Data Min Knowl Disc
DOI 10.1007/s10618-011-0222-1

Hellinger distance decision trees are robust and skew-insensitive

David A. Cieslak · T. Ryan Hoens · Nitesh V. Chawla · W. Philip Kegelmeyer

Received: 29 December 2010 / Accepted: 12 May 2011

Abstract Learning from imbalanced data is an important and common problem. Decision trees, supplemented with sampling techniques, have proven to be an effective way to address the imbalanced data problem. Despite their effectiveness, however, sampling methods add complexity and the need for parameter selection. To bypass these difficulties we propose a new decision tree technique called Hellinger Distance Decision Trees (HDDT) which uses Hellinger distance as the splitting criterion. We analytically and empirically demonstrate the strong skew insensitivity of Hellinger distance and its advantages over popular alternatives such as entropy (gain ratio). We apply a comprehensive empirical evaluation framework testing against commonly used sampling and ensemble methods, considering performance across 58 varied datasets. We demonstrate the superiority (using robust tests of statistical significance) of HDDT on imbalanced data, as well as its competitive performance on balanced datasets. We thereby arrive at the particularly practical conclusion that for imbalanced data it is sufficient to use Hellinger trees with bagging (BG) without any sampling methods. We provide all the datasets and software for this paper online (http://www.nd.edu/~dial/hddt).

Keywords Imbalanced data · Decision tree · Hellinger distance

Responsible editor: Johannes Fürnkranz.

D. A. Cieslak · T. R. Hoens · N. V. Chawla (✉) · W. P. Kegelmeyer
University of Notre Dame, Notre Dame, IN, USA
e-mail: nchawla@cse.nd.edu

Published online: 09 June 2011

1 Introduction

Decision trees are among the more popular classification methods, primarily due to their efficiency, simplicity, and interpretability. While individual trees can be limited in their expressiveness due to using only axis-parallel splits, this shortcoming can be mitigated by using an ensemble of decision trees as they have demonstrated statistically significant improvements over a single decision tree classifier (Breiman 1996, 2001; Freund and Schapire 1996; Banfield et al. 2007). When demonstrating the success of decision trees, however, most of the work has focused on relatively balanced datasets. Exacerbating this oversight, previous work has demonstrated the innate weakness of the traditional decision tree splitting criteria (i.e., entropy and Gini) when datasets have high degrees of class imbalance (Flach 2003; Drummond and Holte 2000; Cieslak and Chawla 2008a) due to their sensitivity to skew.

One of the weaknesses of decision trees is dealing with imbalanced datasets. A dataset is considered "imbalanced" if one class (the majority class) vastly outnumbers the other (minority class) in the training data[1] (Chawla et al. 2004). Due to the nature of learning algorithms, class imbalance is often a major challenge as it impedes the ability of classifiers to learn the minority class concept. This is due to the fact that when learning under highly imbalanced training data, classifying all instances as negative will result in high classification accuracy.

To overcome the class imbalance problem, sampling methods have become the de facto standard for improving the performance of these decision tree algorithms (Japkowicz 2000; Kubat and Matwin 1997; Batista et al. 2004; Van Hulse et al. 2007; Chawla et al. 2002, 2008; Cieslak and Chawla 2008b). Although successful, they add an additional—and sometimes awkward—responsibility for determining the sampling parameters. We are therefore motivated to ask: *Can we improve the performance of decision trees on highly imbalanced datasets without using sampling*?

In response to this question we previously proposed Hellinger distance as a decision tree splitting criterion to build Hellinger distance decision trees (HDDT) (Cieslak and Chawla 2008a). We compared this method, in single trees, to C4.4 (gain ratio) and CART (Gini), however since CART demonstrated consistently inferior performance to both other algorithms, we omit it here.

We would like to note that we use C4.4 (Provost and Domingos 2003)—unpruned, and uncollapsed C4.5 with Laplace smoothing at the leaves—as opposed to traditional C4.5; that is, C4.4 is C4.5 with slightly modified default parameters. While the decision to modify the default parameters is a popular choice in the community when applying to imbalanced data, most people still use the term C4.5 (Zadrozny and Elkan 2001; Drummond and Holte 2003; Chawla 2003). As we believe the term C4.4 helps disambiguate the two learning methods, we adopt the terminology and recommend its use.

The use of C4.4 instead of C4.5 is supported by prior research demonstrating that C4.4 results in improved class probability estimates (Chawla 2003) and is more apt

[1] Note: By convention, the negative class is the majority class, and positive class is the minority class.

for highly imbalanced datasets. In order to ensure fair comparisons, we also build uncollapsed, unpruned HDDTs with Laplace smoothing.

In this paper we extend the comparative analysis between Hellinger distance and gain ratio as decision tree splitting criteria, investigate its effectiveness in ensembles of trees, and further demonstrate the robustness of Hellinger distance to high degrees of class imbalance We also include a number of ensemble and sampling methods to arrive at a compelling conclusion: *we recommend bagged HDDTs as the preferred method for dealing with imbalanced datasets when using decision trees.* This conclusion is supported by (to the best of our knowledge) one of the most comprehensive experimental studies on decision trees for imbalanced datasets to date. This is not only in terms of datasets considered (a total of 58) but also in the techniques applied (HDDT, C4.4, two bagging (BG) variants, boosting, synthetic minority oversampling technique (SMOTE), and several combinations of techniques) (Cieslak and Chawla 2008a; Freund and Schapire 1996; Ho 1998; Breiman 2001; Dietterich 2000; Banfield et al. 2007). Our conclusions are supported by comparing the performance of these different classifiers using robust statistical significance tests (Demšar 2006; Dietterich 1998).

In summary, the key contributions of this paper are:

(1) Expand on the analysis of Hellinger distance as a decision tree splitting criterion to establish the robustness and skew insensitivity first presented in Cieslak and Chawla (2008a) (Sect. 2). The HDDT algorithm is presented in Sect. 3.

(2) Empirical evaluation and analysis of the performance of HDDT and C4.4 under a comprehensive framework over a variety of measures: single trees versus ensembles, both with and without sampling. A number of different ensemble methods—BG, boosting, majority bagging (MB)—are used. The sampling methods considered in this paper include SMOTE and undersampling. The sampling amounts are determined via the wrapper method from Chawla et al. (2008) (Sect. 4).

The analysis is broken into three parts. Part one includes only binary class imbalanced datasets. Part two includes multiple class datasets with different proportions of imbalance across the classes. In both parts we consider area under the receiver operating characteristic (ROC) curve (AUC) and F_1-measure as the performance criteria. Finally, part three is comprised of balanced datasets in order to evaluate and compare HDDT versus C4.5 for relatively balanced class distributions with standard overall accuracy as the performance measure (note that we use the original C4.5 with balanced datasets, which is a standard). A total of 58 datasets are used in this paper (as compared to only 19 binary class datasets in our prior work (Cieslak and Chawla 2008a)).

(3) Establish HDDT as a general decision tree algorithm broadly applicable to both imbalanced and balanced datasets, achieving statistically significantly superior performance over C4.4 for imbalanced datasets and comparable performance (neither significantly better nor worse) to C4.4 for balanced datasets. We also show that HDDTs, when used with BG or boosting, remove the need of sampling methods, which is a big jump forward for learning decision trees for imbalanced data.

2 Hellinger distance as a splitting criterion

Hellinger distance is a measure of distributional divergence (Kailath 1967; Rao 1995 which was first applied as a decision tree splitting criterion in Cieslak and Chawla (2008a). Let (Ω,B,ν) be a measure space (Halmos 1950), where P is the set of all probability measures on B that are absolutely continuous with respect to ν. Consider two probability measures P_1, $P_2 \in P$. The Bhattacharyya coefficient between P_1 and P_2 is defined as:

$$p(P_1, P_2) = \int_{\Omega} \sqrt{\frac{dP_1}{d\nu} \cdot \frac{dP_2}{d\nu}} d\nu. \qquad (1)$$

The Hellinger distance is derived using the Bhattacharyya coefficient as:

$$h_H(P_1, P_2) = 2\left[1 - \int_{\Omega} \sqrt{\frac{dP_1}{d\nu} \cdot \frac{dP_2}{d\nu}} d\nu\right] = \sqrt{\int_{\Omega} \left(\sqrt{\frac{dP_1}{d\nu}} - \sqrt{\frac{dP_2}{d\nu}}\right)^2 d\nu}. \qquad (2)$$

Within machine learning, we typically compare conditional probabilities stemming from discrete counts of data, rather than continuous functions. The information available may often be expressed as $P(Y = y|X = x)$ (which we abbreviate to $P(Y_y|X_x)$) where y is drawn from some finite set of classes like $+, -$ and x is drawn from a finite set of attribute values V such as $\{red, blue, green\}$. In the case of continuous features, a variety of splits are investigated and the set of such values becomes $\{left, right\}$. Since we are interested in evaluating over a countable rather than continuous space, we may convert the integral in Eq. 2 to a summation of all values and reexpress our distributions within the context of the above conditional probability as:

$$d_H(P(Y_+), P(Y_-)) = \sqrt{\sum_{i \in V} \left(\sqrt{P(Y_+|X_i)} - \sqrt{P(Y_-|X_i)}\right)^2}. \qquad (3)$$

This presents a distance which quantifies the separability of two classes of data conditioned over the full set of feature values. (As an aside, we note a strong relationship between this metric and confidence-rated boosting (Schapire and Singer 1999).) This lends itself as a decision tree splitting criterion with the following properties:

(1) $d_H(P(Y_+), P(Y_-))$ is bounded in $[0, \sqrt{2}]$
(2) $d_H(\cdot, \cdot)$ is symmetric and non-negative, i.e.,
$d_H(P(Y_+), P(Y_-)) = d_H(P(Y_-), P(Y_+)) \geq 0$
(3) squared Hellinger distance is the lower bound of KL divergence (Nguyen et al. 2007).

One contribution of this paper is to demonstrate the skew insensitivity of Hellinger distance (Sect. 2.1). As can be seen from Eqs. 2 and 3, class priors do not influence the Hellinger distance calculation, indicating a degree of skew insensitivity. Also, it essentially captures the divergence between the feature value distributions, given the different classes. We will further study how it manages skew in the next section.

2.1 Skew insensitivity

In our prior work Cieslak and Chawla (2008a), we demonstrated the skew insensitivity of Hellinger distance as a decision tree splitting criterion by considering the shape of the function. In this section we will revisit these considerations, and then extend this analysis by demonstrating the effects of skew in a synthetic example.

2.1.1 Comparing isometrics

Vilalta and Oblinger (2000) proposed the use of isometric lines to define the bias of an evaluation measure by plotting contours for a given measure over the range of possible values. In their paper they presented a case study on information gain, and while they did not produce isometrics under class skew, they note that "A highly skewed distribution may lead to the conclusion that two measures yield similar generalization effects, when in fact a significant difference could be detected under equal class distribution (Vilalta and Oblinger 2000)." Subsequently Flach (2003) connected the isometric plots to ROC analysis, demonstrating the effects of true positives (TP) and false positives (FP) on several common evaluation measures: accuracy, precision, and F-measure. In addition, he also presented isometrics for three major decision tree splitting criteria: entropy (used in gain ratio) (Quinlan 1986), Gini index (Breiman et al. 1984), and DKM (Dietterich et al. 1996). Flach also established the effect of class skew on the shape of these isometrics (Flach 2003).

This can be extended to Hellinger distance as follows:

$$d_H(tpr, fpr) = \sqrt{\left(\sqrt{tpr} - \sqrt{fpr}\right)^2 + \left(\sqrt{1-tpr} - \sqrt{1-fpr}\right)^2} \tag{4}$$

We adopt the formulation of Flach in this paper; that is, the isometric plots show the contour lines in 2D ROC space representative of the performance of different decision tree splitting criteria with respect to their estimated TP and FP rates, conditioned on the skew ratio ($c = neg/pos$). A decision tree split—for the binary class problem—can be defined by a confusion matrix as follows. A parent node will have POS positive examples and NEG negative examples. Assuming a binary split, one child will carry the TP and FP instances, and the other child will carry the true negatives (TN) and false negatives (FN) instances. The different decision tree splitting criteria, as considered in this paper, can then be modeled after this impurity (distribution of positives and negatives). Thus, in the isometric plots, each contour represents the combinations of TP and FN negatives that will generate a particular value for a given decision tree splitting criterion. For example, the 0.1 contour in Fig. 1a

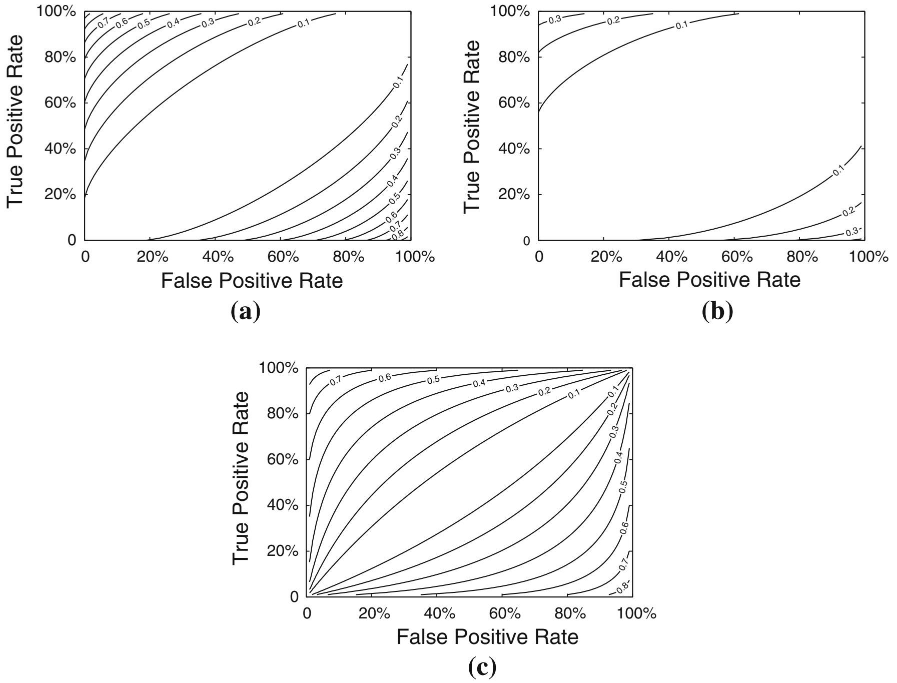

Fig. 1 Isometrics for information gain and Hellinger distance over a variety of class skews. **a** Information gain isometrics for an imbalance ratio of (1:1). **b** Information gain isometrics for an imbalance ratio of (1:10). **c** Hellinger distance isometric for any imbalance ratio

indicates that the value of information gain[2] is 0.1 at (fpr, tpr) of approximately $(0, 20\%)$, $(20, 60\%)$, $(80, 100\%)$, $(20, 0\%)$, $(60, 20\%)$, $(100, 80\%)$, and all other combinations along the contour. In Fig. 1a and b, information gain is observed as contours formed in ROC space under a $(+ : -)$ skew of $(1 : 1)$ and $(1 : 10)$, respectively. As the skew increases, the isometrics become flatter and information gain will operate more poorly as a splitting criterion. Vilalta and Oblinger (2000) and Flach (2003) observed similar trends. Note that we only considered the two class proportions of $(1 : 1)$ and $(1 : 10)$ to highlight the impact of even a marginal class skew. We point the interested reader to the paper by Flach for a more elaborate analysis of class skew using isometrics on these three metrics (Flach 2003).

Given the nature of information gain's isometric plots, we now turn our attention to Hellinger distance. First, using Flach's model of relative impurity allowed us to derive Equation 4 as an extension to Hellinger distance. In Fig. 1c, we see the isometric plots for Hellinger distance. While information gain showed dependence on skew in its isometric plots, we note that the Hellinger distance isometric plots do not deviate from the contours with varying class skew (c). This is due to the fact that there is no factor of c

[2] Note that for these plots show information gain instead of gain ratio. The choice of information gain over gain ratio is merely for consistency with Flach (2003), however, as gain ratio and information gain are equivalent over binary splits.

in the relative impurity formulation. The isometric contours for Hellinger distance are therefore unaffected by an increase in the class skew rate, making Hellinger distance much more robust in the presence of skew.

2.1.2 Synthetic example

Given the analytic results from the previous section, we now wish to gain a more intuitive understanding of the potential impact of selecting between Hellinger distance and gain ratio as a decision tree splitting criteria. Consider an artificially created dataset with two classes generated by separate Gaussian distributions of equal standard deviation with means separated by 2.5 standard deviations. In our first scenario, we simulate the effects of two equal distributions by generating 10,000 examples per class (the experiment for each class distribution is repeated over 1,000 repetitions to ensure robustness against random noise). For each repetition, we calculate the splits which are empirically chosen by C4.4, and HDDT, as well as the split which maximizes AUC (see Sect. 5.2 for more details), and determine the average for each. This experiment is illustrated in Fig. 2a, with each vertical line indicating the average for a particular optimized split and with the error bars representing one standard deviation for each split. We note that the error bars for all three ideal splits overlap the Bayesian optimal split (where error is minimized) i.e., where the two distributions intersect. Thus, when data is balanced we expect HDDT to perform similarly to C4.4 when determining both accuracy and AUC, a result confirmed later in this report. We note that the AUC boundary is also the boundary for F-measure in this setting.

In Fig. 2b we introduce a 2:1 class imbalance by sampling only 5000 points from the left distribution. We note the error bars for C4.4's split, HDDTs split, and AUC all overlap with each other, although not with that of the Bayesian optimal. This indicates that these splitting measures may not be ideal for determining accuracy, but should be theoretically optimal for AUC. We further increase class skew in Fig. 2c to a ratio of 10:1. Here we begin to notice some separation between the splits of C4.4 and HDDT, as their error bars no longer overlap. The HDDT split region intersects with that of AUC, but not of accuracy. C4.4's split region, on the other hand, intersects with neither AUC nor the Bayesian optimal. Finally, we present Fig. 2d, which exhibits a class imbalance ratio of 100:1. C4.4 once again chooses a split region which overlaps neither the Bayesian optimal split nor the AUC split, while HDDTs split and AUC again overlap. This suggests that at levels of extreme imbalance, HDDTs can be expected to produce trees with better AUC than C4.4, and that C4.4 does not not choose ideal splits for AUC or accuracy. This conclusion is supported by observations in Cieslak and Chawla (2008a), which note that whereas the possible value continuum for C4.4 is influenced by relative class balance, the same continuum is immutable for HDDT through all possible imbalance ratios.

3 HDDT: Hellinger distance decision tree

Algorithms 1 and 2 outline how Hellinger distance is incorporated into learning decision trees. We will refer to Hellinger distance and Hellinger distance based decision

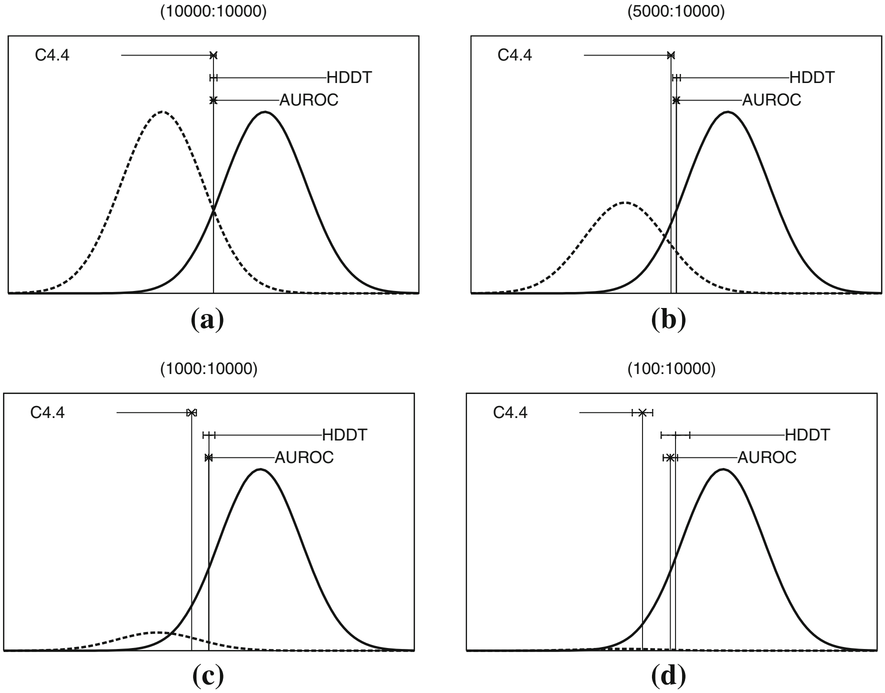

Fig. 2 Comparison of the effects of various class distributions on the ability of gain ratio and Hellinger distance to correctly determine the class boundary which optimizes AUC. Note that the Bayesian optimal split is located where the two curves intersect. **a** Synthetic example with a balanced class distribution. **b** Synthetic example with a 2:1 class distribution. **c** Synthetic example with a 10:1 class distribution. **d** Synthetic example with a 100:1 class distribution

trees as HDDT for the remainder of the paper. In our algorithm, T_i indicates the subset of training set T which has all class i instances, $T_{x_k=j}$ specifies the subset with value j for feature k, and $T_{k,j,i}$ identifies the subset with class i and has value j for feature k.

Note that Algorithm 1 is slightly different than the original definition of the Hellinger splitting criterion, in that it recommends binary splits for nominal attributes. This is due to the fact that, empirically, Hellinger distance performs better on highly

Algorithm 1 *Calc_Binary_Hellinger*

```
Require: Training set $T$, Feature $f$
Let Hellinger $\leftarrow -1$.
Let $V_f$ be the set of values of feature $f$.
for each value $v \in V_f$ do
    Let $w \leftarrow V_f \setminus v$
    cur_value $\leftarrow (\sqrt{|T_{f,v,+}|/|T_+|} - \sqrt{|T_{f,v,-}|/|T_-|})^2 + (\sqrt{|T_{f,w,+}|/|T_+|} - \sqrt{|T_{f,w,-}|/|T_-|})^2$
    if $cur\_value >$ Hellinger then
        Hellinger $\leftarrow cur\_value$
    end if
end for
return $\sqrt{\text{Hellinger}}$
```

branching nominal attributes with this restriction and no simple extension (similar to gain ratio vs. information gain) exists. In the case that a given feature is continuous, a slight variant to Algorithm 1 is used in which *Calc_Binary_Hellinger* sorts based on the feature value, finds all meaningful splits, calculates the binary Hellinger distance at each split, and returns the highest distance; this is identical to the methodology used by C4.5 (and, by extension, C4.4). With this practical distance calculator, Algorithm 2 outlines the procedure for inducing HDDT.

Algorithm 2 *HDDT*

```
Require: Training set $T$, Cut-off size $C$, Tree node $n$
if $|T| < C$ then
    return
end if
$n \leftarrow argmax_f Calc\_Binary\_Hellinger(T, f)$
for each value v of $b$ do
    create $n'$, a child of $n$
    $HDDT(T_{x_b=v}, C, n')$
end for
```

Note that Algorithm 2 does not include any pruning or collapsing with HDDT, and we smooth the leaf frequencies with the Laplace estimate. This was primarily motivated by the observations of Provost and Domingos (2003) on C4.5.

4 Combining sampling, ensembles, and decision trees

Sampling is a popular solution to the class imbalance problem; consequently a number of effective sampling methods have been proposed and studied (Japkowicz 2000; Kubat and Matwin 1997; Batista et al. 2004; Van Hulse et al. 2007; Chawla et al. 2002). We compare against two popular and effective sampling methods in this paper: random undersampling and SMOTE (Chawla et al. 2002). SMOTE and undersampling have both been shown to outperform oversampling by replication when using decision trees. In prior work (Chawla et al. 2008) we demonstrated that a combination of undersampling and SMOTE generally outperforms each of the individual sampling methods as well, and proposed a wrapper method to determine the potentially optimal amounts of sampling. In the evaluations reported here, we use the same wrapper methodology to determine the amounts of sampling for both HDDT and C4.4. The wrapper discovers the sampling strategies that optimize AUC by first determining undersampling levels for majority classes in order from largest to smallest and then finding SMOTE levels for minority classes in order from smallest to largest.

In addition to these sampling methods, we evaluate multiple ensemble methods including: BG, boosting, and MB. BG is applied using both HDDT and C4.4 decision trees, and, to avoid any variation in results due to the choice of bootstrap replicates, we use the same bags for both HDDT and C4.4. When boosting, we use AdaBoost.M1 for binary class datasets and AdaBoost.M1W for multi-class datasets as proposed

Table 1 Legend of method abbreviations

GR	Gain ratio (C4.4)
HD	Hellinger distance
T	Single tree using either HD (HDDT) or GR (C4.4)
BG	Bagging
BT	Boosting
MB	Majority bagging
SE	Balance classes with SMOTE
SW-*X* w/*Y*	Optimize sampling using classifier *X*, then use final classifier *Y*

by Freund and Schapire (1996). On imbalanced datasets we also consider “majority bagging” (Hido and Kashima 2008) which randomly selects examples with replacement from the original training data to generate new training samples. Unlike traditional BG, however, selection weights are assigned to ensure class balance in each new training bag. In other words, to generate each bag the majority class is undersampled and the minority class oversampled (if necessary) to generate a bag with a balanced class distribution from an imbalanced training set.

For consistency, we chose to build all ensembles with 100 decision trees (as recommended for boosting by Breiman (1998)). The ensemble methods are also used with the sampling strategies. To date, ensembles have not been widely used in conjunction with sampling wrappers; hence, best practices regarding this fusion are as yet unknown. To this end, we consider multiple permutations for optimization, i.e., comparing the use of a single tree against an ensemble of trees in order to select the appropriate sampling levels.

Essentially, our experimental framework includes: single trees (T), BG, MB, AdaBoost (BT), sampling methods with parameters optimized on single trees and built with single trees, and sampling methods with ensembles of trees. We believe our work is the most extensive study with decision trees for imbalanced data to date.

5 Experimental setup

In this section we outline how we compare the methods outlined previously in Sect. 4. Table 1 provides the abbreviations used throughout the rest of the paper.

5.1 Evaluation

In order to compare the methods, a total of 58 datasets (Tables 2 and 3) were chosen from a wide variety of application areas such as finance, biology and medicine. These datasets originate from public sources such as UCI (Asuncion and Newman 2007), LibSVM (Chang and Lin 2001), and previous studies (Cieslak and Chawla 2008a; Chawla et al. 2002). In order to measure each dataset’s level of imbalance, we compute the coefficient of variation (CV) which provides a measure of skew that generalizes to more than two classes (Wu et al. 2010). Specifically, CV is the

Table 2 Statistics for the balanced datasets used in this paper

Dataset	Number of features	Number of classes	Number of examples	CV
breast-w	9	2	699	0.31
bupa	6	2	345	0.16
credit-a	15	2	690	0.11
crx	15	2	690	0.11
fourclass	2	2	862	0.29
heart-c	13	2	303	0.08
heart-h	13	2	294	0.28
horse-colic	22	2	368	0.26
ion	34	2	351	0.28
krkp	36	2	3196	0.04
led-24	24	10	5000	0.03
letter-26	16	26	36000	0.03
pendigits-10	16	11	10993	0.32
pima	8	2	768	0.30
promoters	57	2	106	0.00
ringnorm	20	2	300	0.09
segment-7	19	7	2310	0.00
sonar	60	2	208	0.07
splice-libsvm	60	2	1000	0.03
SVMguide1	4	2	3089	0.29
threenorm	20	2	300	0.00
tic-tac-toe	9	2	958	0.31
twonorm	20	2	300	0.01
vehicle	18	4	846	0.04
vote	16	2	435	0.23
vote1	15	2	435	0.23
vowel	10	11	528	0.00
waveform	21	3	5000	0.01
zip	256	10	9298	0.28

proportion of the deviation in the observed number of examples for each class versus the expected number of examples in each class. For our purposes, datasets with a CV above 0.35—a class ratio of 2:1 on a binary dataset—are considered imbalanced. This evenly divides our pool of available datasets into 29 balanced and 29 imbalanced datasets. When evaluating each of the classifiers on the datasets, 5 × 2 cross-validation is used as recommended by Dietterich (1998). In this procedure, each dataset is broken into class stratified halves, allowing two experiments in which each half is once used as the training and the other in testing. This halving is iterated five times, and the average result over these ten repetitions is considered (Alpaydin 1999).

Table 3 Statistics for the imbalanced datasets used in this paper

Dataset	Number of features	Number of classes	Number of examples	CV
bgp	9	4	24984	1.26
boundary	175	2	3505	0.93
breast-y	9	2	286	0.41
cam	132	2	18916	0.90
car	6	4	1728	1.08
compustat	20	2	13657	0.92
covtype	10	2	38500	0.86
credit-g	20	2	1000	0.40
dna	180	3	3186	0.39
estate	12	2	5322	0.76
germannumer	24	2	1000	0.40
glass	9	6	214	0.76
heart-v	13	2	200	0.49
hypo	25	2	3163	0.90
ism	6	2	11180	0.95
letter	16	2	20000	0.92
nursery	8	5	12961	0.95
oil	49	2	937	0.91
optdigits	64	2	5620	0.80
page	10	2	5473	0.80
page-5	10	5	5473	1.75
pendigits	16	2	10992	0.79
phoneme	5	2	5404	0.41
PhosS	480	2	11411	0.89
sat	36	6	6435	0.37
satimage	36	2	6430	0.81
segment	19	2	2310	0.71
shuttle	9	7	58000	1.87
splice	60	3	3190	0.39

In this paper, we slightly modify the procedure from Chawla et al. (2008) when using the sampling wrapper. Each training fold is further subdivided, again using the 5×2 cross-validation methodology in order to reduce the effects of variance which may be underestimated when using fivefold cross-validation as in the original method (Dietterich 1998). Each sub-training fold thus is comprised of one quarter of the original data, and the standard methodology in Chawla et al. (2008) is used to identify optimal sampling levels which are in turn applied to the original training sample to induce a final classifier evaluated on the respective testing sample.

5.2 Evaluation measures

In order to compare different classifiers' performance on a dataset, they must be evaluated by some evaluation measure. Typically this measure is the predictive accuracy, however this measure assumes all errors are weighted equally. This assumption is not always appropriate, e.g., when the data is imbalanced. The ROC curve is a standard technique for summarizing classifier performance on imbalanced datasets. Given this, a popular evaluation metric is the AUC, which measures the probability of ranking a random positive class example over a random negative class example. We use the rank-order formulation of AUC which is akin to setting different thresholds on the probabilistic estimates and generating a tpr and fpr (Hand and Till 2001). The AUC is then calculated as, given n_0 points of class 0, n_1 points of class 1, and S_0 as the sum of ranks of class 0 examples (Hand and Till 2001): $AUC = \frac{2S_0 - n_0(n_0+1)}{2n_0 n_1}$ For a multiple class dataset, we average AUC over all pairs of classes (Hand and Till 2001) using: $AUC_m = \frac{2}{c(c-1)} \sum_{i<j} AUC(i, j)$.

One advantage of AUC is that it does not rely on any threshold. This allows one to evaluate the general performance of a classifier across the different trade-offs between the tpr and fpr at varying decision thresholds. One disadvantage of AUC is that it does not entirely distinguish between the curves that may cross in the ROC space. Thus, at a specific operating point classifier A may outperform classifier B, but the overall AUC of A may be lower. Thus choosing a classifier based on AUC may not be optimal in all cases. Under such circumstances, the problem then becomes choosing the right operating point. If one is working in a domain where the relative weights of class importance or costs of making errors are available, then the operating point can be directly chosen. Often, however, this is not the case for the datasets used in the academic literature. Hence AUC has become a popular measure of choice.

Another popular evaluation measure is F-measure. F-measure is a class of measures which captures the harmonic mean of the precision and recall of a classifier. In this paper, we consider the F_1-measure, where equal importance is given to both precision and recall. We consider the TP, FP, and FN as defined by a standard confusion matrix. The F_1-measure is defined as: $F_1 = \frac{2PR}{P+R}$, where $P = \frac{TP}{TP+FP}$ is precision and $R = \frac{TP}{TP+FN}$ is recall. For multiple-class imbalanced datasets, we applied a strategy similar to computing AUC over multiple classes, i.e., we average F_1-measure over all pairs of classes (Hand and Till 2001).

Finally, for balanced datasets we evaluate using the traditional accuracy measure.

5.3 Statistical tests

Demšar (2006) suggests that the best way to consider the performance of classifiers across multiple datasets is through a comparative analysis of averaged performance ranks. As previously noted, we use accuracy to rank the methods on balanced datasets, and AUC and F-measure for imbalanced datasets, where rank 1 denotes the best method. The Friedman test (1940) is then performed to determine if there is a significant difference in the rankings through the Holm procedure (1979), which is a step-down approach. If this procedure determines method A to rank statistically significantly ahead of method B across the considered datasets we may generally recommend the use of A over B. We note that this test requires the conservation of the sum of ranks on each dataset. Thus, in the case of a tie (scores within 0.0025) the average rank is assigned. For example, if two classifiers tie for first, they both receive a rank of 1.5, or if three tie for first, they each receive a rank of 2.

6 Imbalanced datasets results

For the sake of clarity, we divide the results into binary and n-ary imbalanced datasets in addition to providing a combined analysis based on the two. This differentiation is necessary as the sampling methods exhibit different performance characteristics between cases. To account for this, each minority class in the n-ary datasets will need to be considered separately to counter the problem of class imbalance.

6.1 Binary classes

Table 4 contains the results of our experiments on binary class imbalanced datasets. The numbers reported represent the average classifier rank in terms of AUC across all the binary class imbalanced datasets for each considered method. An "×" next to a given method indicates that the method performs statistically significantly worse at that column's confidence level than the best average classifier (in the case of Table 4 that is bagged HDDT).

From Table 4 we make the following observations when using C4.4 decision trees for imbalanced data:

(1) Sampling methods (SE, SW-T w/T), as expected, help C4.4 when learning on the imbalanced datasets.
(2) Ensemble methods (BG, BT) are statistically significantly preferred over not only the single tree (T), but also single decision trees learned from the sampled dataset (SE, SW-T w/T). They also drive performance improvements over sampling (SW-X w/X).
(3) When considering BG in combination with the sampling wrapper (SW-X w/Y), we note that there is only a marginal separation of ranks when a single tree or ensemble of classifiers is used to optimize sampling levels (SW-T vs. SW-BG and SW-BT), indicating that a single tree is a sufficient heuristic for BG in these circumstances and may be used in lieu of BG in the optimization phase to conserve computational expense.

Based on these overall results, we recommend the use of boosting, BG, and the sampling wrapper with boosting when using C4.4 on imbalanced datasets.

The following observations can be derived for HDDTs from Table 4:

(1) A single HDDT (T) removes the need for sampling (SE, SW-T w/T). This seems a significant result, as it shows how to learn (single) decision trees for skewed data without sampling while still improving performance.
(2) Ensemble methods (BG, MB, BT) significantly outperform the single HDDTs (T), as was also observed with C4.4.

Table 4 AUC ranks and statistical significance test results (at 90, 95, and 99% confidence levels) for binary imbalanced datasets

			Confidence		
Base learner	Classifier	Average rank	90%	95%	99%
C4.4	BG	6.50			
	T	17.05	×	×	×
	BT	6.65			
	MB	10.45			
	SE	16.40	×	×	×
	SW-T w/T	16.65	×	×	×
	SW-T w/BG	8.90			
	SW-BG w/BG	8.15			
	SW-T w/BT	8.88			
	SW-BT w/BT	7.72			
HDDT	**BG**	**4.40**			
	T	14.95	×	×	×
	BT	7.22			
	MB	9.20	×		
	SE	15.40	×	×	×
	SW-T w/T	16.50	×	×	×
	SW-T w/BG	10.05	×	×	
	SW-BG w/BG	8.20			
	SW-T w/BT	8.55			
	SW-BT w/BT	8.18			

"×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., bagged HDDT

(3) Bagging HDDT (BG) rather than boosting (BT) is the top performer in this set of results, and has the best overall rank among all considered classifiers. In fact, bagged HDDTs are the best performing classifiers across all (including C4.4 based) classifiers. We do note that BG and boosting both types of decision trees will typically produce favorable AUC results.

To summarize, BG HDDT is the strongly preferred method, as it averages two ranks ahead of the next best method (the C4.4 BG solution). This indicates that BG will generally give the best AUC performance on imbalanced datasets with two classes and we therefore recommend the use of HDDT with BG when the class imbalance CV is above 0.35. In Sect. 7 we extend this result by showing that no harm is done when using HDDT with BG when the CV is lower than 0.35.

6.1.1 Leaf probability estimates

To try to understand the differing impact of splitting criteria on highly imbalanced binary class data, it might be useful to examine the actual probability estimates generated by the decision tree. In this section we compare the probability estimates generated by C4.4 and HDDT, and its impact on the classifier's AUC performance.

In order to compare the methods, for each of the datasets we ran 5x2-fold cross-validation. For each dataset we then determined how many leaves predicted (1) the minority class (2) give no prediction (i.e., contain an equal number of majority and minority class instances (3) the majority class. The results of these tests can be found in Tables 5 and 6.

In the majority of the cases we see that the Hellinger trees produced, on average, more leaves than the C4.4 trees. Furthermore, while for the cases where C4.4 built deeper trees the sizes were comparable, this was not always the case when Hellinger trees were built deeper. For the cam dataset, for instance, the average C4.4 tree had 235.22 leaves, while the average Hellinger tree had $1,344$ leaves. This difference shows that Hellinger trees have the potential of growing vastly deeper trees than C4.4 is able to on the same dataset. This enables Hellinger trees to find more fine-grained differences in the datasets since it can better differentiate the data, as evidenced by making more splits to further distinguish between the positive (minority) and negative (majority) class. Previous research has demonstrated that unpruned decision trees are more effective in their predictions on minority class, and also result in improved calibrated estimates (Provost and Domingos 2003; Chawla 2003).

This becomes most obvious as the imbalance becomes worse (i.e., a CV ≥ 0.80, such datasets are denoted by bold in Tables 5 and 6). In such instances, C4.4 only builds deeper trees twice (hypo and oil), and only results in higher AUC once (letter). This is very strong evidence to the effectiveness of Hellinger trees in highly imbalanced data, and their ability to pick out fine differences in instances which lead to more accurate predictions overall.

In addition to building deeper trees, Hellinger trees are also better able to create leaves which predict a class. That is, on imbalanced datasets an average C4.4 tree creates leaves with an equal number of majority and minority class instances 22.2% of the time, while Hellinger trees create such leaves only 16.0% of the time. This is significant in classification scenarios, as it means that a randomly drawn instance from the feature space is more likely to be classified by a Hellinger tree than a C4.4 tree. This observation is equally extensible to the case of only considering datasets where the CV ≥ 0.80, in which case C4.4 averages 18.1% of such leaves and Hellinger only 12.4%.

6.2 Multiple classes

Table 4 examined imbalanced data with binary classes; Table 7 repeats the analysis for imbalanced data with more than two classes. Here we note that boosting C4.4 had the best rank. Though it is not statistically significantly better than BG or boosting with HDDT, perhaps this result indicates one area of improvement for HDDT. Given that distance is defined as a separation between two distributions (i.e., classes in this case), it is not trivially extensible to multiple classes, thus creating a slight dip in the performance estimates.

Table 5 Comparing the leaves of 50 C4.4 trees and 50 Hellinger trees

Dataset	Pred. min.	Pred. equal	Pred. maj.	Total
	C4.4 leaf distributions			
boundary	955 (11.7)	**3127** (**38.5**)	4050 (49.8)	8132
breast-y	1556 (14.6)	5896 (55.5)	3176 (**29.9**)	10628
cam	1576 (13.4)	3569 (**30.3**)	6616 (56.3)	11761
compustat	1611 (**29.5**)	434 (8.0)	3412 (**62.5**)	5457
covtype	2607 (35.9)	662 (9.1)	3996 (**55.0**)	7265
credit-g	5191 (24.7)	**7785** (**37.1**)	**8009** (38.2)	**20985**
estate	230 (19.4)	84 (7.1)	874 (**73.6**)	1188
germannumer	**3609** (34.8)	**1522** (**14.7**)	**5248** (**50.6**)	**10379**
heart-v	**1996** (**64.3**)	308 (9.9)	798 (25.7)	**3102**
hypo	**1826** (**63.3**)	**369** (12.8)	**690** (23.9)	**2885**
ism	1104 (**33.4**)	303 (9.2)	1901 (57.5)	3308
letter	1529 (**26.6**)	**677** (**11.8**)	3532 (61.6)	5738
oil	**428** (26.4)	**173** (**10.7**)	**1023** (**63.0**)	**1624**
page	1945 (**43.4**)	485 (10.8)	2055 (**45.8**)	4485
pendigits	**1562** (31.7)	**460** (9.3)	**2906** (**59.0**)	**4928**
phoneme	3817 (**42.8**)	683 (7.7)	4416 (**49.5**)	8916
PhosS	5415 (**33.3**)	**3175** (**19.6**)	7649 (47.1)	16239
satimage	**4722** (**35.0**)	**1595** (**11.8**)	7174 (53.2)	13491
segment	**441** (**37.6**)	**86** (**7.3**)	**647** (55.1)	**1174**
Totals	42120 (**29.7**)	31393 (**22.2**)	68172 (48.1)	141685
	HDDT leaf distributions			
boundary	**1983** (**19.1**)	1288 (12.4)	**7138** (**68.6**)	**10409**
breast-y	**1614** (**14.7**)	**6194** (**56.4**)	**3176** (28.9)	**10984**
cam	**11793** (**17.5**)	**10275** (15.3)	**45132** (**67.2**)	**67200**
compustat	**5950** (28.5)	**2122** (**10.2**)	**12794** (61.3)	**20866**
covtype	**2983** (**36.5**)	**870** (**10.6**)	**4327** (52.9)	**8180**
credit-g	**5279** (**27.1**)	6348 (32.6)	7861 (**40.3**)	19488
estate	**9159** (**24.9**)	**6304** (**17.2**)	**21260** (57.9)	**36723**
germannumer	3445 (**37.1**)	1207 (13.0)	4639 (49.9)	9291
heart-v	1261 (52.1)	**346** (**14.3**)	**812** (**33.6**)	2419
hypo	1017 (51.7)	308 (**15.7**)	643 (**32.7**)	1968
ism	**2108** (25.5)	**1140** (**13.8**)	**5008** (**60.7**)	**8256**
letter	**1559** (26.6)	584 (9.9)	**3728** (**63.5**)	**5871**
oil	397 (**27.8**)	147 (10.3)	885 (61.9)	1429
page	**2896** (41.8)	**993** (**14.3**)	**3039** (43.9)	**6928**
pendigits	1259 (**31.9**)	404 (**10.2**)	2288 (57.9)	3951
phoneme	**11133** (41.5)	**3298** (**12.3**)	**12388** (46.2)	**26819**
PhosS	**8692** (25.1)	3118 (9.0)	**22859** (**65.9**)	**34669**
satimage	4661 (33.7)	1349 (9.7)	**7838** (**56.6**)	**13848**

Table 5 continued

Dataset	Pred. min.	Pred. equal	Pred. maj.	Total
segment	275 (33.3)	54 (6.5)	498 (**60.2**)	827
Totals	**77464** (26.7)	**46349** (16.0)	**166313** (**57.3**)	**290126**

For each tree type, the total number of leaves (and relative percentages) are given which predict (1) the minority class (2) give no prediction (i.e., contain an equal number of majority and minority class instances) (3) the majority class (4) the total number of leaves. Dataset names in bold indicate a CV ≥ 0.80

Table 6 AUC performance results (rank in parenthesis) of the experiments performed as in Table 5

Dataset	C4.4	HDDT
boundary	0.57722 (2)	**0.60206 (1)**
breast-y	**0.60304 (1)**	0.58859 (2)
cam	0.64260 (2)	**0.68248 (1)**
compustat	0.81276 (2)	**0.83553 (1)**
covtype	0.97960 (2)	**0.98309 (1)**
credit-g	**0.68062 (1)**	0.68055 (2)
estate	**0.59645 (1)**	0.58821 (2)
germannumer	0.69741 (2)	**0.70887 (1)**
heart-v	**0.62668 (1)**	0.58499 (2)
hypo	0.97721 (2)	**0.98138 (1)**
ism	0.89895 (2)	**0.91360 (1)**
letter	**0.99518 (1)**	0.99214 (2)
oil	0.81574 (2)	**0.83104 (1)**
page	0.97802 (2)	**0.97877 (1)**
pendigits	0.98781 (2)	**0.99254 (1)**
phoneme	0.89706 (2)	**0.90443 (1)**
PhosS	0.60976 (2)	**0.68539 (1)**
satimage	0.90868 (2)	**0.91592 (1)**
segment	0.98473 (2)	**0.99208 (1)**
rank	1.68421	1.31579

Dataset names in bold indicate a CV ≥ 0.80

6.3 Summary on all datasets

Table 8 contains the results for all 29 imbalanced datasets combined (binary and multiple class). As the bagged ensemble and the boosted ensemble were the most competitive, we only show the results on the single tree, bagged ensemble, and the boosted ensemble. Once all the datasets and methods are combined, bagged HDDT achieves the best overall performance.

6.4 Using F_1-measure

As stated in Sect. 1, we wanted to evaluate HDDTs with different popular evaluation methods to avoid possible generalization of results stemming from one measure.

Table 7 AUC ranks and statistical significance test results (at 90, 95, and 99% confidence levels) for multiple class imbalanced datasets

			Confidence		
Base learner	Classifier	Average rank	90%	95%	99%
C4.4	BG	7.75			
	T	14.12	×	×	×
	BT	5.19			
	MB	7.38			
	SE	13.25	×	×	×
	SW-T w/T	16.00	×	×	×
	SW-T w/BG	10.25			
	SW-BG w/BG	10.12			
	SW-T w/BT	8.38			
	SW-BT w/BT	7.44			
HDDT	BG	7.25			
	T	16.75	×	×	×
	BT	7.00			
	MB	7.25	×		
	SE	14.88	×	×	×
	SW-T w/T	17.75	×	×	×
	SW-T w/BG	10.12	×	×	
	SW-BG w/BG	9.69			
	SW-T w/BT	10.31			
	SW-BT w/BT	9.12			

"×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., bagged HDDT

Table 8 AUC ranks and statistical significance test results (at 90, 95, and 99% confidence levels) for all imbalanced datasets

			Confidence		
Base learner	Classifier	Average rank	90%	95%	99%
C4.4	BG	6.23			
	T	16.21	×	×	×
	BT	6.86			
HDDT	BG	5.21			
	T	15.46	×	×	×
	BT	7.10			

"×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., bagged HDDT

To this end we present the F_1-measure, which is another popular measure for evaluation on imbalanced datasets. Again due to the performance characteristics of the other methods, and in order to increase clarity of presentation, the point of comparison

Table 9 F_1-measure ranks and statistical significance test results (at 90, 95, and 99% confidence levels) for all imbalanced datasets

			Confidence		
Base learner	Classifier	Average rank	90%	95%	99%
C4.4	BT	7.60			
	T	12.05	×	×	×
	BG	9.18			
	SW-T w/T	12.00	×	×	×
HDDT	BT	7.92			
	T	10.55	×	×	×
	BG	8.62			
	SW-T w/T	13.30	×	×	×

"×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., boosted C4.4

is largely restricted to ensembles, single trees, and the sampling wrapper with a single tree. We now investigate the questions: *Do bagged HDDTs generally outperform single HDDTs?* and: *Are HDDTs superior to C4.4 (C4.5)?*

Table 9 agree with the observations obtained via AUC, i.e., HDDT is superior to C4.4. Bagged HDDT is significantly better than a single HDDT.

Thus, based on both AUC and F_1-Measure we are able to recommend Bagged HDDTs as the preferred method when dealing with imbalanced data.

7 Balanced datasets results

In addition to examining the results of several methods using gain ratio and Hellinger distance based trees as base classifiers on imbalanced data, we also explore performance across a number of balanced datasets to determine if there is the same delineation between the two splitting metrics. For the balanced data sets, we use the original C4.5 method. Our conjecture was that the differences would diminish and both gain ratio and Hellinger distance would prove to be comparable for balanced datasets. As before, results are reported as average performance ranks across all considered datasets. However, for balanced datasets we used the overall accuracy performance measure, since under these conditions it is an appropriate measure. We also greatly reduce the number of methods considered to single tree, BG, and boosting, since the other methods are appropriate only to learning from imbalance datasets.

Table 10 shows the results for these experiments. Note that there was no statistically significant difference in performance between C4.5 and HDDT, indicating that the use of HDDT is not detrimental when applied to balanced data. Only the single tree methods are statistically significantly worse than the best ensemble method. This confirms the point (already well demonstrated for gain ratio) that ensembles generally improve accuracy over single decision trees, although this was an as yet unknown result for Hellinger distance trees.

Table 10 Accuracy ranks and statistical significance test results (at 90, 95, and 99% confidence levels) for all balanced datasets

			Confidence		
Base learner	Classifier	Average rank	90%	95%	99%
C4.5	BT	2.12			
	T	5.10	×	×	×
	BG	3.03			
HDDT	BT	2.16			
	T	5.55	×	×	×
	BG	3.03			

"×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., boosted C4.5

8 Conclusion

In this paper we compared BG, boosting, and a sampling wrapper, in addition to combinations of each method with respect to two separate splitting criterion for decision trees: gain ratio and Hellinger distance. An experimental framework using 5x2 cross-validation compared AUC and F_1-measure performance on 29 imbalanced datasets and accuracy for 29 balanced datasets, allowing a large-scale and robust analysis of relative performances. The Holm procedure of the Friedman test was used to determine the significance of results across multiple datasets.

Based on the experiments, we make a novel and practical recommendation for learning decision trees on imbalanced data, especially binary classification data. We demonstrated that HDDTs are robust in the presence of class imbalance, and when combined with BG they mitigate the need for sampling. This is a compelling result, as it makes bagged HDDTs particularly relevant for practitioners who don't have to then concern themselves with more expensive sampling methods. We also showed that HDDTs are not significantly worse than C4.5 for balanced datasets; thus, it is sensible to use Hellinger distance over gain ratio even on balanced datasets.

In light of the observations within this report, we claim that HDDT are not only skew-insensitive as suggested in Cieslak and Chawla (2008a), but also robust in their applicability to wide variety of datasets. Thus, based on the reported findings, we recommend Hellinger distance for use in place of gain ratio in generating decision tree splits. All the datasets and software used in this paper are available via http://www.nd.edu/~dial/hddt.

Acknowledgements This work was supported in part by the Arthur J. Schmitt Fellowship, NSF ECCS-0926170, and the US Department of Energy through the ASC CSEE Data Discovery Program, administered by Sandia National Laboratories, contract number: DE-AC04-76DO00789. The authors would like to thank Ken Buch for his help with Avatar Tools, in addition to the reviewers and editor assigned to refereeing this report for their helpful feedback.

References

Alpaydin E (1999) Combined 5×2cv F test for comparing supervised classification learning algorithms. Neural Comput 11(8):1885–1892

Asuncion A, Newman D (2007) UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html

Banfield R, Hall LO, Bowyer KW, Kegelmeyer WP (2007) A comparison of decision tree ensemble creation techniques. IEEE Trans Pattern Anal Mach Intell 29(1):832–844

Batista G, Prati R, Monard M (2004) A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor 6(1):20–29

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Breiman L (1998) Rejoinder to the paper 'Arcing Classifiers' by Leo Breiman. Ann Stat 26(2):841–849

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Breiman L, Friedman J, Stone CJ, Olshen R (1984) Classification and regression trees. Chapman and Hall, Boca Raton

Chang C, Lin C (2001) LIBSVM: a library for support vector machines. software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. Accessed June 2011

Chawla NV (2003) C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: ICML workshop on learning from imbalanced data sets II. Washington, DC, USA, pp 1–8

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Chawla NV, Japkowicz N, Kolcz A (2004) Editorial: learning from imbalanced datasets. SIGKDD Explor 6:1–16

Chawla NV, Cieslak DA, Hall LO, Joshi A (2008) Automatically countering imbalance and its empirical relationship to cost. Data Min Knowl Discov 17(2):225–252

Cieslak DA, Chawla NV (2008a) Learning decision trees for unbalanced data. In: European conference on machine learning (ECML). Antwerp, Belgium, pp 241–256

Cieslak DA, Chawla NV (2008b) Analyzing classifier performance on imbalanced datasets when training and testing distributions differ. In: Pacific-Asia conference on knowledge discovery and data mining (PAKDD). Osaka, Japan, pp 519–526

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Dietterich TG (1998) Approximate statistical tests for comparing supervised classiffication learning algorithms. Neural Comput 10(7):1895–1923

Dietterich T (2000) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn 40(2):139–157

Dietterich T, Kearns M, Mansour Y (1996) Applying the weak learning framework to understand and improve C4.5. In: Proceeings of the 13th international conference on machine learning. Morgan Kaufmann, Bari, Italy, pp 96–104

Drummond C, Holte R (2000) Exploiting the cost (in)sensitivity of decision tree splitting criteria. In: International conference on machine learning (ICML). Stanford University, California, USA, pp 239–246

Drummond C, Holte R (2003) C4.5, Class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: ICML workshop on learning from imbalanced datasets II. Washington, DC, USA, pp 1–8

Flach PA (2003) The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In: International conference on machine learning (ICML). Washington, DC, USA, pp 194–201

Freund Y, Schapire R (1996) Experiments with a new boosting algorithm. In: Proceedings of the 13th national conference on machine learning. Bari, Italy, pp 148–156

Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Ann Math Stat 11(1):86–92

Halmos P (1950) Measure theory. Van Nostrand and Co., Princeton

Hand D, Till R (2001) A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach Learn 45:171–186

Hido S, Kashima H (2008) Roughly balanced bagging for imbalanced data. In: SIAM international conference on data mining (SDM). Atlanta, Georgia, USA, pp 143–152

Ho T (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844

Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70

Japkowicz N (2000) Class imbalance problem: significance & strategies. In: International conference on artificial intelligence (ICAI). Las Vegas, Nevada, USA, pp 111–117

Kailath T (1967) The divergence and Bhattacharyya distance measures in signal selection. IEEE Trans Commun 15(1):52–60

Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. In: International conference on machine learning (ICML). Nashville, Tennessee, USA, pp 179–186

Nguyen X, Wainwright MJ, Jordan MI (2007) Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In: Advances in neural information processing systems (NIPS). Vancouver, BC, Canada, pp 1–8

Provost F, Domingos P (2003) Tree induction for probability-based ranking. Mach Learn 52(3):199–215

Quinlan R (1986) Induction of decision trees. Mach Learn 1:81–106

Rao C (1995) A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance. Questiio (Quaderns d'Estadistica i Investig Oper) 19:23–63

Schapire R, Singer Y (1999) Improved boosting algorithms using confidence-rated predictions. Mach Learn 37:297–336

Van Hulse J, Khoshgoftaar T, Napolitano A (2007) Experimental perspectives on learning from imbalanced data. In: International conference on machine learning (ICML). Corvalis, Oregon, USA, pp 935–942

Vilalta R, Oblinger D (2000) A quantification of distance-bias between evaluation metrics in classification. In: International conference on machine learning (ICML). Stanford University, California, USA, pp 1087–1094

Wu J, Xiong H, Chen J (2010) Cog: local decomposition for rare class analysis. Data Min Knowl Discov 20:191–220

Zadrozny B, Elkan C (2001) Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: Proceedings of the 18th international conference on machine learning. Morgan Kaufmann, San Francisco, CA, USA, 2001, pp. 609–616