Working Paper Series
ISSN 1170-487X

ENSEMBLES OF NESTED DICHOTOMIES FOR MULTICLASS PROBLEMS

Eibe Frank and Stefan Kramer

Working Paper: 06/2004
February 2004

Ensembles of Nested Dichotomies for Multi-class Problems

Eibe Frank
Department of Computer Science
University of Waikato
Hamilton, New Zealand
eibe@cs.waikato.ac.nz

Stefan Kramer
Institut für Informatik
Technische Universität München
Munich, Germany
kramer@in.tum.de

February 25, 2004

Abstract

Nested dichotomies are a standard statistical technique for tackling certain polytomous classification problems with logistic regression. They can be represented as binary trees that recursively split a multi-class classification task into a system of dichotomies and provide a statistically sound way of applying two-class learning algorithms to multi-class problems (assuming these algorithms generate class probability estimates). However, there are usually many candidate trees for a given problem and in the standard approach the choice of a particular tree is based on domain knowledge that may not be available in practice. An alternative is to treat every system of nested dichotomies as equally likely and to form an ensemble classifier based on this assumption. We show that this approach produces more accurate classifications than applying C4.5 and logistic regression directly to multi-class problems. Our results also show that ensembles of nested dichotomies produce more accurate classifiers than pairwise classification if both techniques are used with C4.5, and comparable results for logistic regression. Compared to error-correcting output codes, they are preferable if logistic regression is used, and comparable in the case of C4.5. An additional benefit is that they generate class probability estimates. Consequently they appear to be a good general-purpose method for applying binary classifiers to multi-class problems.

1 Introduction

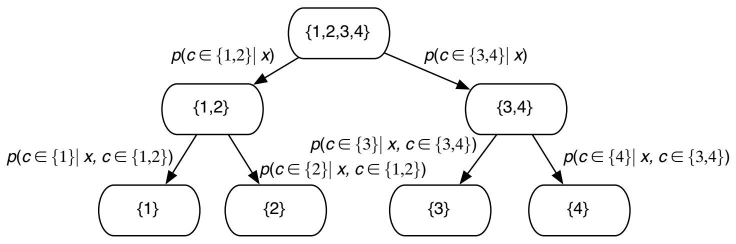
A system of nested dichotomies is a binary tree that recursively splits a set of classes from a multi-class classification problem into smaller and smaller subsets. In statistics, nested dichotomies are a standard technique for tackling polytomous (i.e. multi-class) classification problems with logistic regression by fitting binary logistic models to the individual dichotomous (i.e. two-class) classification problems at the tree's internal nodes. However, this technique is only recommended if a "particular choice of dichotomies is substantively compelling" (Fox, 1997) based on domain knowledge. There are usually many possible tree structures that can be generated for a given set of classes, and in many practical applications—namely, where the class is truly a nominal quantity and does not exhibit any structure—there is no a priori reason to prefer one particular tree structure over another one. However, in that case it makes sense to assume that every hierarchy of nested dichotomies is equally likely and to use an ensemble of these hierarchies for prediction. This is the approach we propose and evaluate in this paper.

Using C4.5 and logistic regression as base learners we show that ensembles of nested dichotomies produce more accurate classifications than applying these learners directly to multi-class problems. We also show that they compare favorably to three other popular techniques for converting a multi-class classification task into a set of binary classification problems: the simple "one-vs-rest" method, error-correcting output codes (Dietterich & Bakiri, 1995), and pairwise classification (Fürnkranz, 2002). More specifically, we show that ensembles of nested dichotomies produce more accurate classifiers than the one-vs-rest method for both C4.5 and logistic regression; that they are more accurate than pairwise classification in the case of C4.5, and comparable in the case of logistic regression; and that, compared to error-correcting output codes, nested dichotomies have a distinct edge if logistic regression is used, and are on par if C4.5 is employed. In addition, and in contrast to all three of these other popular techniques, they have the nice property that they do not require any form of post-processing to return proper probability estimates. They do have the drawback that they require the base learner to produce class probability estimates but this is not a severe limitation given that most practical learning algorithms are able to do so or can be made to do so.

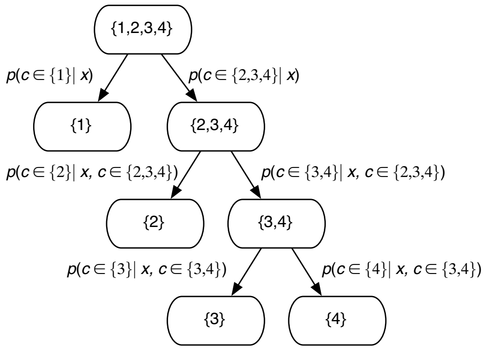
This paper is structured as follows. In Section 2 we describe more precisely how nested dichotomies work. In Section 3 we present the idea of using ensembles of nested dichotomies. In Section 4 this approach is evaluated and compared to other techniques for tackling multi-class problems. Related work is discussed in Section 5. Section 6 summarizes the main findings of this paper.

2 Nested Dichotomies

Nested dichotomies can be represented as binary trees that, at each node, divide the set of classes A associated with the node into two subsets B and C that are mutually exclusively and taken together contain all the classes in A. The nested dichotomies' root node contains all the classes of the corresponding multi-class classification problem. Each leaf node contains a single class (i.e. for an n-class problem, there are n leaf nodes and $n-1$ internal nodes). To build a classifier based on such a tree structure we do the following: at every internal node we store the instances pertaining to the classes associated with that node, and no other instances; then we group the classes pertaining to each node into two subsets, so that each subset holds the classes associated with exactly one of



(a)



(b)

Figure 1: Two different systems of nested dichotomies for a classification problem with four classes.

the node's two successor nodes; and finally we build binary classifiers for the resulting two-class problems. This process creates a tree structure with binary classifiers at the internal nodes.

We assume that the binary classifiers produce class probability estimates. For example, they could be logistic regression models. The question is how to combine the estimates from the individual two-class problems to obtain class probability estimates for the original multi-class problem. It turns out that the individual dichotomies are statistically independent because they are nested (Fox, 1997), enabling us to form multi-class probability estimates simply by multiplying together the probability estimates obtained from the two-class models. More specifically, let C_{i1} and C_{i2} be the two subsets of classes generated by a split of the set of classes C_i at internal node i of the tree (i.e. the subsets associated with the successor nodes), and let $p(c \in C_{i1}|x, c \in C_i)$ and $p(c \in C_{i2}|x, c \in C_i)$ be the conditional probability distribution estimated by the two-class model at node i for a given instance x. Then the estimated class probability distribution for the original multi-class problem is given by:

$$p(c = C|x) = \prod_{i=1}^{n-1} \quad (I(c \in C_{i1})p(c \in C_{i1}|x, c \in C_i) + \\ I(c \in C_{i2})p(c \in C_{i2}|x, c \in C_i)),$$

where $I(.)$ is the indicator function, and the product is over all the internal nodes of the tree.

Note that not all nodes have to actually be examined to compute this probability for a particular class value. Evaluating the path to the leaf associated with that class is sufficient. Let $p(c \in C_{i1}|x, c \in C_i)$ and $p(c \in C_{i2}|x, c \in C_i)$ be the labels of the edges connecting node i to the nodes associated with C_{i1} and C_{i2} respectively. Then computing $p(c|x)$ amounts to finding the single path from the root to a leaf for which c is in the set of classes associated with each node along the path, multiplying together the probability estimates encountered along the way.

Consider Figure 1, which shows two of the 15 possible nested dichotomies for a four-class classification problem. Using the tree in Figure 1a the probability of class 4 for an instance x is given by

$$\begin{aligned} p(c = 4|x) &= p(c \in \{3,4\}|x) \times \\ &\quad p(c \in \{4\}|x, c \in \{3,4\}). \end{aligned}$$

Based on the tree in Figure 1b we have

$$\begin{aligned} p(c = 4|x) &= p(c \in \{2,3,4\}|x) \times \\ &\quad p(c \in \{3,4\}|x, c \in \{2,3,4\}) \times \\ &\quad p(c \in \{4\}|x, c \in \{3,4\}). \end{aligned}$$

Both trees represent equally valid class probability estimators—like all other trees that can be generated for this problem. However, the estimates obtained from different trees will usually differ because they involve different two-class learning problems. If there is no a priori reason to prefer a particular nested dichotomy—e.g., because some classes are known to be related in some fashion—there is no reason to trust one of the estimates more than the others. Consequently it makes sense to treat all possible trees as equally likely and form overall class probability estimates by averaging the estimates obtained from different trees. This is the approach we investigate in the rest of this paper.

3 Ensembles of Nested Dichotomies

The number of possible trees for an n-class problem grows extremely quickly. It is given by the following recurrence relation:

$$T(n) = \frac{1}{2}\sum_{i=1}^{n-1}\binom{n}{i} \times [T(n-i) \times T(i)],$$

where $T(1) = 1$.

For two classes we have $T(2) = 1$, for three $T(3) = 3$, for four $T(4) = 15$, and for five $T(5) = 95$. A lower bound $T^*(n)$ for $T(n)$ is given by

$$T^*(n) = n \times T^*(n-1) = n!/2,$$

where $T^*(1) = 1$.

This means the growth in the number of trees is at least exponential, making it impossible to generate them exhaustively in a brute-force manner even for problems with a moderate number of classes. This is the case even if we cache models for the individual two-class problems that are encountered when building each tree.[1] There are $(3^n - (2^{n+1} - 1))/2$ possible two-class problems for an n-class dataset. The term 3^n arises because a class can be either in the first subset, the second one, or absent; the term $(2^{n+1} - 1)$ because we need to subtract all problems where either one of the two subsets is empty; and the factor $1/2$ from the fact that the two resulting subsets can be swapped without any effect on the classifier. Hence there are 6 possible two-class problems for a problem with 3 classes, 25 for a problem with 4 classes, 90 for a problem with 5 classes, etc.

Given these growth rates we chose to evaluate the performance of ensembles of randomly generated trees. (Of course, only the structure of each tree was generated randomly. We applied a standard learning scheme at each internal node of the randomly sampled trees.) More specifically, we sampled uniformly with replacement from the space of all distinct trees for a given n-class problem, and formed class probability estimates for a given instances x by averaging the estimates obtained from the individual ensemble members. Because of the uniform sampling process these averages form an unbiased estimate of the estimates that would have been obtained by building the complete ensemble of all possible distinct trees for a given n-class problem.

4 Empirical Comparison

We performed experiments with 21 multi-class datasets from the UCI repository (Blake & Merz, 1998), summarized in Table 1. Two learning schemes were employed: C4.5 and logistic regression.[2] We used these two because (a) they produce class probability estimates, (b) they inhabit opposite ends of the bias-variance spectrum, and (c) they can deal with multiple classes directly without having to convert a multi-class problem into a set of two-class problems (in the case of logistic regression, by optimizing the multinomial likelihood directly). The latter condition is important for testing whether any of the multi-class "wrapper" methods that we included in our experimental comparison can actually improve upon the performance of the learning schemes applied directly to the multi-class problems.

To compare the performance of the different learning schemes for each dataset, we estimated classification accuracy based on 50 runs of the stratified hold-out method, in each run using 66% of the data for training and the rest for testing. We tested for significant differences in accuracy by using the corrected resampled t-test at the 5% significance level. This test has been shown to have Type I error at the significance level and low Type II error if used in conjunction with the hold-out method (Nadeau & Bengio, 2003).

[1]Note that different trees may exhibit some two-class problems that are identical.

[2]As implemented in Weka version 3.4.1 (Witten & Frank, 2000).

Dataset	Num. insts	% Miss.	Num. atts	Nom. atts	Num. class.
anneal	898	0.0	6	32	5
arrhythmia	452	0.3	206	73	13
audiology	226	2.0	0	69	24
autos	205	1.1	15	10	6
bal.-scale	625	0.0	4	0	3
ecoli	336	0.0	7	0	8
glass	214	0.0	9	0	6
hypothyroid	3772	6.0	23	6	4
iris	150	0.0	4	0	3
letter	20000	0.0	16	0	26
lymph	148	0.0	3	15	4
optdigits	5620	0.0	64	0	10
pendigits	10992	0.0	16	0	10
prim.-tumor	339	3.9	0	17	21
segment	2310	0.0	19	0	7
soybean	683	9.8	0	35	19
splice	3190	0.0	0	61	3
vehicle	846	0.0	18	0	4
vowel	990	0.0	10	3	11
waveform	5000	0.0	40	0	3
zoo	101	0.0	1	15	7

Table 1: Datasets used for the experiments

In the first set of experiments, we compared ensembles of nested dichotomies (ENDs) with several other standard multi-class methods. In the second set we varied the number of ensemble members to see whether this has any impact on the performance of ENDs.

4.1 Comparison to other approaches for multi-class learning

In the first set of experiments we used ENDs consisting of 20 ensemble members (i.e. each classifier consisted of 20 trees of nested dichotomies) to compare to other multi-class schemes. As the experimental results in the next section will show, 20 ensemble members are often sufficient to get close to optimum performance. We used both C4.5 and logistic regression to build the ENDs. The same experiments were repeated for both standard C4.5 and polytomous logistic regression applied directly to the multi-class problems. In addition, the following other multi-class-to-binary conversion methods were compared with ENDs: one-vs-rest, pairwise classification, random error-correcting output codes, and exhaustive error-correcting output codes.

Dataset (#classes)	END	C4.5	1-vs-rest	1-vs-1	RECOCs	EECOCs
anneal (5)	98.05±0.67	98.45±0.72	97.64±0.78	97.81±0.86	98.08±0.56	98.35±0.70
arrhythmia (13)	72.91±2.36	65.37±3.09●	57.62±3.38●	66.24±3.04●	71.24±2.55	
audiology (24)	78.68±3.32	77.91±3.19	59.15±7.47●	77.03±4.11	80.49±3.68	
autos (6)	73.32±4.72	73.20±5.56	58.87±5.80●	65.79±6.29	69.18±6.34	75.09±5.06
balance-scale (3)	80.28±2.18	78.47±2.34	78.62±2.43	79.38±2.18	78.82±2.76	78.62±2.43
ecoli (8)	84.33±2.73	81.36±3.09	80.67±3.41	82.62±3.33	82.80±3.08	85.22±2.26
glass (6)	70.89±4.42	67.29±5.51	59.37±5.91●	68.77±4.72	67.10±5.08	70.95±5.06
hypothyroid (4)	99.51±0.20	99.49±0.13	99.45±0.21	99.41±0.19	99.43±0.23	99.48±0.20
iris (3)	94.04±3.17	94.12±3.19	93.92±3.18	94.12±3.19	94.00±3.20	93.92±3.18
letter (26)	94.86±0.29	86.34±0.52●	84.99±0.43●	90.10±0.36●	94.62±0.29	
lymphography (4)	77.29±5.48	76.30±4.98	76.75±5.43	77.03±5.23	75.71±4.10	76.94±5.51
optdigits (10)	97.43±0.33	89.45±0.67●	89.28±0.72●	94.01±0.55●	95.82±0.56●	98.13±0.27○
pendigits (10)	98.75±0.21	95.90±0.31●	94.77±0.39●	96.41±0.34●	98.32±0.30	99.12±0.14○
primary-tumor (21)	45.61±3.46	38.98±2.59●	26.15±3.42●	42.37±2.94	45.58±3.81	
segment (7)	97.15±0.70	95.86±0.81●	94.93±0.77●	95.90±0.75●	96.35±0.88	97.44±0.70
soybean (19)	94.16±1.38	88.75±2.14●	89.41±1.79●	92.32±1.51	93.43±1.45	
splice (3)	94.22±0.95	93.34±0.89	94.16±0.80	94.17±0.72	92.46±2.33	94.16±0.80
vehicle (4)	73.73±2.20	71.27±2.15	70.30±2.56	70.27±2.3	70.03±3.34	72.86±2.22
vowel (11)	88.57±2.34	75.82±2.59●	72.53±3.19●	75.60±3.06●	86.08±2.50	93.17±1.98○
waveform (3)	78.62±1.61	75.00±0.98●	72.49±1.18●	75.80±1.02●	72.86±1.04●	72.49±1.18●
zoo (7)	92.59±3.33	93.14±2.94	92.27±2.66	91.22±3.29	90.04±4.38	92.02±3.92

●, ○ statistically significant improvement or degradation

Table 2: Comparison of different multi-class methods for C4.5.

Dataset (#classes)	END	LR	1-vs-rest	1-vs-1	RECOCs	EECOCs
anneal (5)	99.36±0.60	98.93±0.78	98.01±1.04	99.10±0.68	98.86±0.88	99.27±0.60
arrhythmia (13)	59.28±2.72	52.76±4.06●	44.65±3.94●	60.84±3.11	47.66±4.07●	
audiology (24)	80.77±4.11	75.44±4.36	72.35±5.11●	74.64±4.15●	73.03±4.27●	
autos (6)	71.82±4.96	64.74±5.46●	57.77±5.73●	70.83±6.15	61.56±5.38●	66.31±5.14
balance-scale (3)	87.49±1.42	88.78±1.19	87.11±1.27	89.25±1.26	87.91±1.62	87.11±1.27
ecoli (8)	85.72±2.37	84.57±2.59	85.28±2.44	84.28±2.72	84.69±3.29	85.98±2.31
glass (6)	64.08±4.75	63.06±5.09	48.33±5.26●	62.29±5.59	61.09±4.20	62.73±4.31
hypothyroid (4)	96.70±0.61	96.66±0.42	95.28±0.43●	97.40±0.40	94.85±0.93●	95.42±0.40●
iris (3)	95.88±3.07	95.25±3.32	95.49±2.78	95.80±2.96	87.88±8.65	95.37±2.96
letter (26)	75.95±0.71	77.21±0.34○	72.17±0.41●	84.14±0.34○	47.51±2.32●	
lymphography (4)	78.31±5.40	77.12±6.16	76.97±5.41	78.50±6.21	76.10±5.71	76.78±5.64
optdigits (10)	96.98±0.36	93.17±0.58●	94.28±0.56●	96.96±0.33	91.68±1.13●	94.47±0.46●
pendigits (10)	95.44±0.62	95.47±0.34	93.53±0.40●	97.57±0.28○	83.74±2.33●	88.90±0.55●
primary-tumor (21)	44.48±3.24	35.56±3.79●	31.09±3.42●	38.25±3.86●	45.41±2.98	
segment (7)	94.46±0.78	95.28±0.59	92.05±0.67●	95.67±0.64○	88.60±2.58●	91.41±0.71●
soybean (19)	93.08±1.39	89.99±3.04	89.96±2.80●	90.62±1.45●	92.32±1.52	
splice (3)	92.32±1.21	89.01±1.23●	90.82±1.00	89.20±0.97●	90.37±1.66	91.66±1.00
vehicle (4)	80.07±1.75	79.27±1.97	78.62±2.11	79.15±1.81	75.92±3.99	78.93±1.80
vowel (11)	81.27±3.31	78.09±2.99	65.23±2.62●	88.42±1.86○	39.55±4.63●	46.45±2.86●
waveform (3)	86.35±0.77	86.47±0.71	86.57±0.71	86.16±0.70	83.62±2.52	86.57±0.71
zoo (7)	95.18±2.95	90.23±6.85	92.19±5.40	94.73±3.21	91.36±6.01	93.10±5.00

●, ○ statistically significant improvement or degradation

Table 3: Comparison of different multi-class methods for logistic regression.

One-vs-rest creates n dichotomies for an n-class problem, in each case learning one of the n classes against all the other classes (i.e. there is one classifier for each class). At classification time, the class that gets maximum probability from its corresponding classifier is predicted. Pairwise classification learns a classifier for each pair of classes, ignoring the instances pertaining to the other classes (i.e. there are $n \times (n-1)/2$ classifiers). A prediction is obtained by voting, where each classifier casts a vote for either one of the two classes it was built from. The class with the maximum number of votes is predicted.

In error-correcting output codes (ECOCs), each class is assigned a binary code vector of length k, which make up the row vectors of a code matrix. These row vectors determine the set of k dichotomies to be learned, corresponding to the column vectors of the code matrix. At prediction time, a vector of classifications is obtained by collecting the predictions from the individual k classifiers learned from the dichotomies. The original approach to ECOCs predicts the class whose corresponding row vector has minimum Hamming distance to the vector of 0/1 predictions obtained from the k classifiers (Dietterich & Bakiri, 1995). However, accuracy can be improved by computing the distance based on predicted class probabilities rather than 0/1 values: each 0/1 prediction is replaced by the predicted probability that the class is 1, and the distance becomes the sum of the absolute differences between the elements of the corresponding row vector and the vector of predicted probabilities. (In the case where the base learner never generates probabilities different from 0 and 1 the two approaches are identical.)

Random error-correcting output codes (RECOCs) are based on the fact that random code vectors have good error-correcting properties. We used random code vectors of length $k = 2 \times n$, where n is the number of classes.[3] Code vectors consisting only of 0s or only of 1s were discarded. This results in a code matrix with row vectors of length $2 \times n$ and column vectors of length n. Code matrices with column vectors exhibiting only 0s or only 1s were also discarded. In contrast to random codes, exhaustive error correcting codes (EECOCs) are generated deterministically. They are maximum-length code vectors of length $2^{n-1} - 1$, where the resulting dichotomies (i.e. column vectors) correspond to every possible n-bit configuration, excluding complements and vectors exhibiting only 0s or only 1s. We applied EECOCs to benchmark problems with up to 11 classes.

Table 2 shows the results obtained for C4.5 and Table 3 those obtained for logistic regression (LR). They show that ENDs produce more accurate classifications than applying C4.5 and logistic regression directly to multi-class problems. In the case of C4.5 the win/loss ratio is 18/3, in the case of logistic regression 16/5. ENDs compare even more favorably with one-vs-rest, confirming previous findings that this method is not competitive. More importantly, the experiments show that ENDs are more accurate than pairwise classification (1-vs-1) with C4.5 as base classifier (win/loss ratio: 20/1), and comparable in the case of logistic regression (win/loss ratio: 13/8).

[3]This is the default in Weka.

ENDs outperform RECOCs for both base learners: the win/loss ratio is 19/2 for both C4.5 and logistic regression. However, in the case of C4.5 only two of the differences are statistically significant, and for exhaustive codes (EECOCs) the win/loss ratio becomes 8/8 (with 3 significant wins for EECOCs and only one significant win for ENDs). In contrast, both RECOCs and EECOCs appear to be incompatible with logistic regression. Even for EECOCs the win/loss ratio is 14/2 in favor of ENDs for logistic regression (and ENDs produce five significant wins and no significant loss). We conjecture that this is due to logistic regression's inability to represent non-linear decision boundaries—an ability which may be required to adequately represent the dichotomies occurring in ECOCs. Sometimes logistic regression+ECOCs performs very poorly (see, e.g., the performance on vowel, optdigits, and pendigits). This appears to be consistent with previous findings (Dekel & Singer, 2002).

The results show that ENDs are a viable alternative to both pairwise classification and error-correcting output codes, two of the most widely-used methods for multi-class classification, and their performance appears to be less dependent on the base learner.

4.2 Effect of changing the size of the ensemble

In a second set of experiments we investigated how the performance of ENDs depends on the size of the ensemble. The results are shown in Tables 4 and 5. The first observation is that using more members never hurts performance. Also, and perhaps not surprisingly, more classes require more ensemble members. However, 20 members appear to be sufficient in most cases to obtain close-to-optimum performance. Moreover, the results show that the required ensemble size is largely independent of the learning scheme.

5 Related Work

There is an extensive body of work on using (variants of) error-correcting output codes and pairwise classification for multi-class classification. For this paper we used error-correcting codes that can be represented as bit vectors. Allwein et al. (2000) introduced extended codes with "don't care" values (in addition to 0s and 1s), but they did not observe an improvement in performance over binary codes. Interestingly, the learning problems occurring in nested dichotomies can be represented using these extended codes. For example, Table 6 shows the code vectors corresponding to the tree from Figure 1a (where "X" stands for a "don't care"). However, the "decoding process" used in ensembles of nested dichotomies is quite different and has the advantage that it generates class probability estimates.

Other approaches on improving ECOCs are based on adapting the code vectors during or after the learning process. Crammer and Singer (2001) present a quadratic programming algorithm for post-processing the code vectors and show some theoretical properties of this algorithm. Dekel and Singer (2002)

Dataset (#classes)	20 members	1 member	5 members	10 members	40 members
anneal (5)	98.05±0.67	97.99±0.68	98.09±0.76	98.13±0.69	98.13±0.70
arrhythmia (13)	72.91±2.36	63.22±3.56●	70.18±2.82	71.97±2.20	73.39±2.28
audiology (24)	78.68±3.32	72.75±4.81	77.25±3.96	78.68±3.48	79.33±3.17
autos (6)	73.32±4.72	66.64±6.50	71.25±5.54	72.60±4.50	73.45±4.92
balance-scale (3)	80.28±2.18	80.50±2.01	80.27±1.97	80.00±2.05	80.23±2.07
ecoli (8)	84.33±2.73	81.33±3.12	83.41±2.87	83.91±2.89	84.79±2.86
glass (6)	70.89±4.42	65.32±5.14	68.67±5.00	69.93±4.00	71.49±4.51
hypothyroid (4)	99.51±0.20	99.45±0.22	99.49±0.20	99.50±0.20	99.51±0.19
iris (3)	94.04±3.17	94.16±3.25	94.12±3.24	94.08±3.21	94.00±3.15
letter (26)	94.86±0.29	83.30±0.66●	91.77±0.35●	93.76±0.29●	95.39±0.22○
lymphography (4)	77.29±5.48	76.77±4.93	77.35±5.75	76.98±5.78	77.58±5.47
optdigits (10)	97.43±0.33	88.40±1.35●	95.74±0.52●	96.92±0.45●	97.67±0.32
pendigits (10)	98.75±0.21	94.80±0.53●	98.03±0.22●	98.54±0.23●	98.86±0.21
primary-tumor (21)	45.61±3.46	39.84±3.57●	43.90±2.91	45.40±3.08	45.31±3.29
segment (7)	97.15±0.70	95.05±0.81●	96.74±0.72	97.03±0.74	97.29±0.69
soybean (19)	94.16±1.38	89.31±2.60●	93.52±1.70	93.87±1.46	94.28±1.17
splice (3)	94.22±0.95	92.68±1.60	93.66±1.20	94.17±0.97	94.43±0.76
vehicle (4)	73.73±2.20	69.96±2.33	72.81±2.27	73.56±2.24	73.88±2.02
vowel (11)	88.57±2.34	70.22±3.48●	83.05±2.84●	86.76±2.43	89.62±2.21
waveform (3)	78.62±1.61	75.43±0.89●	77.22±2.08	77.77±1.77	79.18±1.31
zoo (7)	92.59±3.33	89.59±4.58	91.65±3.31	92.30±3.22	93.33±3.02

●, ○ statistically significant improvement or degradation

Table 4: Comparison of different numbers of ensemble members for C4.5.

Dataset (#classes)	20 members	1 member	5 members	10 members	40 members
anneal (5)	99.36±0.60	98.97±0.93	99.25±0.57	99.31±0.60	99.41±0.55
arrhythmia (13)	59.28±2.72	46.85±4.47 ●	54.39±3.59●	57.96±2.49	59.93±2.93
audiology (24)	80.77±4.11	66.35±5.99 ●	77.75±4.64	79.36±4.27	81.38±3.83
autos (6)	71.82±4.96	62.24±6.82 ●	68.12±5.20	70.17±5.59	72.07±5.25
balance-scale (3)	87.49±1.42	88.76±1.62	87.93±1.51	87.57±1.41	87.22±1.25
ecoli (8)	85.72±2.37	82.34±3.12	85.17±2.53	85.82±2.37	85.86±2.43
glass (6)	64.08±4.75	60.50±5.04	63.23±4.82	63.70±4.48	64.29±4.51
hypothyroid (4)	96.70±0.61	96.39±1.52	96.47±1.19	96.71±0.72	96.73±0.57
iris (3)	95.88±3.07	87.73±10.73	94.71±4.16	95.61±3.60	95.88±2.89
letter (26)	75.95±0.71	48.48±3.75 ●	69.55±1.50●	73.77±1.12●	77.02±0.52○
lymphography (4)	78.31±5.40	77.03±6.07	77.28±6.44	77.95±5.90	78.07±5.64
optdigits (10)	96.98±0.36	87.51±2.81 ●	95.59±0.63●	96.53±0.44●	97.17±0.37
pendigits (10)	95.44±0.62	83.21±4.31 ●	93.67±1.05●	94.81±0.86	95.67±0.49
primary-tumor (21)	44.48±3.24	37.07±3.55 ●	41.89±3.27	43.50±3.12	44.98±3.35
segment (7)	94.46±0.78	87.96±3.82 ●	93.39±1.47	93.83±1.23	94.64±0.62
soybean (19)	93.08±1.39	87.89±2.80 ●	92.44±1.48	92.95±1.57	93.12±1.36
splice (3)	92.32±1.21	89.57±0.92 ●	91.70±1.31	92.12±1.24	92.64±0.95
vehicle (4)	80.07±1.75	75.71±3.95	79.38±2.12	79.74±1.88	80.07±1.76
vowel (11)	81.27±3.31	49.67±8.55 ●	71.59±4.51●	78.28±3.87	83.24±2.72
waveform (3)	86.35±0.77	83.21±1.14 ●	86.10±0.90	86.34±0.76	86.38±0.73
zoo (7)	95.18±2.95	90.70±5.35	92.95±4.39	94.25±3.74	95.99±2.77

●, ○ statistically significant improvement or degradation

Table 5: Comparison of different numbers of ensemble members for logistic regression.

	c_1	c_2	c_3
1	1	1	X
2	1	0	X
3	0	X	1
4	0	X	0

Table 6: Code vectors for the tree in Figure 1a.

describe an iterative algorithm called "Bunching" that adapts the code vectors during the learning process, and show that it improves performance for the case of logistic regression. Along similar lines, Rätsch et al. (2002) propose an algorithm for adaptive ECOCs and present some preliminary results.

There is also some work on generating probability estimates based on ECOCs and pairwise classification. Kong and Dietterich (1997) introduce a post-processing step for ECOCs that recovers probability estimates. However, this step only finds an approximate solution because the underlying problem is over-constrained. Similarly, Hastie and Tibshirani (1998) proposed a method called "pairwise coupling" as a post-processing step for pairwise classification. Again, the problem is over-constrained but an approximate solution can be given, and this work has recently been extended by Wu et al.(2003).

Platt et al. (2000) show that the $n \times (n-1)/2$ classifiers in pairwise classification can be arranged into a directed acyclic graph (DAG), where each node represents a model discriminating between two classes: if we discriminate between two classes A and B at an inner node, then we just conclude that it is *not* class A if the model decides for B and vice versa. In the leaves, after excluding all classes except two, a final decision is taken. Compared to voting, this process improves classification time and does not appear to negatively affect accuracy.

Finally, in a very recent paper, Rifkin and Klautau (2004) claim that the one-vs-rest method works as well as pairwise classification and error-correcting output codes if "the underlying binary classifiers are well-tuned regularized classifiers such as support vector machines". Hence it may be possible to improve the poor performance of one-vs-rest that we observed in our experiments by optimizing the pruning parameter in C4.5 and using a carefully tuned shrinkage parameter in logistic regression.

6 Conclusions

In this paper we introduced a new, general-purpose method for reducing multi-class problems to a set of binary classification tasks, based on ensembles of nested dichotomies (ENDs). The method requires binary classifiers that are able to provide class probability estimates and in turn returns class probability estimates for the original multi-class problem. Our experimental results show that ENDs are a promising alternative to both pairwise classification and error-correcting output codes; in particular, and in contrast to both these other methods, they appear to significantly improve classification accuracy independent of which base learner is used. As future work, we plan to investigate deterministic methods for generating ENDs and the use of ENDs for ordinal classification problems.

References

Allwein, E., Schapire, R. & Singer, Y. (2000). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, *1*, 113–141.

Blake, C. & Merz, C. (1998). UCI repository of machine learning databases. University of California, Irvine, Dept. of Inf. and Computer Science. [www.ics.uci.edu/∼mlearn/MLRepository.html].

Crammer, K. & Singer, Y. (2001). On the learnability and design of output codes for multiclass problems. *Machine Learning*, *47(2/3)*, 201–234.

Dekel, O. & Singer, Y. (2002). Multiclass learning by probabilistic embeddings. In *Advances in Neural Information Processing Systems 15* (pp. 945–952). MIT Press.

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

Fox, J. (1997). *Applied Regression Analysis, Linear Models, and Related Methods*. Sage.

Fürnkranz, J. (2002). Round robin classification. *Journal of Machine Learning Research*, *2*, 721–747.

Hastie, T. & Tibshirani, R. (1998). Classification by pairwise coupling. *Annals of Statistics*, *26(2)*, 451–471.

Kong, E. & Dietterich, T. (1997). Probability estimation using error-correcting output coding. In *Proceedings of the IASTED International Conference: Artificial Intelligence and Soft Computing*. ACTA Press.

Nadeau, C. & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*, 239–281.

Platt, J., Cristianini, N. & Shawe-Taylor, J. (2000). Large margin DAGS for multiclass classification. In *Advances in Neural Information Processing Systems 12* (pp. 547–553). MIT Press.

Rätsch, G., Mika, S. & Smola, A. (2002). Adapting codes and embeddings for polychotomies. In *Advances in Neural Information Processing Systems 15* (pp. 513–520). MIT Press.

Rifkin, R. & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, *5*, 101–141.

Witten, I. H. & Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

Wu, T.-F., Lin, C.-J. & Weng, R. C. (2003). Probability estimates for multi-class classification by pairwise coupling. In *Advances in Neural Information Processing Systems 16*. MIT press.