Optimizing the Structure of Nested Dichotomies: A Comparison of Two Heuristics

Vitalik Melnikov[1], Eyke Hüllermeier[2]

Department of Computer Science
Paderborn University, Germany
E-Mail: [1]melnikov@mail.upb.de, [2]eyke@upb.de

Abstract

In machine learning, nested dichotomies are used to decompose a multi-class classification problem into a set of binary problems. The performance of a nested dichotomy strongly depends on the structure of such a decomposition. In this paper, we compare the random-pair heuristic, a state-of-the-art approach for optimizing the structure, with an extremely simple alternative: Leveraging a procedure for uniform sampling of dichotomies, the Best-of-K heuristic picks the (presumably) best among K randomly generated dichotomies. Interestingly, Best-of-K turns out to be highly competitive, and outperforms the random-pair heuristic in the case of simple base learners such as logistic regression.

1 Introduction

Nested dichotomies are known as models for polychotomous data in statistics and used as classifiers for multi-class problems in machine learning [4]. Based on a recursive binary partitioning of the set of classes, nested dichotomies reduce the original multi-class problem to a set of binary problems, for which any (probabilistic) binary classifier can be used. For example, the dichotomy shown in Fig. 1 decomposes a problem with four classes into three binary problems: The first classifier (C_1) is supposed to separate class 3 from the meta-class $\{1, 2, 4\}$, i.e., the union of classes 1, 2, and 4; likewise, the second classifier separates classes $\{2, 4\}$ from 1, and the third classifier the classes 2 and 4.

Once the hierarchy of classifiers required by a nested dichotomy have been trained, a new instance x can be classified in a straightforward way,

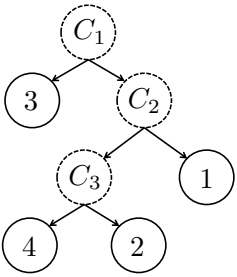

Figure 1: Example of a nested dochotomy.

namely by submitting x to the root of the dichotomy and propagating it to one of the leaf nodes, based on the decisions of the classifiers along the corresponding path. As an appealing property, we note that nested dichotomies allow for producing probability estimates (instead of hard class assignments) in a very natural way, provided each classifier predicts suitable conditional class probabilities at the inner nodes. In that case, the probability of any class y is simply given by the product of conditional probabilities on the path from the root to the leaf node marked with y.

In practice, nested dichotomies have been shown to yield superb predictive accuracy [4, 6, 8]. Yet, the performance of the multi-class classifier eventually produced may strongly depend on the structure of the dichotomy. In fact, the structure of a dichotomy specifies the subset of all binary problems that need to be solved, and some of them might be much more difficult than others. This is illustrated in Fig. 2, where the distribution of predictive accuracies (on the test data) of nested dichotomies is shown for the pendigits dataset. As can be seen, the variance is higher if logistic regression is used as a base learner, and smaller with decision trees. This observation can be explained by the fact that the latter is much more flexible than the former: A decision tree is a complex, highly nonlinear model, which can compensate a suboptimal structure much better than a simple linear model as fit by logistic regression (but of course also comes with a higher danger of poor generalization due to overfitting); or, stated differently, the choice of a suitable structure is much more critical when using simple models such as linear discriminants.

Consequently, finding a suitable structure is an important prerequisite for successful learning with nested dichotomies. Since the number of candidate structures is huge (double factorial in the number of classes),

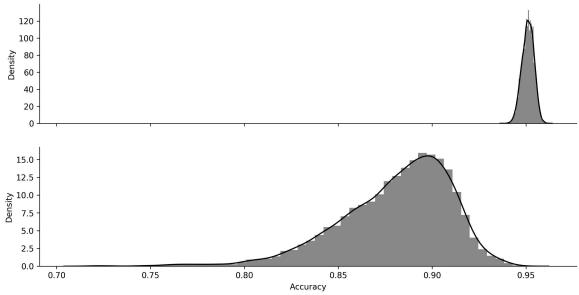

Figure 2: Gaussian kernel density estimation of accuracy distribution for the pendigits dataset based on a random sample of 10000 nested dichotomies, using decision tree (CART, in the top plot) and logistic regression (in the bottom plot) as base learner.

several heuristic methods have been proposed for constructing dichotomies specifically tailored for the data at hand [2, 6, 3]. Although some heuristics are designed to select an optimal *subset* of nested dichotomies, to be used as an ensemble of classifiers, we focus on finding a *single* dichotomy in this paper (for example because an ensemble is not desirable, due to reasons of complexity or interpretability).

According to recent empirical studies, the current state of the art is the *random-pair* selection heuristic (RPND) proposed by Leathart et al. [6]. At each inner node of a nested dichotomy, this approach obtains a split of the classes $\mathcal{Y}$ associated with that node into two meta-classes as follows: Two classes $Y_i, Y_j \in \mathcal{Y}$ are selected (uniformly) at random, and the base learner is used to train a classifier $C_{i,j}$ discriminating these two classes. Both classes define the seed for a meta-class $\mathcal{Y}_i$ and $\mathcal{Y}_j$, respectively, and each remaining class $Y \in \mathcal{Y} \setminus \{Y_i, Y_j\}$ is added to one of these meta-classes, depending on whether $C_{i,j}$ assigns the majority of instances of Y to $\mathcal{Y}_i$ or $\mathcal{Y}_j$. Once the complete structure of the dichotomy has been determined, the classifiers at the inner nodes are trained on the corresponding splits.

In this paper, we reconsider the random-pair heuristic, challenging it with the arguably most simple heuristic one may image: generating K dichotomies at random and selecting the (presumably) best one. In Section 4, we demonstrate that this simple Best-of-K heuristic is highly competitive and, depending on the value K, even able to outperform the random-pair heuristic. As a side product, the paper makes another

interesting contribution, namely an efficient algorithm for sampling nested dichotomies uniformly at random. Although sampling procedures have already been used in several papers [4, 6, 2], an explicit algorithm has never been provided; interestingly, a naive sampling procedure will yield a non-uniform distribution.

2 Uniform random sampling of nested dichotomies

On the structural side, there is a one-to-one correspondence between nested dichotomies and binary trees. Since each node in a nested dichotomy has either none or exactly two children, every nested dichotomy is a (rooted) full binary tree. In addition, each leaf node in a nested dichotomy is labeled with the corresponding class. This labeling uniquely determines the dichotomies at all inner nodes. The problem of generating a nested dichotomy can therefore be divided into two steps: (i) generation of a rooted terminally labeled full binary tree and (ii) propagation of leaf labels towards the root in order to determine the dichotomies at the inner nodes.

For the problem (i), several strategies were proposed in the literature [9, 5]. Here, we provide a two-step procedure for uniform random sampling of rooted terminally labeled full binary trees as suggested by Furnas [5]:

1. Generation of an unrooted terminally labeled full binary tree T_c. Given is a set of c terminal nodes (leafs).
   a) Create a doublet "tree" T_2 by connecting nodes 1 and 2 by a single edge.
   b) Until all c terminal nodes are connected to the tree, proceed with the following random augmentation:
      i. Given a tree T_k on $k < c$ terminal nodes, select an edge of T_k uniformly at random.
      ii. On this edge, a new internal node of degree 3 is added and the $(k+1)$st terminal node is connected. The result is a binary tree T_{k+1} on $k+1$ terminal nodes.
2. Transformation of T_c into a rooted tree.
   a) Choose an edge of T_c randomly with uniform probability.
   b) Introduce a new root node of degree 2 on this edge.

Furnas proved this procedure to provide a uniform random sampling of rooted terminally labeled full binary trees. The time complexity of the first step is linear in the number of classes c. The second step can be implemented in constant time, leading to an overall time complexity of $O(c)$.

For the label propagation problem (ii), we suggest the following solution. Every leaf node is labeled with the number $b_{c_i} = 2^{c_i}$, where c_i is the corresponding class in the original problem. The dichotomies at the inner nodes are encoded with a pair of numbers $[d_l, d_r]$, denoting the sum of all leaf labels in the left and right subtree, respectively. Since all leaf nodes are encoded uniquely (with only a single '1' in the corresponding bit string), the dichotomy encodings are also unique. To propagate the leaf labels, we traverse the nested dichotomy in postorder and encode every inner node with the sum of the dichotomies of its child nodes $[d_{l_l} + d_{l_r}, d_{r_l} + d_{r_r}]$. Since the complexity of tree traversal is linear in the number of nodes and the summation, and generating the leaf labels is nearly constant for a usual number of classes[1], step (ii) has an overall time complexity of $O(c)$. Using this approach, a single nested dichotomy can be sampled uniformly with the time complexity $O(c)$, i.e., linear in the number of classes in the original problem.

3 The Best-of-K Heuristic

Equipped with the sampling procedure for nested dichotomies from the previous section, the algorithm for the Best-of-K heuristic is quite straightforward. It is parametrized by the number K of nested dichotomies to be generated and evaluated. Given a (training) dataset with c classes and a base learner, Best-of-K consists of three steps:

1. Sample K nested dichotomies on c terminal nodes uniformly at random.
2. Train these models on the given data with the provided base learner.
3. Select the best performing model based on the validation on the training data (e.g., with the smallest training error).

[1]The length of the encoding number (as a bit string) is at most c. Both encoding operations (bit shift and summation) will have roughly constant time for up to several hundred classes.

Several remarks on this approach are in order. First, since the sampling procedure only depends on the number of classes c, and not on the data itself, it can be carried out in a preprocessing step, thereby reducing the overall execution time. Second, the training in the second step can be done independently for each nested dichotomy. Thus, Best-of-K is naturally implemented in a parallel way. If K cores are available, a speedup factor close to K can be achieved in comparison to a standard (sequential) implementation (the runtime is the maximal training time of a single nested dichotomy). Third, while being computationally cheap, the selection of the best performing model based on the training error is arguably not optimal—a better selection could probably be made on the basis of a validation error. However, our experiments in the next section suggest this strategy to be good enough in general.

4 Empirical Study

We compare the Best-of-K heuristic with the state of the art RPND on 15 multi-class datasets (Table 1) from the UCI repository [1]. On every dataset, we perform the following evaluation: For a given parameter $K \in \{1, 5, 10, 20, 50, 100\}$, we generate a single nested dichotomy using the Best-of-K heuristic and a single nested dichotomy using RPND. Both models are tested with two base learners: decision tree (CART) and logistic regression from the scikit-learn framework (ver. 0.19) [7]. The hyper-parameters of the base learners are set to default values, except for the minimal decrease of the impurity in CART, which is set to .0001 to prevent overfitting. Every generated nested dichotomy is trained multiple times by using 10-fold cross-validation, and the mean predictive accuracy is stored. Since both heuristics are randomized, we repeat this procedure 50 times for every dataset to stabilize the results. The mean and the standard deviation of predictive accuracy over these runs are given in the Table 2.

In Fig. 3, the difference $A_{diff} = A_{BoK} - A_{RPND}$ in mean accuracy is shown. Unsurprisingly, the performance of Best-of-K increases for higher values of K. For $K > 5$, Best-of-K outperforms the RPND heuristic on most of the datasets if logistic regression is used as the base learner. For example, for $K = 5$ and logistic regression as base learner, the mean time ratio over all datasets is 0.58 and the mean accuracy difference is 0.015. In the case of CART as a base learner, both heuristics perform comparable.

Table 1: The datasets used in the study. For the datasets krkopt and vowel, a one-hot encoding has been used for categorical features.

dataset	Classes	Features	Instances
LED24	10	25	5000
zoo	7	18	101
krkopt-bin	18	7	28056
segment	7	20	2310
mfeat-morphological	10	7	2000
mfeat-factors	10	217	2000
mfeat-fourier	10	77	2000
mfeat-karhunen	10	65	2000
mfeat-pixel	10	241	2000
letter	26	17	20000
optdigits	10	65	5620
page-blocks	5	11	5473
pendigits	10	17	10992
vowel-bin	11	14	990
yeast	10	9	1484

This seems plausible for the reason already mentioned: Since (unpruned) decision trees are able to generate complex models (with a tendency to overfit), they can compensate for a possibly suboptimal structure of a dichotomy. The small decrease in the performance of such models for higher values of K suggests that, at least for some datasets, the selection of a nested dichotomy based on the training error is too optimistic.

In Table 3, the mean and standard deviation of the generation time are given. The time was measured for the Best-of-K heuristic with parallelization, i.e., we assume to have K cores available and measure the maximal generation time for a single nested dichotomy. On a single core CPU, this heuristics would be roughly K times slower. The time comparison is provided in Fig. 4. The mean time ratio is defined as the ratio of mean times for generating a single nested dichotomy with both heuristics: $T = T_{BoK}/T_{RPND}$. Since the RPND heuristic depends on the size of the dataset, this ratio is lower for larger datasets such as letter or krkopt.

Table 2: Mean and standard deviation of the predictive accuracy for (a) CART and (b) logistic regression as the base learner. 'Bo' is the corresponding Best-of-K heuristic.

(a)

dataset	Bo1	Bo5	Bo10	Bo20	Bo50	Bo100	RPND
LED24	$.61 \pm .01$	$.61 \pm .01$	$.61 \pm .01$	$.60 \pm .01$	$.60 \pm .01$	$.60 \pm .02$	$.62 \pm .01$
zoo	$.92 \pm .02$	$.92 \pm .02$	$.92 \pm .02$	$.92 \pm .02$	$.92 \pm .02$	$.92 \pm .02$	$.90 \pm .01$
krkopt-bin	$.71 \pm .02$	$.72 \pm .01$	$.73 \pm .01$	$.74 \pm .01$	$.74 \pm .01$	$.75 \pm .01$	$.70 \pm .01$
segment	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .00$	$.96 \pm .00$	$.96 \pm .00$	$.96 \pm .00$	$.96 \pm .00$
mfeat-morph.	$.65 \pm .01$	$.65 \pm .01$	$.65 \pm .01$	$.65 \pm .01$	$.65 \pm .01$	$.65 \pm .01$	$.65 \pm .00$
mfeat-factors	$.87 \pm .01$	$.87 \pm .01$	$.87 \pm .01$	$.87 \pm .01$	$.87 \pm .01$	$.87 \pm .01$	$.87 \pm .01$
mfeat-fourier	$.69 \pm .01$	$.69 \pm .01$	$.69 \pm .01$	$.69 \pm .01$	$.69 \pm .01$	$.69 \pm .01$	$.70 \pm .00$
mfeat-karhunen	$.78 \pm .01$	$.78 \pm .01$	$.78 \pm .01$	$.78 \pm .01$	$.78 \pm .02$	$.78 \pm .02$	$.78 \pm .01$
mfeat-pixel	$.86 \pm .01$	$.85 \pm .01$	$.86 \pm .01$	$.85 \pm .01$	$.85 \pm .01$	$.85 \pm .01$	$.86 \pm .01$
letter	$.85 \pm .00$	$.85 \pm .00$	$.85 \pm .00$	$.85 \pm .00$	$.85 \pm .00$	$.85 \pm .00$	$.84 \pm .00$
optdigits	$.89 \pm .01$	$.89 \pm .01$	$.89 \pm .01$	$.89 \pm .01$	$.89 \pm .01$	$.89 \pm .01$	$.89 \pm .00$
page-blocks	$.97 \pm .00$	$.97 \pm .00$	$.97 \pm .00$	$.97 \pm .00$	$.97 \pm .00$	$.97 \pm .00$	$.97 \pm .00$
pendigits	$.95 \pm .00$	$.95 \pm .00$	$.95 \pm .00$	$.95 \pm .00$	$.95 \pm .00$	$.95 \pm .00$	$.95 \pm .00$
vowel-bin	$.76 \pm .01$	$.76 \pm .01$	$.76 \pm .01$	$.76 \pm .01$	$.76 \pm .01$	$.76 \pm .01$	$.77 \pm .01$
yeast	$.52 \pm .01$	$.52 \pm .01$	$.52 \pm .01$	$.51 \pm .01$	$.51 \pm .01$	$.51 \pm .01$	$.51 \pm .01$

(b)

dataset	Bo1	Bo5	Bo10	Bo20	Bo50	Bo100	RPND
LED24	$.71 \pm .02$	$.72 \pm .01$	$.72 \pm .01$	$.72 \pm .01$	$.72 \pm .01$	$.72 \pm .00$	$.72 \pm .00$
zoo	$.91 \pm .02$	$.91 \pm .02$	$.92 \pm .02$	$.91 \pm .02$	$.91 \pm .02$	$.92 \pm .02$	$.91 \pm .01$
krkopt-bin	$.29 \pm .01$	$.31 \pm .01$	$.31 \pm .01$	$.31 \pm .01$	$.32 \pm .01$	$.32 \pm .01$	$.29 \pm .01$
segment	$.89 \pm .04$	$.91 \pm .04$	$.92 \pm .03$	$.93 \pm .03$	$.94 \pm .03$	$.94 \pm .03$	$.86 \pm .01$
mfeat-morph.	$.64 \pm .04$	$.67 \pm .03$	$.68 \pm .02$	$.68 \pm .02$	$.69 \pm .02$	$.69 \pm .01$	$.59 \pm .02$
mfeat-factors	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .00$
mfeat-fourier	$.79 \pm .01$	$.79 \pm .01$	$.80 \pm .01$	$.80 \pm .01$	$.80 \pm .01$	$.80 \pm .01$	$.79 \pm .00$
mfeat-karhunen	$.91 \pm .01$	$.91 \pm .01$	$.91 \pm .01$	$.92 \pm .01$	$.92 \pm .01$	$.92 \pm .01$	$.91 \pm .00$
mfeat-pixel	$.88 \pm .02$	$.88 \pm .02$	$.89 \pm .02$	$.89 \pm .01$	$.89 \pm .01$	$.89 \pm .01$	$.91 \pm .01$
letter	$.55 \pm .03$	$.58 \pm .03$	$.59 \pm .02$	$.60 \pm .02$	$.61 \pm .01$	$.62 \pm .01$	$.58 \pm .01$
optdigits	$.93 \pm .01$	$.94 \pm .01$	$.94 \pm .01$	$.94 \pm .01$	$.95 \pm .01$	$.95 \pm .00$	$.94 \pm .00$
page-blocks	$.96 \pm .01$	$.96 \pm .01$	$.96 \pm .00$	$.96 \pm .00$	$.96 \pm .00$	$.96 \pm .00$	$.95 \pm .00$
pendigits	$.89 \pm .02$	$.91 \pm .02$	$.91 \pm .02$	$.92 \pm .01$	$.93 \pm .01$	$.93 \pm .01$	$.91 \pm .00$
vowel-bin	$.42 \pm .04$	$.47 \pm .04$	$.49 \pm .03$	$.51 \pm .03$	$.53 \pm .02$	$.54 \pm .02$	$.46 \pm .02$
yeast	$.52 \pm .01$	$.53 \pm .01$	$.53 \pm .01$	$.53 \pm .01$	$.54 \pm .01$	$.54 \pm .01$	$.52 \pm .01$

Table 3: Mean and standard deviation of the generation time in seconds with (a) CART and (b) logistic regression as the base learner.

(a)

dataset	Bo1	Bo5	Bo10	Bo20	Bo50	Bo100	RPND
LED24	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.09 ± .00
zoo	.00 ± .00	.00 ± .00	.00 ± .00	.01 ± .00	.01 ± .00	.01 ± .00	.02 ± .00
krkopt-b.	.39 ± .02	.41 ± .02	.42 ± .02	.43 ± .02	.44 ± .01	.45 ± .01	2.04 ± .04
segment	.04 ± .01	.04 ± .00	.04 ± .01	.04 ± .01	.04 ± .01	.05 ± .01	.07 ± .01
mfeat-mo.	.02 ± .00	.02 ± .00	.02 ± .01	.02 ± .01	.02 ± .01	.03 ± .01	.06 ± .00
mfeat-fa.	.70 ± .05	.74 ± .05	.76 ± .05	.78 ± .04	.80 ± .04	.82 ± .03	1.03 ± .03
mfeat-fo.	.51 ± .04	.54 ± .05	.55 ± .04	.56 ± .04	.58 ± .04	.59 ± .03	.69 ± .02
mfeat-ka.	.50 ± .04	.53 ± .04	.55 ± .04	.56 ± .04	.57 ± .04	.59 ± .03	.66 ± .01
mfeat-pi.	.19 ± .01	.20 ± .01	.20 ± .01	.20 ± .01	.20 ± .01	.21 ± .01	.40 ± .02
letter	.52 ± .04	.56 ± .03	.58 ± .03	.59 ± .02	.60 ± .02	.61 ± .02	3.01 ± .04
optdigits	.23 ± .01	.23 ± .01	.24 ± .01	.24 ± .01	.24 ± .01	.25 ± .01	.45 ± .03
page-bl.	.05 ± .00	.05 ± .00	.06 ± .00	.06 ± .00	.06 ± .00	.06 ± .00	.07 ± .01
pendigits	.20 ± .01	.21 ± .01	.21 ± .01	.21 ± .01	.22 ± .01	.22 ± .01	.42 ± .01
vowel-bin	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.06 ± .01
yeast	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.05 ± .00

(b)

dataset	Bo1	Bo5	Bo10	Bo20	Bo50	Bo100	RPND
LED24	.23 ± .02	.25 ± .01	.25 ± .01	.25 ± .01	.26 ± .01	.26 ± .01	.42 ± .07
zoo	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.03 ± .00	.05 ± .01
krkopt-b.	.97 ± .12	1.11 ± .13	1.17 ± .11	1.20 ± .10	1.25 ± .08	1.30 ± .07	5.46 ± 1.1
segment	.27 ± .04	.30 ± .03	.30 ± .03	.31 ± .03	.33 ± .03	.34 ± .03	.37 ± .09
mfeat-mo.	.04 ± .00	.05 ± .00	.05 ± .00	.05 ± .00	.05 ± .00	.05 ± .00	.10 ± .01
mfeat-fa.	.93 ± .08	1.02 ± .07	1.05 ± .06	1.07 ± .07	1.12 ± .11	1.18 ± .14	1.54 ± .19
mfeat-fo.	.27 ± .03	.29 ± .01	.29 ± .01	.30 ± .01	.31 ± .01	.31 ± .01	.41 ± .09
mfeat-ka.	.45 ± .04	.49 ± .03	.50 ± .03	.51 ± .03	.53 ± .02	.54 ± .02	.65 ± .10
mfeat-pi.	.68 ± .05	.74 ± .03	.76 ± .03	.76 ± .03	.79 ± .02	.80 ± .02	1.10 ± .15
letter	1.6 ± .21	1.89 ± .21	2.03 ± .19	2.11 ± .17	2.24 ± .16	2.32 ± .14	9.34 ± 1.6
optdigits	.74 ± .04	.78 ± .03	.80 ± .03	.82 ± .03	.83 ± .03	.85 ± .02	1.31 ± .14
page-bl.	.57 ± .12	.70 ± .06	.72 ± .05	.73 ± .04	.73 ± .03	.73 ± .03	.69 ± .13
pendigits	.95 ± .08	1.02 ± .07	1.05 ± .07	1.08 ± .07	1.11 ± .07	1.14 ± .05	1.70 ± .20
vowel-bin	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.06 ± .01
yeast	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.02 ± .00	.06 ± .02

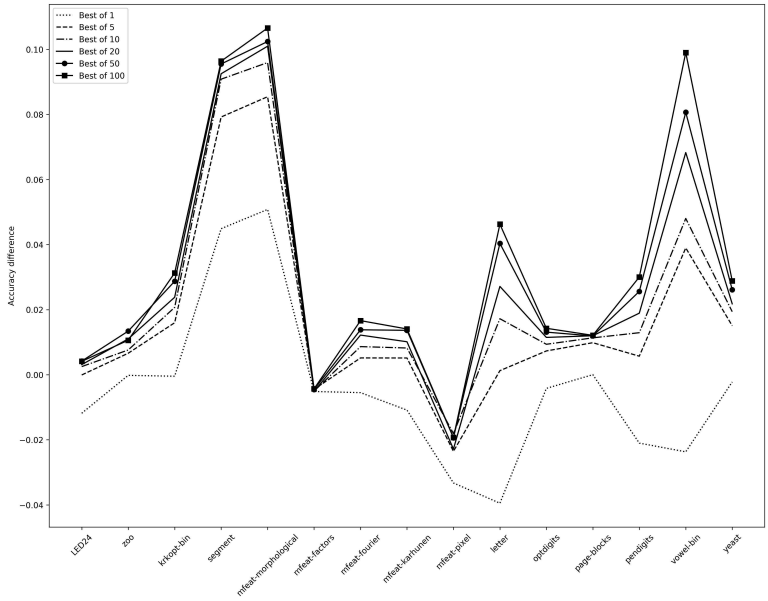

Figure 3: Difference in the mean accuracy between Best-of-K and RPND heuristic with logistic regression as the base learner.

5 Conclusion and Outlook

In this paper, we compare the state-of-the-art RPND heuristic for optimizing the structure of nested dichotomies with an extremely simple Best-of-K heuristic. To this end, an efficient algorithm for uniform sampling of nested dichotomies for a given number of classes c is also provided; the time complexity of this algorithm is $O(c)$. As the main result, we observe an increase in the predictive accuracy if logistic regression is used as a base learner. In the case of decision trees, both heuristics show similar performance.

While the focus of this study was on the optimization of a single nested dichotomy, the Best-of-K heuristic can easily be adapted for building ensembles of nested dichotomies. For example, this can be achieved by returning not only the single best performing nested dichotomy but multiple ones.

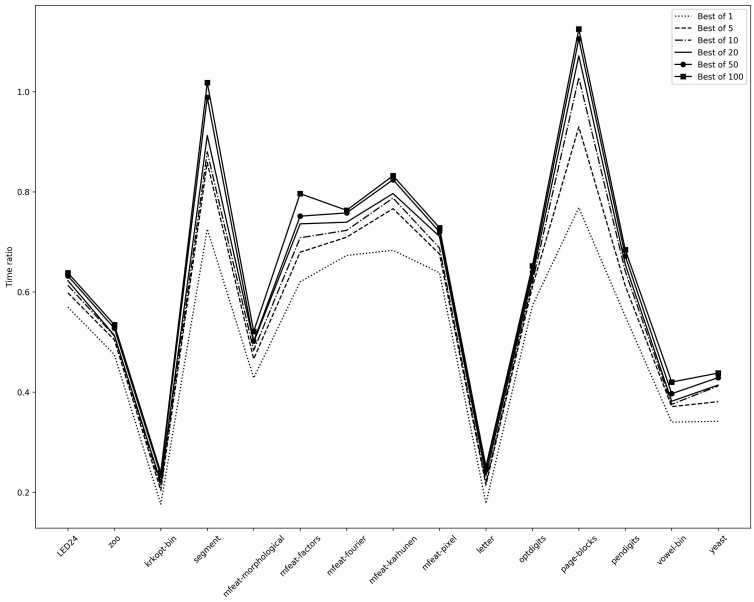

Figure 4: The ratio of mean total time (single nested dichotomy generation and training) between Best-of-K and RPND heuristic with logistic regression as base learner.

Acknowledgment

This work is part of the Collaborative Research Center "On-the-Fly Computing" at Paderborn University, which is supported by the German Research Foundation (DFG).

References

[1] D. N. A. Asuncion. UCI machine learning repository, 2007.

[2] L. Dong, E. Frank, and S. Kramer. Ensembles of balanced nested dichotomies for multi-class problems. In *Knowledge Discovery in Databases*, volume 3721 of *Lecture Notes in Computer Science*, pages 84–95. Springer, Berlin and Heidelberg and New York, 2005.

[3] M. M. Duarte-Villaseñor, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and M. Flores-Garrido. Nested dichotomies based on clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: Proceedings 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina*, pages 162–169. Springer, Berlin, Heidelberg, 2012.

[4] E. Frank and S. Kramer. Ensembles of nested dichotomies for multi-class problems. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, New York, NY, USA, 2004. ACM.

[5] G. W. Furnas. The generation of random, binary unordered trees. *Journal of Classification*, 1(1):187–233, 1984.

[6] T. Leathart, B. Pfahringer, and E. Frank. Building ensembles of adaptive nested dichotomies with random-pair selection. In *European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2016, Riva del Garda, Italy, Proceedings, Part II*, pages 179–194. Springer International Publishing, 2016.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] J. J. Rodríguez, C. García-Osorio, and J. Maudes. Forests of nested dichotomies. *Pattern Recognition Letters*, 31(2):125–132, 2010.

[9] F. J. Rohlf. Numbering binary trees with labeled terminal vertices. *Bulletin of Mathematical Biology*, 45(1):33 – 40, 1983.