On the Calibration of Nested Dichotomies

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. Nested dichotomies (NDs) are used as a method of transforming a multiclass classification problem into a series of binary problems. A tree structure is induced that recursively splits the set of classes into subsets, and a binary classification model learns to discriminate between the two subsets of classes at each node. In this paper, we demonstrate that these NDs typically exhibit poor probability calibration, even when the base binary models are well calibrated. We also show that this problem is exacerbated when the binary models are poorly calibrated. We discuss the effectiveness of different calibration strategies and show that accuracy and log-loss can be significantly improved by calibrating both the internal base models and the full ND structure, especially when the number of classes is high.

1 Introduction

As the amount of data collected online continues to grow, modern datasets utilised in machine learning are increasing in size. Not only do these datasets exhibit a large number of examples and features, but many also have a very high number of classes. It is not uncommon in some application areas to see datasets containing tens of thousands or even millions of classes [2,10].

An attractive option to handle datasets with such large label spaces is to induce a binary tree structure over the label space. At each split node k, the set of classes present, $\mathcal{C}_k$, is split into two disjoint subsets $\mathcal{C}_{k1}$ and $\mathcal{C}_{k2}$. Then, a binary classification model is trained to distinguish between these two subsets of classes. Many algorithms have been proposed that fit this general description, for example [3,5,8,9]. Often, a greedy inference approach is taken in these tree structures, i.e., test examples only take a single path from the root node to leaf nodes. This has the inherent drawback that a single mistake along the path to a leaf node results in an incorrect prediction.

In this paper, we consider methods with probabilistic classifiers at the internal nodes, called *nested dichotomies* (NDs) in the literature [15]. Utilising probabilistic binary classifiers to make routing decisions for test examples has several advantages over simply taking a hard 0/1 classification. For example, multiclass class probability estimates can be computed in a natural way by taking the product of binary probability estimates on the path from the root to the leaf node [14]. However, although hard classification decisions are avoided, small

errors in the binary probability estimates can accumulate over this product, resulting in inaccurate predictions. Datasets with more classes result in deeper trees, exacerbating this issue.

In this paper, we investigate approaches to reduce the impact of the accumulation of errors by utilising *probability calibration* techniques. Probability calibration is the task of transforming the probabilities output by a model to reflect their true empirical distribution; for the group of test examples that are predicted to belong to some class with probability 0.8, we expect about 80% of them to actually belong to that class if our model is well calibrated. Our main hypothesis is that the overall predictive performance of NDs can be improved by calibrating the individual binary models at internal nodes (which we refer to as *internal calibration*). However, we also observe that significant performance gains can be achieved by calibrating the predictions made from the entire ND (referred to as *external calibration*), even if the internal models are well calibrated.

This paper is structured as follows. First, we briefly review NDs and probability calibration. We then discuss internal and external calibration, providing theoretical motivation and showing experimental results for each method. Finally, we conclude and discuss future research directions.

2 Nested Dichotomies

NDs are used as a binary decomposition method for multiclass problems [15]. In this paper, we only consider the case where a single ND structure is built, although generally superior performance can be achieved by training an ensemble of NDs with different structures. Ensembles of NDs have been shown to outperform binary decomposition methods like one-vs-all [33], one-vs-one [16] and error-correcting output codes [12], on some classification problems [15].

The structure of an ND can have a large impact on the predictive performance, training time and prediction time. To this end, several methods have been proposed for deciding the structure of NDs [13,23,22,26,35]. In this paper, we focus on a simple method that randomly splits the class set into two at each internal node.

As previously stated, a useful feature of NDs is the ability to produce multiclass probability estimates $\hat{\mathbf{p}}_i$ for a test instance $(\mathbf{x}_i, y_i)$ from the product of binary estimates on the path $\mathcal{P}_c$ to the leaf node corresponding to class c:

$$\begin{aligned}\hat{\mathbf{p}}_i^{(c)} &= p(y_i = c|\mathbf{x}_i) \\ &= \prod_{k \in \mathcal{P}_c} \left(\mathbb{I}(c \in \mathcal{C}_{k1}) p(c \in \mathcal{C}_{k1}|\mathbf{x}_i, y_i \in \mathcal{C}_k) + \mathbb{I}(c \in \mathcal{C}_{k2}) p(c \in \mathcal{C}_{k2}|\mathbf{x}_i, y_i \in \mathcal{C}_k) \right)\end{aligned}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\mathcal{C}_k$ is the set of classes present at node k and $\mathcal{C}_{k1}, \mathcal{C}_{k2} \subset \mathcal{C}_k$ are the sets of classes present at the left and right child of node k, respectively. If desired, one can still perform greedy inference by taking the most promising branch at each split point, but this is not guaranteed to find the best solution [5]. Having binary class probability estimates facilitates efficient tree search techniques [11,20,27] for better inference, as well as top-k prediction.

3 Probability Calibration

Probability calibration is the task of transforming the outputs of a classifier to accurate probabilities. It is useful in a range of settings, such as cost-sensitive classification and scenarios where the outputs of a model are used as inputs for another. It is also important in real world decision making systems to know when a prediction from a model is likely to be incorrect.

Some models, like logistic regression, tend to be well calibrated out-of-the-box, while other models like naïve Bayes and boosted decision trees usually exhibit poor calibration, despite high classification accuracy [30]. Some other models such as support vector machines cannot output probabilities at all, but calibration can be applied to produce a probability estimate. Calibration is typically applied as a post-processing step—a calibration model is trained to transform the output score from a model into a well calibrated probability.

3.1 Calibration Methods

The most commonly used calibration methods in practice are Platt scaling (PS) [32] and isotonic regression (IR) [36]. Both of these methods are only directly applicable to binary problems, but standard multiclass transformation techniques can be used to apply them to multiclass problems [37].

Platt Scaling is a technique that fits a sigmoid curve

$$\sigma(z_i) = \frac{1}{1 + e^{\alpha z_i + \beta}}$$

from the output of a binary classifier z_i to the true labels. The parameters α and β are fitted using logistic regression. PS was originally proposed for scaling the output of SVMs, but has been shown to be an effective calibration technique for a range of models [30]. Usually, PS is applied to the log-odds (sometimes called logits) of the positive class, rather than the probability.

Matrix and Vector Scaling are simple extensions of PS for multiclass problems [17]. In matrix scaling (MS), instead of single parameters α and β, a matrix $\mathbf{W}$ and bias vector $\mathbf{b}$ are learned:

$$\boldsymbol{\sigma}(\mathbf{z}_i) = \frac{1}{1 + e^{\mathbf{W}\mathbf{z}_i + \mathbf{b}}}$$

where $\mathbf{z}_i$ is the vector of the log-odds of each class for instance i. MS is equivalent to a standard multiple logistic regression model applied to the log-odds. It is expensive for datasets with many classes, as the weight matrix $\mathbf{W}$ grows quadratically with the number of classes. Vector scaling (VS) is designed to overcome this. It is a variant where $\mathbf{W}$ is restricted to be a diagonal matrix to achieve scaling that is linear in the number of classes.

Isotonic Regression is a non-parametric technique for probability calibration [36]. It fits a piecewise constant function to minimise the mean squared error between the estimated class probabilities and the true labels. IR is a more general method than PS because no assumptions are made about the function used to map classifier outputs, other than that the function is non-decreasing (isotonicity). IR has been found to work well as a calibration model, but the flexibility of the fitted function means it can overfit on small samples.

Other related work. In this paper, efficiency is a concern as there are many models to be calibrated. For this reason, we opt for the simple calibration methods mentioned above in our experiments. However, there are several more expressive (and expensive) calibration methods in the literature. Zhong and Kwok [38] and Jiang et al. [19] propose methods for creating a smooth spline from the piecewise constant function produced in IR. Naeini et al. [29] propose a method for performing Bayesian averaging over all possible binning schemes—schemes that split the probability space into several bins and establish a calibrated probability value per bin. Leathart et al. [21] split the feature space into regions using a decision tree and build a localised calibration model in each region.

3.2 Measuring Miscalibration

The level of probability calibration that a model exhibits is frequently measured by the negative log-likelihood (NLL):

$$\text{NLL} = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i \log \hat{\mathbf{p}}_i$$

where n is the number of examples, $\mathbf{y}_i$ is the one-hot true label for an instance i and $\hat{\mathbf{p}}_i$ is the estimated probability distribution. NLL heavily penalises probability estimates that are far from the true label. For this reason, models which optimise NLL in training tend to be well calibrated, although interestingly it has been shown recently that the kinds of architectures used in modern neural networks can also produce poorly calibrated models [17]. NLL is also commonly used as a general measure of model accuracy.

Probability calibration for classification tasks can be visualised through reliability diagrams [28]. In reliability diagrams, the probability range $[0, 1]$ is discretised into K bins $B_1, \ldots, B_k$. These bins are chosen such that they have equal width, or equal numbers of examples. The *confidence* of each bin is given as the average estimated probability of examples that fall inside the bin, while the *accuracy* of each bin is the empirical accuracy:

$$\text{conf}(B_k) = \frac{1}{|B_k|}\sum_{i \in B_k} \hat{p}_i, \qquad \text{acc}(B_k) = \frac{1}{|B_k|}\sum_{i \in B_k} \mathbb{I}(\hat{y}_i = y_i)$$

where y_i is the true binary label, $\hat{y}_i$ is the predicted label, and $\hat{p}_i$ is the estimated probability for an instance i [17]. Intuitively, a well calibrated classifier

should have comparable confidence and accuracy for each bin. The confidence and accuracy are plotted against each other for each bin, producing a straight diagonal line for a well calibrated classifier.

Naeini et al. applied the same idea to give a direct quantitative measure of calibration [29], called the *expected calibration error* (ECE). This is simply a weighted average of the residuals in a reliability diagram, weighted by the number of instances that fall inside each bin.

4 Internal Calibration

In this section, we investigate the effect of calibrating the internal models of NDs. Our hypothesis is that by improving the quality of the binary probability estimates, the final multiclass predictive performance will be improved. This is due to the fact that multiclass probability estimates are produced by computing the product of a series of binary probability estimates. If the binary probability estimates are not well calibrated, then these errors will accumulate throughout the calculation.

4.1 Theoretical Motivation

It seems reasonable that improving the calibration of internal models will result in superior probability estimates for the ND, but can we theoretically quantify this improvement? It turns out that reducing the binary NLL of any internal model by some amount δ strictly reduces the multiclass NLL of the ND, and depending on the depth of the internal model being calibrated, the reduction in multiclass NLL can be as high as δ.

Proposition 1. *The NLL of an instance under an ND is equal to the sum of NLLs of the instance under the binary models on the path from the root node to the leaf node.*

Proof. The NLL of an instance i is given by

$$\text{NLL} = -\mathbf{y}_i \log \hat{\mathbf{p}}_i = -\log \hat{\mathbf{p}}_i^{(c)} \tag{1}$$

where $\hat{\mathbf{p}}_i^{(c)}$ is the probability estimate for the true class c. Let $\mathcal{P}_c$ be the set of internal nodes on the path from the root to the leaf corresponding to class c. Then, $\hat{\mathbf{p}}_i^{(c)}$ can be expressed as

$$\hat{\mathbf{p}}_i^{(c)} = \prod_{k \in \mathcal{P}_c} \tilde{y}_{ik}\hat{p}_{ik} + (1 - \tilde{y}_{ik})(1 - \hat{p}_{ik}) \tag{2}$$

where $\tilde{y}_{ik} \in \{0, 1\}$ is the binary meta-label and $\hat{p}_{ik} \in [0, 1]$ is the estimated probability of the positive meta-label for instance i for the binary model at node k respectively. Because $\tilde{y}_{ik} \in \{0, 1\}$, it is equivalent to write

$$\hat{\mathbf{p}}_i^{(c)} = \prod_{k \in \mathcal{P}_c} \hat{p}_{ik}^{\,\tilde{y}_{ik}}(1 - \hat{p}_{ik})^{(1-\tilde{y}_{ik})}. \tag{3}$$

Plugging this into (1) yields

$$\text{NLL} = -\log \prod_{k \in \mathcal{P}_c} \hat{p}_{ik}^{\tilde{y}_{ik}} (1 - \hat{p}_{ik})^{(1-\tilde{y}_{ik})} \tag{4}$$

$$= -\sum_{k \in \mathcal{P}_c} \log \left(\hat{p}_{ik}^{\tilde{y}_{ik}} (1 - \hat{p}_{ik})^{(1-\tilde{y}_{ik})} \right) \tag{5}$$

$$= -\sum_{k \in \mathcal{P}_c} \tilde{y}_{ik} \log \hat{p}_{ik} + (1 - \tilde{y}_{ik}) \log(1 - \hat{p}_{ik}), \tag{6}$$

the sum of NLLs from the models $k \in \mathcal{P}_c$. □

It directly follows that reducing the binary NLL for the model at internal node k by some amount δ results in a reduction of the multiclass NLL by δ for each class corresponding to the leaf nodes that are descendants of node k. This means that a calibration resulting in a binary NLL reduction of δ for some internal node k reduces the multiclass NLL by $\delta(n_k/n)$, where n_k is the number of examples in classes whose corresponding leaf nodes are descendants of k.

5 External Calibration

As well as calibrating each internal model, we also consider external calibration of the entire ND. Even models like logistic regression are usually not perfectly calibrated in practice. We hypothesise that these minor miscalibrations accumulate as the ND gets deeper, which can be rectified by an external calibration model. More specifically, the accumulated miscalibration is likely to be realised as *under-confident* predictions. This is because the final multiclass probability estimates are established by computing a product of probability estimates along the path from the root to the leaf. For example, consider an ND of depth six for some multiclass problem. If each binary model on the path is highly confident with a probability estimate of 0.9, the correct class will be assigned a relatively low probability estimate of $0.9^6 = 0.531$. Naturally, this effect is greater for problems with more classes, as the paths to leaf nodes will be longer.

As an illustrative investigation into the effect of ND depth on their calibration, we built an ND with logistic regression base learners for the ALOI dataset (see Tab. 1). Figure 1 shows reliability plots for versions of this ND that have been "cut-off" at incrementally increasing depths. A test example is considered to be classified correctly at depth d if its actual class is in the subset of classes $\mathcal{C}_k$ of the node k with highest probability and maximum depth d. Limited to a depth of one, the ND is simply a single binary logistic regression model, which exhibits good calibration. However, as the depth cut-off limit increases, it is clear that the ND becomes increasingly under-confident, i.e., bins that have high accuracy often have low confidence (Fig. 1, top row). This corresponds to the curve sitting above the diagonal line. The ECE increases linearly with the depth of the tree.

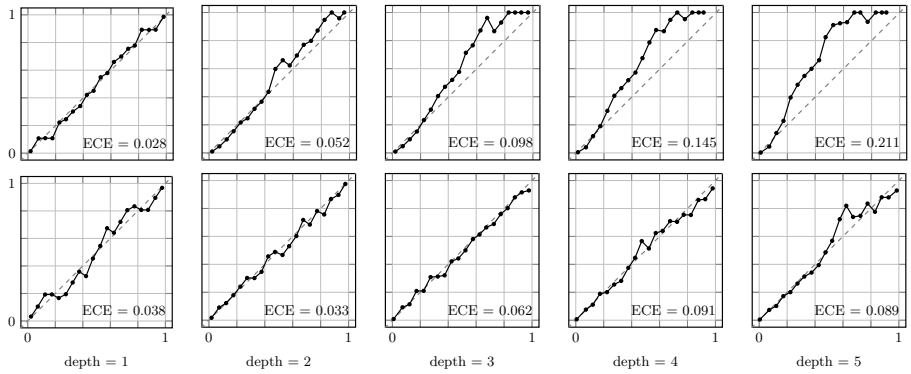

Fig. 1. Reliability plots for an ND with logistic regression base learners, cut off at increasing depth. Top: no external calibration. As the depth increases, the ND becomes increasingly under-confident because of the effect of multiplying probabilities together. Bottom: externally calibrated using vector scaling.

Table 1. Datasets used in our experiments.

Name	Instances	Features	Classes
optdigits[1]	5,620	64	10
micromass[1]	571	1,301	20
letter[1]	20,000	16	26
devanagari[1]	92,000	1,000	46

Name	Instances	Features	Classes
RCV1[2]	15,564/518,571	47,236	53
sector[2]	6,412/3,207	55,197	105
ALOI[2]	97,200/10,800	128	1,000
ILSVR2010[3]	1,111,406/150,000	1,000	1,000
ODP-5K[4]	361,488/180,744	422,712	5,000

[1] UCI Repository [24] , [2] LIBSVM Repository [7], [3] ImageNet [34], [4] ODP [4]

This is adequately and efficiently compensated for by applying VS (Fig. 1, bottom row). VS exhibits low complexity in the number of classes—only two parameters per class—making it suitable for problems with many classes typically handled by NDs. For externally calibrated NDs, the ECE initially increases linearly with the depth of the tree (although for $d > 1$, the ECE values are much lower than their uncalibrated counterparts). However, at $d = 5$, the ECE levels off and even begins to decrease slightly.

6 Experiments

In this section, we present experimental results of calibration of NDs using different base classifiers on a series of datasets. The datasets we used in our experiments are listed in Table 1, and were chosen to span a range of numbers of classes. Optdigits, letter and devanagari [1] are character recognition datasets for digits, latin letters, and Devanagari script respectively. Micromass [25] is for the identification of microorganisms from mass spectroscopy data. RCV1, sector and ODP-5K are text categorisation tasks, while ALOI and ILSVR2010 are object recognition tasks. We use the visual codewords representation for ILSVR2010.

Table 2. NLL (left) and classification accuracy (right) of NDs with logistic regression, before and after external calibration.

Dataset	Baseline	Ext. VS
optdigits	0.30 (0.07)	**0.30 (0.08)**
micromass	5.92 (1.83)	**1.88 (0.51)**
letter	1.50 (0.06)	**1.44 (0.08)**
devanagari	2.43 (0.11)	**2.03 (0.05)**
RCV1	1.00 (0.02)	**0.58 (0.01)**
sector	2.86 (0.01)	**1.25 (0.02)**
ALOI	3.60 (0.02)	**3.05 (0.03)**
ILSVR2010	6.44 (0.01)	**5.78 (0.01)**
ODP-5K	5.80 (0.01)	**4.97 (0.01)**

Dataset	Baseline	Ext. VS
optdigits	90.5 (0.02)	**90.6 (0.03)**
micromass	**80.4 (0.06)**	77.2 (0.05)
letter	51.2 (0.03)	**53.6 (0.03)**
devanagari	42.8 (0.02)	**42.8 (0.02)**
RCV1	81.4 (0.01)	**85.5 (0.00)**
sector	84.8 (0.01)	**86.7 (0.00)**
ALOI	27.4 (0.01)	**33.1 (0.01)**
ILSVR2010	**6.3 (0.00)**	5.3 (0.00)
ODP-5K	18.9 (0.00)	**22.8 (0.00)**

In order to obtain performance estimates, we performed 10 times 10-fold cross-validation for the datasets from the UCI repository, while adopting the standard train/test splits for the larger datasets with a larger number of classes ($m > 50$). The number of instances stated in Table 1 for the larger datasets is split into number of training and test instances. Note that in each fold and run of 10 times 10-fold cross-validation, a different random ND structure is constructed. In the case of the larger datasets, the average of 10 randomly constructed NDs is reported. Standard deviations are given in parentheses, and the best result per row appears in bold face. The original ODP dataset contains 105,000 classes—we took the subset of the most frequent 5,000 classes to create ODP-5K for the purposes of this investigation. We also reduce the dimensionality to 1,000 when evaluating NDs with boosted trees, by using a Gaussian random projection [6].

We implemented VS [17] and NDs in Python, and used the implementations of the base learners, IR and PS available in `scikit-learn` [31]. Our implementations are available online at `https://github.com/timleathart/pynd`.

6.1 Well Calibrated Base Learners

As shown in Figure 1, overall calibration of NDs can degrade to systematically under-confident predictions as the depth of the tree increases, even if the base learners are well calibrated. To further investigate the effects of ND depth on predictions, we performed experiments with external calibration to determine the extent to which the classification accuracy and NLL are affected as well.

Table 2 shows the NLL and accuracy of NDs with logistic regression, before and after external calibration is applied. Logistic regression models are known to be well-calibrated [30]. VS [17] is used as the external calibration model. We use a 10% sample of the training data to train the external calibration model, and the remaining 90% to build the ND including the base models.

Discussion. Table 2 shows that, for all datasets, a reduction in NLL is observed after applying external calibration with VS (Ext. VS). For some of the datasets with fewer classes (optdigits and letter), the reduction is modest, but the larger datasets see substantial improvements. Interestingly, for some datasets

a large improvement in classification accuracy is also observed, especially for the datasets with more classes. A surprising finding is that the accuracy degrades for ILSVR2010 and micromass, despite a large improvement in NLL.

6.2 Poorly Calibrated Base Learners

Tables 3 and 4 show the NLL and classification accuracy respectively of NDs trained with poorly calibrated base learners, when different calibration strategies are applied. Specifically, we considered NDs with naïve Bayes and boosted decision trees as the base learners. The calibration schemes compared are internal PS (Int. PS), internal IR (Int. IR) and external VS (Ext. VS), as well as each internal calibration scheme in conjunction with external VS (Both PS and Both IR, respectively). Three-fold cross validation is used to produce the training data for the internal calibration models, rather than splitting the training data. This is to ensure that each internal calibration model has a reasonable amount of data points to train on. When external calibration is performed, 10% of the data is held out to train the external calibration model. Note that this means 10% less data is available to train the ND and (if applicable) perform internal calibration. Gaussian naïve Bayes is applied for optdigits, micromass, letter and devanagari, and multinomial naïve Bayes is used for RCV1, sector, ALOI, ILSVR2010 and ODP-5K as they have sparse features. We use 50 decision trees with AdaBoost [18], limiting the depth of the trees to three.

Discussion. Tables 3 and 4 show that applying internal calibration is very beneficial in terms of both NLL and classification accuracy. There is no combination of base learner and dataset for which the baseline gives the best results, and there are very few cases where the baseline does not perform the worst out of every scheme. When naïve Bayes is used as the base learner, applying internal calibration with IR always gives better results than the baseline, and when an ensemble of boosted trees is used as the base learner, applying internal PS always outperforms the uncalibrated case. It is well known that these calibration methods are well-suited to the respective base learners [30], and this appears to also apply when they are used in an ND.

External calibration also has a positive effect on both NLL and classification accuracy in most cases compared to the baseline. However, the best results are usually obtained when both internal and external calibration are applied together. For naïve Bayes, the smaller datasets as well as the two object recognition datasets (ALOI and ILSVR2010) generally see the best performance for both NLL and classification accuracy when applying internal IR in conjunction with external calibration. Interestingly, the best results for the three text categorisation datasets were obtained through external calibration only.

Performing both internal PS and external calibration gives the best NLL performance for NDs with boosted trees in most cases, although the improvement compared to IR is usually small. However, performance in terms of classification accuracy is less consistent, often being greater when only calibrated internally.

Table 3. NLL of NDs with poorly calibrated base classifiers.

Model	Dataset	Baseline	Ext. VS	Int. PS	Both PS	Int. IR	Both IR
Naïve Bayes	optdigits	4.25 (0.92)	0.84 (0.13)	0.85 (0.11)	0.80 (0.09)	0.71 (0.12)	**0.64 (0.09)**
	micromass	8.66 (1.72)	1.62 (0.42)	0.92 (0.08)	0.75 (0.12)	0.76 (0.12)	**0.71 (0.15)**
	letter	2.33 (0.08)	2.15 (0.08)	2.16 (0.06)	2.06 (0.07)	2.05 (0.07)	**1.95 (0.06)**
	devanagari	13.14 (0.59)	3.31 (0.16)	2.98 (0.05)	2.60 (0.02)	2.75 (0.07)	**2.44 (0.05)**
	RCV1	1.69 (0.19)	**0.86 (0.01)**	1.14 (0.07)	0.94 (0.03)	0.99 (0.04)	0.91 (0.02)
	sector	3.79 (0.36)	**1.40 (0.09)**	2.07 (0.20)	1.51 (0.10)	1.91 (0.21)	1.77 (0.09)
	ALOI	32.9 (0.43)	6.84 (0.02)	5.53 (0.02)	4.33 (0.03)	4.85 (0.03)	**4.13 (0.01)**
	ILSVR2010	32.3 (0.20)	6.81 (0.00)	6.16 (0.00)	6.12 (0.01)	6.17 (0.00)	**6.11 (0.00)**
	ODP-5K	8.49 (0.31)	**5.16 (0.05)**	6.10 (0.00)	5.51 (0.02)	6.05 (0.11)	5.36 (0.01)
Boosted Trees	optdigits	3.86 (0.57)	0.63 (0.07)	0.40 (0.04)	**0.29 (0.04)**	0.39 (0.03)	0.30 (0.04)
	micromass	10.01 (2.05)	2.51 (0.52)	1.26 (0.11)	1.00 (0.14)	1.23 (0.27)	**0.95 (0.19)**
	letter	4.86 (0.27)	0.92 (0.04)	0.56 (0.02)	**0.44 (0.03)**	0.55 (0.02)	0.44 (0.03)
	devanagari	3.42 (0.28)	1.03 (0.04)	2.26 (0.17)	**0.71 (0.02)**	1.97 (0.12)	0.73 (0.02)
	RCV1	1.96 (0.02)	1.02 (0.00)	0.93 (0.01)	**0.71 (0.01)**	0.86 (0.00)	0.74 (0.01)
	sector	3.63 (0.20)	2.91 (0.11)	2.67 (0.03)	**2.03 (0.03)**	2.59 (0.05)	2.20 (0.07)
	ALOI	4.44 (0.26)	2.51 (0.05)	4.88 (0.03)	**1.05 (0.02)**	4.28 (0.04)	1.17 (0.03)
	ILSVR2010	6.55 (0.10)	5.86 (0.00)	5.64 (0.00)	5.21 (0.00)	5.45 (0.00)	**5.20 (0.00)**
	ODP-5K	7.73 (0.04)	7.19 (0.00)	7.12 (0.00)	6.60 (0.00)	6.98 (0.00)	**6.57 (0.00)**

Table 4. Accuracy of NDs with poorly calibrated base classifiers.

Model	Dataset	Baseline	Ext. VS	Int. PS	Both PS	Int. IR	Both IR
Naïve Bayes	optdigits	71.9 (0.05)	74.9 (0.04)	71.9 (0.05)	73.5 (0.04)	77.4 (0.04)	**79.5 (0.04)**
	micromass	74.9 (0.05)	72.4 (0.05)	77.0 (0.05)	76.2 (0.05)	**77.2 (0.05)**	75.6 (0.05)
	letter	32.9 (0.02)	36.4 (0.03)	31.8 (0.03)	36.5 (0.03)	37.6 (0.03)	**41.2 (0.03)**
	devanagari	20.2 (0.02)	22.4 (0.04)	16.7 (0.02)	26.9 (0.01)	26.5 (0.04)	**34.0 (0.01)**
	RCV1	64.4 (0.04)	**78.1 (0.00)**	69.1 (0.03)	75.6 (0.00)	73.4 (0.01)	76.5 (0.01)
	sector	33.7 (0.07)	**77.2 (0.01)**	63.3 (0.04)	73.7 (0.03)	69.2 (0.04)	69.0 (0.01)
	ALOI	2.4 (0.00)	2.9 (0.00)	1.9 (0.00)	12.4 (0.00)	9.4 (0.00)	**16.6 (0.01)**
	ILSVR2010	0.9 (0.00)	1.5 (0.00)	1.4 (0.00)	1.9 (0.00)	2.1 (0.00)	**2.6 (0.00)**
	ODP-5K	4.3 (0.01)	**21.0 (0.00)**	9.1 (0.00)	16.1 (0.00)	13.5 (0.01)	18.0 (0.00)
Boosted Trees	optdigits	88.8 (0.02)	88.3 (0.02)	**92.2 (0.01)**	91.7 (0.01)	92.1 (0.01)	91.5 (0.01)
	micromass	71.0 (0.06)	65.0 (0.06)	**74.1 (0.06)**	72.9 (0.06)	73.7 (0.06)	72.7 (0.05)
	letter	85.9 (0.01)	85.1 (0.01)	88.8 (0.01)	88.3 (0.01)	**89.0 (0.01)**	88.4 (0.01)
	devanagari	10.3 (0.06)	71.0 (0.01)	48.8 (0.11)	**79.3 (0.01)**	63.6 (0.04)	78.3 (0.01)
	RCV1	68.1 (0.06)	74.8 (0.01)	81.0 (0.00)	**81.4 (0.01)**	81.0 (0.00)	80.7 (0.00)
	sector	17.4 (0.08)	40.9 (0.02)	**60.3 (0.01)**	57.6 (0.01)	57.8 (0.01)	55.2 (0.01)
	ALOI	7.1 (0.03)	45.1 (0.01)	16.1 (0.01)	**74.3 (0.01)**	36.8 (0.01)	72.3 (0.00)
	ILSVR2010	1.9 (0.00)	4.0 (0.00)	9.3 (0.00)	**10.2 (0.00)**	9.7 (0.00)	10.0 (0.00)
	ODP-5K	3.2 (0.00)	3.8 (0.00)	5.5 (0.00)	6.8 (0.00)	6.2 (0.00)	**7.2 (0.00)**

7 Conclusion

In this paper, we show that the predictive performance of NDs can be substantially improved by applying calibration techniques. Calibrating the internal models increases the likelihood that the path to the leaf node corresponding to the true class is assigned high probability, while external calibration can correct for the systematic under-confidence exhibited by NDs. Both of these techniques have been empirically shown to provide large performance gains in terms of accuracy and NLL for a range of datasets when applied individually. Additionally,

when both internal and external calibration are applied together, the performance often improves further, especially so when the number of classes is high.

Future work in this domain includes evaluating alternative external calibration methods. In our experiments, we applied VS as it is an efficient and scalable solution for large multiclass tasks. However, when resources are available, it is possible that employing a more complex method such as matrix scaling, or IR with one-vs-rest, could provide superior results. It would also be interesting to investigate whether such calibration measures are as effective for other methods of constructing NDs than random subset selection [13,23,26,35,22]. We expect that the calibration techniques discussed in this paper will transfer to such methods.

References

1. Acharya, S., Pant, A.K., Gyawali, P.K.: Deep learning based large scale handwritten devanagari character recognition. In: SKIMA. pp. 1–6. IEEE (2015)
2. Agrawal, R., Gupta, A., Prabhu, Y., Varma, M.: Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In: WWW. pp. 13–24 (2013)
3. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: NIPS. pp. 163–171 (2010)
4. Bennett, P.N., Nguyen, N.: Refined experts: improving classification in large taxonomies. In: SIGIR. pp. 11–18. ACM (2009)
5. Beygelzimer, A., Langford, J., Ravikumar, P.: Error-correcting tournaments. In: ALT. pp. 247–262. Springer (2009)
6. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: KDD. pp. 245–250. ACM (2001)
7. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**(3), 27 (2011)
8. Choromanska, A.E., Langford, J.: Logarithmic time online multiclass prediction. In: NIPS. pp. 55–63 (2015)
9. Daumé, III, H., Karampatziakis, N., Langford, J., Mineiro, P.: Logarithmic time one-against-some. In: ICML. pp. 923–932. PMLR (2017)
10. Dekel, O., Shamir, O.: Multiclass-multilabel classification with more classes than examples. In: AISTATS. pp. 137–144. PMLR (2010)
11. Dembczyński, K., Kotłowski, W., Waegeman, W., Busa-Fekete, R., Hüllermeier, E.: Consistency of probabilistic classifier trees. In: ECMLPKDD. pp. 511–526. Springer (2016)
12. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. JAIR **2**, 263–286 (1995)
13. Dong, L., Frank, E., Kramer, S.: Ensembles of balanced nested dichotomies for multi-class problems. In: PKDD, pp. 84–95. Springer (2005)
14. Fox, J.: Applied Regression Analysis, Linear Models, and Related Methods. Sage (1997)
15. Frank, E., Kramer, S.: Ensembles of nested dichotomies for multi-class problems. In: ICML. pp. 39–46. ACM (2004)
16. Friedman, J.H.: Another approach to polychotomous classification. Technical Report, Statistics Department, Stanford University (1996)
17. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML. pp. 1321–1330. PMLR (2017)

18. Hastie, T., Rosset, S., Zhu, J., Zou, H.: Multi-class AdaBoost. Statistics and its Interface **2**(3), 349–360 (2009)
19. Jiang, X., Osl, M., Kim, J., Ohno-Machado, L.: Smooth isotonic regression: A new method to calibrate predictive models. AMIA Summits on Translational Science Proceedings **2011**, 16 (2011)
20. Kumar, A., Vembu, S., Menon, A.K., Elkan, C.: Beam search algorithms for multilabel learning. Machine Learning **92**(1), 65–89 (2013)
21. Leathart, T., Frank, E., Pfahringer, B., Holmes, G.: Probability calibration trees. In: ACML. pp. 145–160. PMLR (2017)
22. Leathart, T., Frank, E., Pfahringer, B., Holmes, G.: Ensembles of nested dichotomies with multiple subset evaluation. In: PAKDD. Springer (2019)
23. Leathart, T., Pfahringer, B., Frank, E.: Building ensembles of adaptive nested dichotomies with random-pair selection. In: ECMLPKDD. pp. 179–194. Springer (2016)
24. Lichman, M.: UCI machine learning repository (2013)
25. Mahé, P., Arsac, M., Chatellier, S., Monnin, V., Perrot, N., Mailler, S., Girard, V., Ramjeet, M., Surre, J., Lacroix, B., et al.: Automatic identification of mixed bacterial species fingerprints in a maldi-tof mass-spectrum. Bioinformatics **30**(9), 1280–1286 (2014)
26. Melnikov, V., Hüllermeier, E.: On the effectiveness of heuristics for learning nested dichotomies: an empirical analysis. Machine Learning **107**(8-10), 1–24 (2018)
27. Mena, D., Montañés, E., Quevedo, J.R., Del Coz, J.J.: Using A* for inference in probabilistic classifier chains. In: IJCAI (2015)
28. Murphy, A.H., Winkler, R.L.: Reliability of subjective probability forecasts of precipitation and temperature. Applied Statistics pp. 41–47 (1977)
29. Naeini, M., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using Bayesian binning. In: AAAI. pp. 2901–2907 (2015)
30. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: ICML. pp. 625–632. ACM (2005)
31. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. JMLR **12**(Oct), 2825–2830 (2011)
32. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers **10**(3), 61–74 (1999)
33. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. JMLR **5**, 101–141 (2004)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV **115**(3), 211–252 (2015)
35. Wever, M., Mohr, F., Hüllermeier, E.: Ensembles of evolved nested dichotomies for classification. In: GECCO. pp. 561–568. ACM (2018)
36. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: ICML. pp. 609–616. ACM (2001)
37. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: KDD. pp. 694–699. ACM (2002)
38. Zhong, W., Kwok, J.T.: Accurate probability calibration for multiple classifiers. In: IJCAI. pp. 1939–1945 (2013)