tr>
<td>Yeast</td>
<td>117</td>
<td>–</td>
<td>14</td>
<td>2417</td>
</tr>
<tr>
<td>Scene</td>
<td>300</td>
<td>–</td>
<td>6</td>
<td>2407</td>
</tr>
</tbody>
</table>
learning algorithms, and we make the software publicly available at the following URL (http://datamining.xmu.edu.cn/main/~qc/libD3C/index.html). To know how well the compared approaches work, we also test the performance of the following classifier learning algorithms: SMO [38] and Random Forests [39]. Additionally, we use the implementations of SMO and Random Forests within the WEKA library. With regard to the performance of D3C with parameter tuning, we compare it against the original D3C and other popular methods for music genre classification in the literature [40], using the Homburg dataset. For a performance evaluation, 10-fold cross validation is performed in each dataset when transformed to the ARFF file format of the WEKA library, and the average value of a 10 times 10-fold cross validation evaluates the predictive performance of each method. Cross validation is helpful for drastically cutting the risk of producing a poor model. It is noted that cross validation is utilised for the evaluation step and the parameter tuning, and it is necessary when avoiding confusion in the role of the cross validation.

For multi-label classification, multi-label D3C is compared against the aforementioned methods BR and LP, which are the simple baseline methods, as well as against the multi-label lazy learning approach called ML-KNN, which is derived from the traditional K-nearest neighbour (KNN) algorithm and has been found to perform better than a number of other multi-label methods on a variety of datasets [41]. BR and LP can work in conjunction with any classifier learning algorithms, and SMO and Random Forests are used as the base-level single-label classification algorithm of BR and LP. The implementation of multi-label D3C is within the libraries WEKA and MULAN. MULAN is an open-source library that is devoted to multi-label data mining; it includes implementation of a large number of learning algorithms, hierarchical multi-label classification and an extensive evaluation framework. The implementations of BR, LP and ML-KNN within MULAN are employed for unified experiments and evaluation. With regard to evaluation measures of multi-label classification, the aforementioned measures of Hamming loss and micro-average F-measure are utilised, and those measures are widely used in the literature and are indicative of the performance of multi-label classification methods.

Although D3C uses cross validation to tune some of the parameters, there are still many parameters that remain to be set. Apart from the parameters of the classifiers that are optimised by cross validation, other parameters are set to the default values of WEKA. The expected accuracy $\lambda$ and the interval $\Delta \lambda$ used by the framework of dynamic selection and circulating combination are set to 1 and 0.05, respectively. Additionally, the number of clusters used by ensemble pruning based on k-means clustering is set to 10 empirically. However, for multi-label classification experiments, we care about the relative performances of the compared approaches instead of their absolute performance. Thus, we do not optimise the architecture and the parameters of those methods, and all of the parameters are set to the default values of MULAN.

### 4.3. Results

The comparisons of the training time in the datasets liver, ionosphere, and pima-diabetes are shown in Fig. 8. Different colours denote the sequential training and the parallel training. For the training time, the parallel training is superior to sequential training.

The comparative results on the datasets liver, ionosphere and pima-diabetes are shown in Table 2. The result in each dataset is the average value of 10 times 10-fold cross validation when evaluating the predictive performance. Table 2 shows that all of the selective ensemble approaches are consistently better than the single
classifier algorithm in the datasets liver and pima-diabetes. In the ionosphere dataset, SFS and D3C are consistently better than the single classifier algorithm, but ensemble pruning based on k-means clustering and SBS are slightly worse than Random Forests. Moreover, pairwise comparisons indicate that D3C is significantly better than ensemble pruning based on k-means clustering, SFS, and SBS in all of the datasets.

The results of D3C with cross validation for parameter tuning and the original D3C in the Homburg dataset are shown in Table 3. CV-D3C represents D3C with cross validation for parameter tuning. For the comparative result, D3C is significantly better by parameter tuning. To show how well those methods work, we also compare them against state-of-the-art methods that have explicit semantic analysis (ESA) [40] for music genre classification. Table 3 shows that both D3C and CV-D3C are better than the support vector machine (SVM) and KNN with ESA.

In summary, Tables 2 and 3 show that D3C is superior to other compared approaches in all of the datasets. Being comparable to D3C, the performances of ensemble pruning based on k-means clustering and SBS are not very stable. Additionally, we believe that this finding arises because the instability of the clustering algorithm has a bad influence on the effect and because some classifiers with useful information are excluded from the ensemble without multilayer optimisation. The optimisation methods, including cross validation for parameter tuning and parallel optimisation for the independent training of D3C, improve the performance of D3C.

In multi-label classification experiments, the comparative results in terms of the Hamming loss and micro-average F-measure are presented in Figs. 9 and 10, respectively, for each dataset, along with the rank of each method. Additionally, Table 4 presents the average rank of each method in terms of the Hamming loss and micro-average F-measure. Over all of the multi-label datasets, D3C exhibits a high rank in terms of the Hamming loss and the micro-average F-measure in the datasets emotions and yeast. However, in the scene dataset, D3C presents only the third rank in both of the evaluation measures. In Table 4, we observe that multi-D3C exhibits a high average rank both in the Hamming loss and the micro-average F-measure. The multi-D3C presents the second best average rank in terms of the Hamming loss and
the best average rank in terms of the micro-average F-measure. In addition, the comparative results show that multi-D3C is highly competitive to ML-KNN. Additionally, it is interesting to note that LP using SMO as the base-level single-label classification algorithm presents the third best average rank with the Hamming loss and the second best average rank with the micro-average F-measure.

5. Conclusions

This paper is involved with selective ensemble learning, and the motivation is the predictive performance of the hybrid approach. The hybrid model called D3C is presented. In addition, the optimisation methods are employed to improve the performance of D3C. A large empirical study shows that D3C exhibits competitive performance against ensemble pruning based on clustering and sequential search methods. Multi-D3C, which is the multi-label version of D3C, is proposed for multi-label classification. Experiments in multi-label datasets verify the feasibility of multi-label D3C and show that multi-label D3C is competitive to ML-KNN, which strongly supports our idea of using the selective ensemble method to accomplish the multi-label classification task.

Note that although D3C has obtained impressive performance in our empirical study, we believe that there are approaches that could perform better than D3C. Moreover, the comparative results of multi-label classification are rather preliminary. Thus, D3C should be compared against other high-performing methods to evaluate its effectiveness. Moreover, D3C requires more experiments with other dataset types, such as class-imbalanced datasets. Handling class-imbalanced datasets by the selective ensemble method will be an interesting issue to be investigated, as most single-label datasets from multi-label datasets are class-imbalanced. We hypothesise that D3C adapting the class-imbalanced datasets can improve the performance for multi-label classification. To justify this hypothesis, we must conduct the relative experiments in conjunction with rigorous theoretical analysis, which is another interesting issue for future research.

Acknowledgement

The work was supported by the Natural Science Foundation of China (Nos. 61001013, 61102136, 61370010, and 81101115), the Natural Science Foundation of Fujian Province of China (Nos. 2011J05158, 2011J01371, and 2010J01351).

References


The following is a table showing the average ranking of methods for multi-label classification tasks.

**Table 4** Average ranking of methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average rank of micro-average F-measure</th>
<th>Average rank of Hamming loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-KNN</td>
<td>1.67</td>
<td>3.67</td>
</tr>
<tr>
<td>Multi-label D3C</td>
<td>2</td>
<td>1.67</td>
</tr>
<tr>
<td>BR (SMO)</td>
<td>3.67</td>
<td>5.33</td>
</tr>
<tr>
<td>BR (Random forest)</td>
<td>4.33</td>
<td>3.67</td>
</tr>
<tr>
<td>LP (SMO)</td>
<td>3.33</td>
<td>2</td>
</tr>
<tr>
<td>LP (Random forest)</td>
<td>6</td>
<td>4.67</td>
</tr>
</tbody>
</table>

This indicates that multi-label D3C is highly competitive to ML-KNN. Additionally, it is interesting to note that LP using SMO as the base-level single-label classification algorithm presents the third best average rank with the Hamming loss and the second best average rank with the micro-average F-measure.
Chen Lin received a B. Eng. degree and a Ph.D. degree both from Fudan University, China in 2004 and 2010. She is currently an assistant professor at School of Information Science and Technology, Xiamen University. Her research interests include social network analysis, web mining and information retrieval.

Quan Zou received the Bachelor Degree in Computer Science and Economics at the same time in 2004, and received Ph.D. degree in 2009 from the School of Computer Science and Technology in Harbin Institute of Technology, China. He is an assistant professor in the Department of Computer Science at Xiamen University now. His research interests include massive data mining and bioinformatics.

Wenqiang Chen received the Bachelor Degree in 2011 from the College of Mathematics and Computer Science in Fuzhou University, China. He is a graduate student in the Department of Computer Science at Xiamen University now. His research interests include web mining and information retrieval.

Cheng Qiu received the Bachelor Degree in 2010 from the College of Information Management in Jiangxi University of Finance and Economics, China, and received B. Eng. degree in 2013 from the Department of Computer Science in Xiamen University, China. His research interests include data mining and machine learning.

Yunfeng Wu received the B.E. and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), China, in 2003 and 2008, respectively. He worked as a Post-Doctoral Fellow at the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada. He is currently an Associate Professor at the School of Information Science and Technology, Xiamen University, Xiamen, Fujian Province, China. His research interests include biomedical signal processing, pattern recognition, and neural engineering. He is a Senior Member of IEEE, and Member of BMES, INNS, and CCF.

Sridhar Krishnan received the B.E. degree from Anna University, Chennai, India, in 1993, and the M.Sc. and Ph.D. degrees from the University of Calgary, Calgary, AB, Canada, in 1996 and 1999, respectively. He joined Ryerson University, Toronto, ON, Canada in 1999, and is currently a Professor and Canada Research Chair in Biomedical Signal Analysis. His research interests include adaptive signal representations and their applications in biomedicine and multimedia. Prof. Krishnan is the founding Chair of the IEEE Signal Processing Chapter of Toronto Section, and is a registered Professional Engineer of the Province of Ontario.