Ensembles of Nested Dichotomies with Multiple Subset Evaluation

Tim Leathart ✉, Eibe Frank, Bernhard Pfahringer and Geoffrey Holmes

Department of Computer Science, University of Waikato, New Zealand
tml15@students.waikato.ac.nz, {eibe,bernhard,geoff}@waikato.ac.nz

Abstract. A system of nested dichotomies (NDs) is a method of decomposing a multiclass problem into a collection of binary problems. Such a system recursively applies binary splits to divide the set of classes into two subsets, and trains a binary classifier for each split. Many methods have been proposed to perform this split, each with various advantages and disadvantages. In this paper, we present a simple, general method for improving the predictive performance of NDs produced by any subset selection techniques that employ randomness to construct the subsets. We provide a theoretical expectation for performance improvements, as well as empirical results showing that our method improves the root mean squared error of NDs, regardless of whether they are employed as an individual model or in an ensemble setting.

1 Introduction

Multiclass classification problems are commonplace in real world applications. Some models, like neural networks and random forests, are inherently able to operate on multiclass data, while other models, such as classic support vector machines, can only be used for binary (two-class) problems. The standard way to bypass this limitation is to convert the multiclass problem into a series of binary problems. There exist several methods of performing this decomposition, the most well-known including one-vs-rest [26], one-vs-one [16] and error-correcting output codes [7]. Models that are directly capable of working with multiclass data may also see improved accuracy from such a decomposition [13,25].

The use of ensembles of nested dichotomies (NDs) is one such method for decomposing a multiclass problem into several binary problems. It has been shown to outperform one-vs-rest and perform competitively compared to the aforementioned methods [11]. In an ND [10], the set of classes is recursively split into two subsets in a tree structure. At each split node of the tree, a binary classifier is trained to discriminate between the two subsets of classes. Each leaf node of the tree corresponds to a particular class. To obtain probability estimates for a particular class from an ND, assuming the base learner can produce probability estimates, one can simply compute the product of the binary probability estimates along the path to the leaf node corresponding to the class.

For non-trivial multiclass problems, the space of potential NDs is very large. An ensemble classifier can be formed by choosing suitable decompositions from

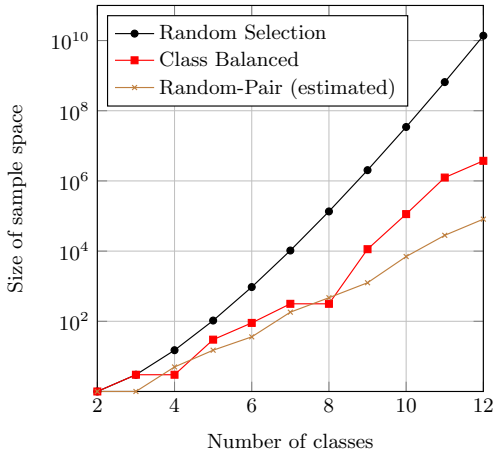

Fig. 1: Growth functions for each subset selection method discussed.

this space. In the original formulation of ensembles of NDs, decompositions are sampled with uniform probability [11], but several other more sophisticated methods for splitting the set of classes have been proposed [8,9,19]. Superior performance is achieved when ensembles of NDs are trained using common ensemble learning methods like bagging or boosting [27].

In this paper, we describe a simple method that can improve the predictive performance of NDs by considering several splits at each internal node. Our technique can be applied to NDs built with almost any subset selection method, only contributing a constant factor to the training time and no additional cost when obtaining predictions. It has a single hyperparameter λ that provides a trade-off between predictive performance and training time, making it easy to tune for a given learning problem. It is also straightforward to implement.

The paper is structured as follows. First, we describe existing methods for class subset selection in NDs. Following this, we describe our method and provide a theoretical expectation of performance improvements. We then present and discuss empirical results for our experiments. Finally, we touch on related work, before concluding and discussing future research directions.

2 Class Subset Selection Methods

At each internal node i of an ND, the set of classes present at the node $\mathcal{C}_i$ is split into two non-empty, non-overlapping subsets, $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$. In this section, we introduce existing class subset selection methods for NDs. These techniques are designed to primarily be used in an ensemble setting, where multiple ND decompositions are generated that each form an ensemble member. Note that other methods than those listed here have been proposed for constructing NDs—these are not suitable for use with our method and are discussed later in Section 5.

2.1 Random Selection

The most basic class subset selection method is to split the set of classes into two subsets using a random split.[1] This approach has several attractive qualities. It is easy to compute, and does not scale with the amount of training data, making it suitable for large datasets. Furthermore, for an n-class problem, the number of possible NDs is very large, given by the recurrence relation

$$T(n) = (2n-3) \times T(n-1)$$

where $T(1) = 1$. This ensures that, in an ensemble of NDs, there is a high level of diversity amongst ensemble members. We refer to this function that relates the number of classes to the size of the sample space of NDs for a given subset selection method as the *growth function*. Figure 1 shows the growth functions for the three selection methods discussed in this chapter.

2.2 Balanced Selection

An issue with random selection is that it can produce very unbalanced tree structures. While the number of internal nodes (and therefore, binary models) is the same in any ND for the same number of classes, an unbalanced tree often implies that internal binary models are trained on large datasets near the leaves, which has a negative effect on the time taken to train the full model. Deeper subtrees also provide more opportunity for estimation errors to accumulate. Dong *et. al.* mitigate this effect by enforcing $\mathcal{C}_i$ to be split into two subsets $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$ such that $abs(|\mathcal{C}_{i1}| - |\mathcal{C}_{i2}|) \leq 1$ [8]. This has been shown empirically to have little effect on the accuracy in most cases, while reducing the time taken to train NDs. Balanced selection has greater benefits for problems with many classes.

It is clear that the sample space of class balanced NDs is smaller than that of random NDs, but it is still large enough to ensure sufficient ensemble diversity. The growth function for class balanced NDs is given by

$$T_{CB}(n) = \begin{cases} \frac{1}{2}\binom{n}{n/2} T_{CB}(\frac{n}{2}) T_{CB}(\frac{n}{2}), & \text{if } n \text{ is even} \\ \binom{n}{(n+1)/2} T_{CB}(\frac{n+1}{2}) T_{CB}(\frac{n-1}{2}), & \text{if } n \text{ is odd} \end{cases}$$

where $T_{CB}(2) = T_{CB}(1) = 1$ [8]. Dong *et. al.* also explored a form of balancing where the amount of data in each subset is roughly equal, which gave similar results for datasets with unbalanced classes [8].

2.3 Random-Pair Selection

Random-pair selection provides a non-deterministic method of creating $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$ that groups similar classes together [19]. In random-pair selection, the base classifier is used directly to identify similar classes in $\mathcal{C}_i$. First, a random pair

[1] This is a variant of the approach from [11], where each member of the space of NDs has an equal probability of being sampled.

of classes $c_1, c_2 \in \mathcal{C}_i$ is selected, and a binary classifier is trained on just these two classes. Then, the remaining classes are classified with this classifier, and its predictions are stored as a confusion matrix M. $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$ are constructed by

$$\mathcal{C}_{i1} = \{c \in \mathcal{C}_i \setminus \{c_1, c_2\} : M_{c,c_1} \leq M_{c,c_2}\} \cup \{c_1\}$$
$$\mathcal{C}_{i2} = \{c \in \mathcal{C}_i \setminus \{c_1, c_2\} : M_{c,c_1} > M_{c,c_2}\} \cup \{c_2\}$$

where $M_{j,i}$ is defined as the number of examples of class j that were classified as class i by the binary classifier. In other words, a class is assigned to $\mathcal{C}_{i1}$ if it is less frequently confused with c_1 than with c_2, and to $\mathcal{C}_{i2}$ otherwise. Finally, the binary classifier is re-trained on the new meta-classes $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$. This way, each split is more easily separable for the base learner than a completely random split, while exhibiting a degree of randomness, which produces diverse ensembles.

Due to the fact that the size of the sample space of NDs under random-pair selection is dependent on the dataset and base learner (different initial random pairs may lead to the same split), it is not possible to provide an exact expression for the growth function $T_{RP}(n)$; using logistic regression (LR) as the base learner, it has been empirically estimated to be

$$T_{RP}(n) = p(n) T_{RP}\left(\frac{n}{3}\right) T_{RP}\left(\frac{2n}{3}\right)$$

where $T_{RP}(2) = T_{RP}(1) = 1$ and $p(n) = 0.3812n^2 - 1.4979n + 2.9027$ [19].

3 Multiple Subset Evaluation

In class subset selection methods, for each node i, a single class split $(\mathcal{C}_{i1}, \mathcal{C}_{i2})$ of $\mathcal{C}_i$ is considered, produced by some splitting function $S(\mathcal{C}_i) : \mathbb{N}^n \to \mathbb{N}^a \times \mathbb{N}^b$ where $a + b = n$. Our approach for improving the predictive power of NDs is a simple extension. We propose to, at each internal node i, consider λ subsets $\{(\mathcal{C}_{i1}, \mathcal{C}_{i2})_1 \dots (\mathcal{C}_{i1}, \mathcal{C}_{i2})_\lambda\}$ and choose the split for which the corresponding model has the lowest training root mean squared error (RMSE). The RMSE is defined as the square root of the Brier score [5] divided by the number of classes:

$$\text{RMSE} = \sqrt{\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} (\hat{y}_{ij} - y_{ij})^2}$$

where n is the number of instances, m is the number of classes, $\hat{y}_{ij}$ is the estimated probability that instance i is of class j, and y_{ij} is 1 if instance i actually belongs to class j, and 0 otherwise. RMSE is chosen over other measures such as classification accuracy because it is smoother and a more sensitive indicator of generalisation performance. Previously proposed methods with single subset selection can be considered a special case of this method where $\lambda = 1$.

Although conceptually simple, this method has several attractive qualities. By choosing the best of a series of models at each internal node, the overall

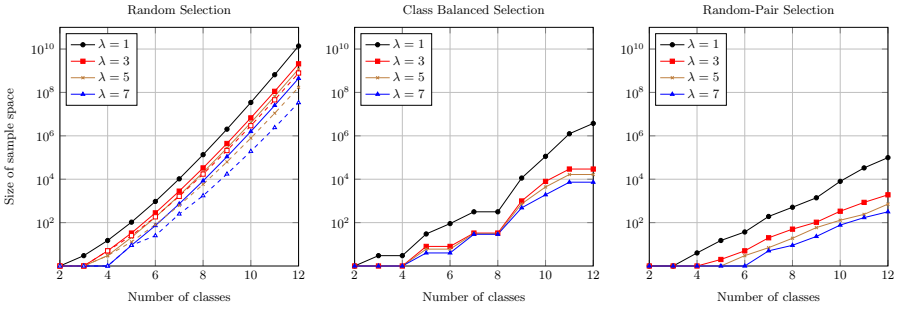

Fig. 2: Left: Growth functions for random selection with multiple subset evaluation and $\lambda \in \{1, 3, 5, 7\}$. Solid lines indicate the upper bound, and dashed lines indicate the lower bound. Middle: Considering class-balanced selection instead of random selection. Right: Growth functions for random-pair selection.

performance should improve, assuming the size of the sample space of NDs is not hindered to the point where ensemble diversity begins to suffer.

Multiple subset evaluation is also widely applicable. If a subset selection method S has some level of randomness, then multiple subset evaluation can be used to improve the performance. One nice feature is that advantages pertaining to S are retained. For example, if class-balanced selection is chosen due to a learning problem with a very high number of classes, we can boost the predictive performance of the ensemble while keeping each ND in the ensemble balanced. If random-pair selection is chosen because the computational budget for training is high, then we can improve the predictive performance further than single subset selection in conjunction with random-pair selection.

Finally, implementing multiple subset evaluation is very simple, and the computational cost for evaluating multiple subsets of classes scales linearly in the size of the tuneable hyperparameter λ, making the tradeoff between predictive performance and training time easy to navigate. Additionally, multiple subset evaluation has no effect on prediction times.

Higher values of λ give diminishing returns on predictive performance, so a value that is suitable for the computational budget should be chosen. When training an ensemble of NDs, it may be desirable to adopt a *class threshold*, where $\lambda = 1$ is used if fewer than a certain number of classes is present at an internal node. This reduces the probability that the same subtrees will appear in many ensemble members, and therefore reduce ensemble diversity. In lower levels of the tree, where the number of classes is small, the number of possible binary problems is relatively low (Fig. 2).

3.1 Effect on Growth Functions

Performance of an ensemble of NDs relies on the size of the sample space of NDs, given an n-class problem, to be relatively large. Multiple subset evaluation

removes the $\lambda - 1$ class splits that correspond to the worst-performing binary models at each internal node i from being able to be used in the tree. The effect of multiple subset evaluation on the growth function is non-deterministic for random selection, as the sizes of $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$ affect the values of the growth function for the subtrees that are children of i. The upper bound occurs when all worst-performing splits isolate a single class, and the lower bound is given when all worst-performing splits are class-balanced. Class-balanced selection, on the other hand, is affected deterministically as the size of $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$ are the same for the same number of classes.

Growth functions for values of $\lambda \in \{1, 3, 5, 7\}$, for random, class balanced and random-pair selection methods, are plotted in Figure 2. The growth curves for random and class balanced selection were generated using brute-force computational enumeration, while the effect on random-pair selection is estimated.

3.2 Analysis of Error

In this section, we provide a theoretical analysis showing that performance of each internal binary model is likely to be improved by adopting multiple subset evaluation. We also show empirically that the estimates of performance improvements are accurate, even when the assumptions are violated.

Let E be a random variable for the training root mean squared error (RMSE) for some classifier for a given pair of class subsets $\mathcal{C}_{i1}$ and $\mathcal{C}_{i2}$, and assume $E \sim N(\mu, \sigma^2)$ for a given dataset under some class subset selection scheme. For a given set of λ selections of subsets $\mathcal{S} = \{(\mathcal{C}_{i1}, \mathcal{C}_{i2})_1, \ldots, (\mathcal{C}_{i1}, \mathcal{C}_{i2})_\lambda\}$ and corresponding training RMSEs $\mathcal{E} = \{E_1, \ldots, E_\lambda\}$, let $\hat{E}_\lambda = min(\mathcal{E})$. There is no closed form expression for the expected value of $\hat{E}_\lambda$, the minimum of a set of normally distributed random variables, but an approximation is given by

$$\mathbb{E}[\hat{E}_\lambda] \approx \mu + \sigma \Phi^{-1}\left(\frac{1-\alpha}{\lambda - 2\alpha + 1}\right) \tag{1}$$

where $\Phi^{-1}(x)$ is the inverse normal cumulative distribution function [28], and the *compromise value* α is the suggested value for λ given by Harter [15].[2]

Figure 3 illustrates how this expected value changes when increasing values of λ from 1 to 5. The first two rows show the distribution of E and estimated $\mathbb{E}[\hat{E}_\lambda]$ on the UCI dataset `mfeat-fourier`, for a LR model trained on 1,000 random splits of the class set $\mathcal{C}$. These rows show the training and testing RMSE respectively, using 90% of the data for training and the rest for testing. Note that as λ increases, the distribution of the train and test error shifts to lower values and the variance decreases.

This reduction in error affects each binary model in the tree structure, so the effects accumulate when constructing an ND. The third row shows the distribution of RMSE of 1,000 NDs trained with multiple subset evaluation on

[2] Appropriate values for α for a given λ can be found in Table 3 of [15].

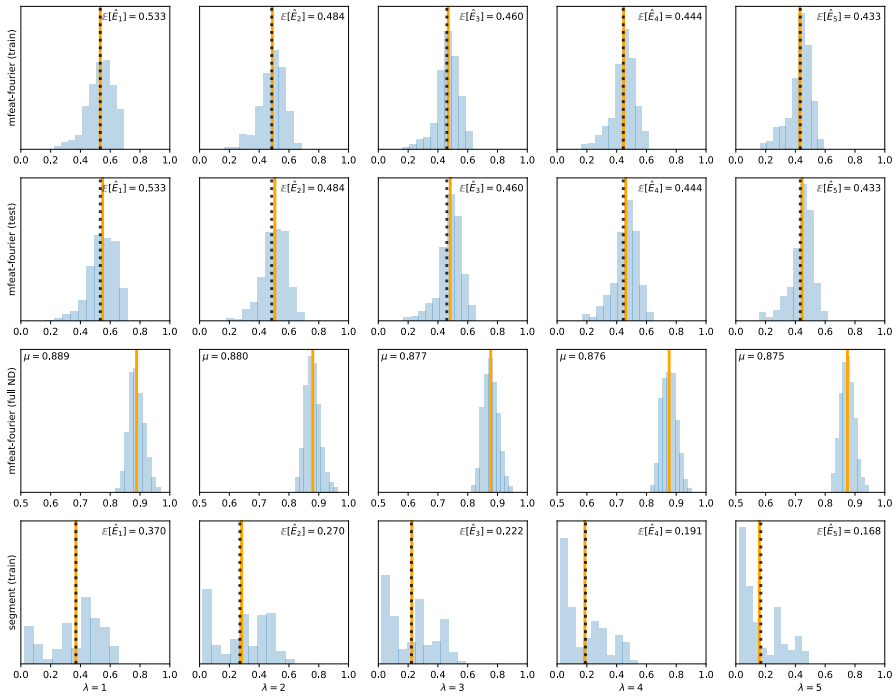

Fig. 3: Empirical distribution of RMSE of LR trained on random binary class splits, for values of λ from one to five. The shaded region indicates empirical histogram, the orange vertical line shows the empirical mean, and the black dotted vertical line is the expected value, estimated from (1). Top two rows: train and test RMSE of LR trained on random binary class splits of `mfeat-fourier` UCI dataset. For the test data, the approximated value of $\mathbb{E}[E_\lambda]$ is estimated from the mean and standard deviation of the train error. Third row: train RMSE of an ND built with random splits and multiple-subset evaluation, trained on `mfeat-fourier` for different values of λ. Bottom row: train RMSE of LR trained on random binary class splits of `segment` data.

`mfeat-fourier`, using LR as the base learner, considering increasing values of λ. As expected, a reduction in train error with diminishing returns is seen.

In order to show an example of how the estimate from (1) behaves when the error is not normally distributed, the distribution of E for LR trained on the `segment` UCI data is plotted in the bottom row. The assumption of normality is commonly violated in real datasets, as the distribution is often skewed towards zero error. As with the other examples, 1,000 different random choices for $\mathcal{C}_1$ and $\mathcal{C}_2$ were used to generate the histogram. Although the distribution in this case is not very well modelled by a Gaussian, the approximation of $\mathbb{E}[\hat{E}_\lambda]$ from (1) still closely matches the empirical mean. This shows that even when the normality assumption is violated, performance gains of the same degree can be achieved.

This example is not atypical; the same behaviour was observed on the entire collection of datasets used in this study.

4 Experimental Results

All experiments were conducted in WEKA 3.9 [14], and performed with 10 times 10-fold cross validation. We use class-balanced NDs and NDs built with random-pair selection, with LR as the base learner. For both splitting methods, we compare values of $\lambda \in \{1, 3, 5, 7\}$ in a single ND structure, as well as in ensemble settings with bagging [4] and AdaBoost [12]. The default settings in WEKA were used for the `Logistic` classifier as well as for the `Bagging` and `AdaBoostM1` meta-classifiers. We evaluate performance on a collection of 15 commonly used datasets from the UCI repository [21], as well as the MNIST digit recognition dataset [20]. Note that for MNIST, we report results of 10-fold cross-validation over the entire dataset rather than the usual train/test split. Datasets used in our experiments and their characteristics are listed in the supplementary material.

We provide critical difference plots [6] to summarise the results of the experiments. These plots present average ranks of models trained with differing values of λ. Models producing results that are not significantly different from each other at the 0.05 significance level are connected with a horizontal black bar. Full results tables showing RMSE for each experimental run, including significance tests, are available in the supplementary materials.

4.1 Individual Nested Dichotomies

Restricting the sample space of NDs through multiple subset evaluation is expected to have a greater performance impact on smaller ensembles than larger ones. This is because in a larger ensemble, a poorly performing ensemble member does not have a large impact on overall performance. On the other hand, in a small ensemble, one poorly performing ensemble member can degrade ensemble performance significantly. In the extreme case, where a single ND is trained, there is no need for ensemble diversity, so a technique for improving the predictive performance of an individual ND should be effective. Therefore, we first compare the performance of single NDs for different values of λ.

Figure 4 shows critical difference plots for both subset selection methods. Class balanced selection shows a clear trend that increasing λ improves the RMSE, with the average rank for $\lambda = 1$ being exactly 4. For random-pair selection, choosing $\lambda = 3$ is shown to be statistically indistinguishable from $\lambda = 1$, while higher values of λ give superior results on average.

4.2 Ensembles of Nested Dichotomies

Typically, NDs are utilised in an ensemble, so we investigate the predictive performance of ensembles of ten NDs with multiple subset evaluation, with bagging and AdaBoost employed as the ensemble methods.

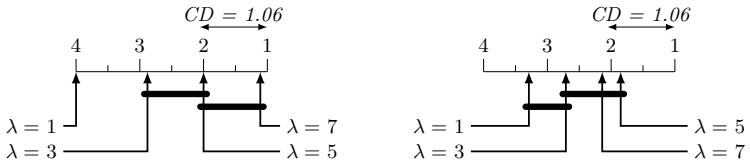

Fig. 4: Critical differences charts for individual NDs. Left: Class balanced selection. Right: Random-pair selection.

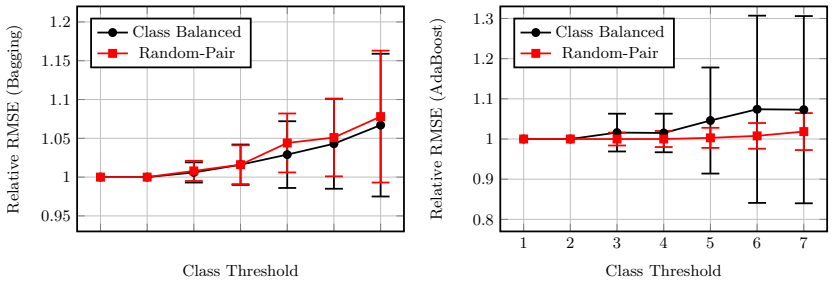

Fig. 5: Effect of changing the class threshold on RMSE for ensembles of NDs.

Class Threshold. The number of binary problems is reduced when multiple subset evaluation is applied, which can have a negative effect on ensemble diversity, potentially reducing predictive performance. To investigate this, we built ensembles of NDs with multiple subset evaluation by introducing a *class threshold*, the number of classes present at a node required to perform multiple subset evaluation, and varying its value from one to seven. We plot the test RMSE, relative to having a class threshold of one, averaged over all the datasets in Figure 5. Interestingly, the RMSE increases monotonically, showing that the potentially reduced ensemble diversity does not have a negative effect on the RMSE for ensembles of this size. Therefore, we use a class threshold of one in our subsequent experiments. However, note that increasing the class threshold has a positive effect on training time, so it may be useful to apply it in practice.

Number of Subsets. We now investigate the effect of λ when using bagging and boosting. Figure 6 shows critical difference plots for bagging. Both subset selection methods improve when utilising multiple subset selection. When class-balanced selection is used, as was observed for single NDs, the average ranks across all datasets closely correspond to the integer values, showing that increasing the number of subsets evaluated consistently improves performance. For random-pair selection, a more constrained subset selection method, each value of $\lambda > 1$ is statistically equivalent and superior to the single subset case.

The critical difference plots in Figure 7 (left) show boosted NDs are significantly improved by increasing the number of subsets sufficiently when class-balanced NDs are used. Results are less consistent for random-pair selection,

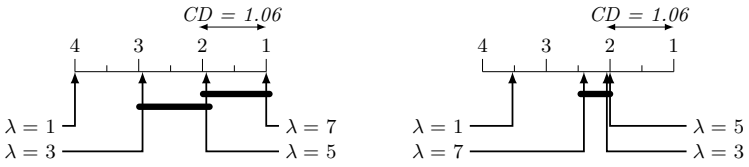

Fig. 6: Critical differences charts for ensemble of ten bagged NDs. Left: Class balanced selection. Right: Random-pair selection.

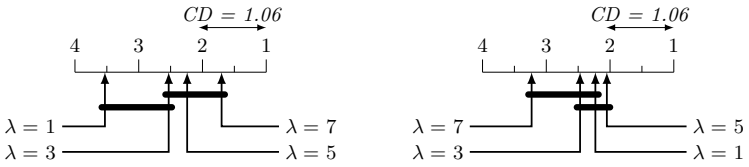

Fig. 7: Critical differences charts for ensemble of ten NDs, ensembled with AdaBoost. Left: Class balanced selection. Right: Random-pair selection.

reflected in the critical differences plot (Fig. 7, right), which shows single subset evaluation statistically equivalent to multiple subset selection for all values of λ, with $\lambda = 7$ performing markedly worse on average. As RMSE is based on probability estimates, this may be in part due to poor probability calibration, which is known to affect boosted ensembles [24] and NDs [18].

5 Related Work

Splitting a multiclass problem into several binary problems in a tree structure is a general technique that has been referred to by different names in the literature. For example, in a multiclass classification context, NDs in the broadest sense of the term have been examined as filter trees, conditional probability trees, and label trees. Beygelzimer et al. proposed algorithms which build balanced trees and demonstrate the performance on datasets with very large numbers of classes. Filter trees, with deterministic splits [3], as well as conditional probability trees, with probabilistic splits [2], were explored. Bengio et al. [1] define a tree structure and optimise all internal classifiers simultaneously to minimise the tree loss. They also propose to learn a low-dimensional embedding of the labels to improve performance, especially when many classes are present. Melnikov and Hullermeier [23] also showed that a method called best-of-k models—simply sampling k random NDs and choosing the best one based on validation error—gives competitive predictive performance to the splitting heuristics discussed so far for individual NDs. However, it is very expensive at training time, as k independent NDs must be built and tested on a held-out set.

A commonality of these techniques is that they attempt to build a single ND structure with the best performance. NDs that we consider in this paper, while conceptually similar, differ from these methods because they are intended to be

trained in an ensemble setting, and as such, each individual ND is not built with optimal performance in mind. Instead, a group of NDs is built to maximise ensemble performance, so diversity amongst the ensemble members is key [17].

NDs based on clustering [9] are deterministic and used in an ensemble by resampling or reweighting the input. They are built by finding the two classes $(c_1, c_2) \in \mathcal{C}_i$ for which the centroids are furthest from each other, and grouping the remaining classes based on the distance of their centroids from c_1 and c_2.

Wever et al. [29] utilise genetic algorithms to build NDs. In their method, a population of random NDs is sampled and is evolved for several generations. The final ND is chosen as the best performing model on a held-out validation set. An ensemble of k NDs is produced by evolving k populations independently, and taking the best-performing model from each population.

6 Conclusion

Multiple subset selection in NDs can improve predictive performance while retaining the particular advantages of the subset selection method employed. We present an analysis of the effect of multiple subset selection on expected RMSE and show empirically in our experiments that adopting our technique can improve predictive performance, at the cost of a constant factor in training time.

The results of our experiments suggest that for class-balanced selection, performance can be consistently improved significantly by utilising multiple subset evaluation. For random-pair selection, $\lambda = 3$ yields the best trade-off between predictive performance and training time, but when AdaBoost is used, multiple subset evaluation is not generally beneficial.

Avenues of future research include comparing multiple subset evaluation with base learners other than LR. It is unlikely that training RMSE of the internal models will be a reliable indicator when selecting splits based on more complex models such as decision trees or random forests, so other metrics may be needed. Also, it may be beneficial to choose subsets such that maximum ensemble diversity is achieved, possibly through information theoretic measures such as variation of information [22].

References

1. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: NIPS. pp. 163–171 (2010)
2. Beygelzimer, A., Langford, J., Lifshits, Y., Sorkin, G., Strehl, A.: Conditional probability tree estimation analysis and algorithms. In: UAI. pp. 51–58 (2009)
3. Beygelzimer, A., Langford, J., Ravikumar, P.: Error-correcting tournaments. In: ALT. pp. 247–262. Springer (2009)
4. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
5. Brier, G.: Verification of forecasts expressed in term of probabilities. Monthly Weather Review **78**, 1–3 (1950)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. JMLR **7**(Jan), 1–30 (2006)

7. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. JAIR pp. 263–286 (1995)
8. Dong, L., Frank, E., Kramer, S.: Ensembles of balanced nested dichotomies for multi-class problems. In: PKDD, pp. 84–95. Springer (2005)
9. Duarte-Villaseñor, M.M., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Flores-Garrido, M.: Nested dichotomies based on clustering. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 162–169. Springer (2012)
10. Fox, J.: Applied Regression Analysis, Linear Models, and Related Methods. Sage (1997)
11. Frank, E., Kramer, S.: Ensembles of nested dichotomies for multi-class problems. In: ICML. p. 39. ACM (2004)
12. Freund, Y., Schapire, R.E.: Game theory, on-line prediction and boosting. In: COLT. pp. 325–332 (1996)
13. Fürnkranz, J.: Round robin classification. JMLR **2**(Mar), 721–747 (2002)
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter **11**(1), 10–18 (2009)
15. Harter, H.L.: Expected values of normal order statistics. Biometrika **48**(1/2), 151–165 (1961)
16. Hastie, T., Tibshirani, R., et al.: Classification by pairwise coupling. The Annals of Statistics **26**(2), 451–471 (1998)
17. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning **51**(2), 181–207 (2003)
18. Leathart, T., Frank, E., Holmes, G., Pfahringer, B.: On calibration of nested dichotomies. In: PAKDD. Springer (2019)
19. Leathart, T., Pfahringer, B., Frank, E.: Building ensembles of adaptive nested dichotomies with random-pair selection. In: ECMLPKDD. pp. 179–194. Springer (2016)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. IEEE **86**(11), 2278–2324 (1998)
21. Lichman, M.: UCI machine learning repository (2013)
22. Meilă, M.: Comparing clusterings by the variation of information. In: Learning Theory and Kernel Machines, pp. 173–187. Springer (2003)
23. Melnikov, V., Hüllermeier, E.: On the effectiveness of heuristics for learning nested dichotomies: an empirical analysis. Machine Learning **107**(8-10), 1–24 (2018)
24. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: ICML. pp. 625–632. ACM (2005)
25. Pimenta, E., Gama, J.: A study on error correcting output codes. In: Portuguese Conference on Artificial Intelligence. pp. 218–223. IEEE (2005)
26. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. JMLR **5**, 101–141 (2004)
27. Rodríguez, J.J., García-Osorio, C., Maudes, J.: Forests of nested dichotomies. Pattern Recognition Letters **31**(2), 125–132 (2010)
28. Royston, J.: Algorithm AS 177: Expected normal order statistics (exact and approximate). Journal of the Royal Statistical Society. Series C (Applied Statistics) **31**(2), 161–165 (1982)
29. Wever, M., Mohr, F., Hüllermeier, E.: Ensembles of evolved nested dichotomies for classification. In: GECCO. pp. 561–568. ACM (2018)