Improving Decision Tree Learning by Optimal Split Scoring Function Estimation

Banghua Zhu*, Jiantao Jiao†, Yanjun Han†, and Tsachy Weissman†

November 15, 2017

Abstract

The construction of efficient decision trees remains an actively researched topic in the field of machine learning. When making splits in decision trees, the Shannon entropy is widely used as scoring function. However, the commonly used Maximum Likelihood Estimator (MLE), which plugs-in the empirical distribution into those functions, has been proven to be significantly biased and to not meet the minimax rates. In particular, there exist estimators whose performance with n samples is essentially that of the MLE with $n \ln n$ samples, which is also minimax rate-optimal.

We show analytically and numerically that replacing the MLE with minimax rate-optimal estimators for the Shannon entropy (or mutual information) significantly improves the performance of the decision tree in terms of classification. We empirically compare thirteen entropy estimation schemes in terms of decision tree classification accuracy, and validate the claim that the minimax rate-optimal split scoring function estimation leads to the best classification performance.

1 Introduction

Decision tree learning is a non-parametric supervised learning method used for classification and regression in the fields of statistics, data mining and machine learning. Although decision tree learning was among the earliest machine learning methods to be developed, it is still popular for its simplicity and efficiency. Furthermore, the theory of decision trees has laid a solid foundation for improving performance in the popular modern machine learning algorithms class of ensemble decision trees, e.g. random forest [1], boosted tree [2] and deep forest [3]. The study of decision tree algorithms can further improve these tree ensemble methods.

Decision tree is a flowchart-like tree structure representing the procedure of decision. Each interior node of decision tree denotes a test on an attribute, where a given dataset is partitioned into subsets by split scoring function on attributes. Leaf nodes of the tree are labeled with a class or real number representing a predicted outcome. An instance of classification is predicted by going through the tree from root to leaf.

The general idea of decision tree learning is to construct a decision tree and predict the value of a certain variable based on several input variables. In classification trees, we are predicting the class of a universe of objects from their attributes. In regression trees, the predicted outcome is a

*Banghua Zhu is with the Department of Electronic Engineering, Tsinghua University, email: 13aeon.v01d@gmail.com

†Jiantao Jiao, Yanjun Han and Tsachy Weissman are with the Department of Electrical Engineering, Stanford University, email: {jiantao,yjhan,tsachy}@stanford.edu

real number. Although decision tree learning is renowned for its simplicity, the optimal induction of decision trees according to some global objective has been proved to be fundamentally hard [4], and most implementations use randomized greedy algorithms for growing a decision tree [5].

There are many well-established decision tree algorithms. The Iterative Dichotomiser 3 (ID3) [6] and C4.5 [7]algorithms are among the most popular. The ID3 algorithm iterates through every unused attribute of the dataset and estimates the Shannon entropy [8]. It then selects the attribute which has the largest information gain and splits the dataset according to the selected attribute. The C4.5 algorithm is an extension of ID3 algorithm. The main difference is that C4.5 adopts a normalized information gain, i.e., the gain ratio, as the split scoring function.

In decision-tree construction, the split scoring function can be the most critical part. Many papers have analyzed the importance of the split scoring function [9] [10]. When splitting the dataset, the underlying discrete distribution of the data is unknown, requiring splitting scoring function to be based on estimates of functions of the underlying distribution. In ID3 and C4.5, the split scoring function is built up by comparing the information gain or gain ratio from the data, where estimating the Shannon entropy is an important step. Traditionally, these quantities have been calculated using the Maximum Likelihood Estimator (MLE), but it is apparent that usage of the MLE is not without its drawbacks. The MLE is a good estimator for Shannon entropy only when the sample size n is much larger than the support size S of the distribution. However, in decision tree learning and construction, the available sample size decreases exponentially fast with each node splitting decision, which renders the splitting operation increasingly unreliable. Thus, it is crucial to employ (near) optimal estimators for split scoring function in order to achieve higher accuracy.

The idea of adopting better estimators for Shannon entropy in decision tree training was proposed in [11]. There, the Grassberger estimator was utilized to estimate the entropy, resulting in statistically significant improvements in classification accuracy. However, the Grassberger estimator is not known to be a minimax rate-optimal estimator. In this paper, we provide theoretical and empirical validation to show that the minimax rate-optimal entropy estimators lead to consistent and significant performance boosts in learning tree models.

1.1 Main contributions

Our main contributions in this work are two-fold.

1. We prove that the improved entropy estimation also leads to improved classification tree construction for a particular choice of split scoring functions. In other words, we explicitly construct an example in which the decision tree induced by the optimal entropy estimator would be able to choose the better feature to split, but the maximum likelihood approach would fail to identify the optimal tree.

2. We extensively test thirteen entropy estimators on their performance in classification tree testing error. A Wilcoxon signed rank test rejects the null hypothesis that the minimax-rate-optimal estimator JVHW perform equally well as MLE for both ID3 and C4.5 decision tree. Furthermore, we provide the comparison between the thirteen entropy estimators for ID3 and C4.5.

We remark that we neither believe nor try to imply that the schemes we present here are necessarily competitive with the state-of-the-art for the applications we experimented with. Our

point, rather, is that machine learning schemes that have an entropy or mutual information estimation component stand to benefit from significant performance boosts via use of improved near optimal estimators. This is particularly true in large-alphabet regimes where use of the latter estimators in lieu of standard ones, such as empirical mutual information, can spell the difference between consistency and complete divergence.

2 Preliminaries

Notation: for non-negative sequences a_γ, b_γ, we use the notation $a_\gamma \lesssim b_\gamma$ to denote that there exists a universal constant C such that $\sup_\gamma \frac{a_\gamma}{b_\gamma} \leq C$, and $a_\gamma \gtrsim b_\gamma$ is equivalent to $b_\gamma \lesssim a_\gamma$. Notation $a_\gamma \asymp b_\gamma$ is equivalent to $a_\gamma \lesssim b_\gamma$ and $b_\gamma \lesssim a_\gamma$. Notation $a_\gamma \gg b_\gamma$ means that $\liminf_\gamma \frac{a_\gamma}{b_\gamma} = \infty$, and $a_\gamma \ll b_\gamma$ is equivalent to $b_\gamma \gg a_\gamma$. We denote by $\mathcal{M}_S$ the space of probability distributions on an alphabet with size S.

2.1 Decision Tree Learning

In decision tree learning, a tree is constructed from a given dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^n$. Here $\boldsymbol{x_i}$ is a m-dimensional vector representing the features of the i-th instance, and y_i is the corresponding label. Denote the set of attributes of the dataset by $\mathcal{A} = \{a_i\}_{i=1}^m$, and the set of labels by $\mathcal{Y} = unique(\{y_i\}_{i=1}^n)$. The basic algorithm is summarized in a pseudo-code format in Algorithm 1.

Algorithm 1 Construct_tree(Dataset $\mathcal{D}$, Attributes $\mathcal{A}$, Class label $\mathcal{Y}$)

1: Create a root node for the tree
2: **if** $|\mathcal{Y}| = 1$ or $|\mathcal{D}| <$ the minimal instance size of a leaf **then**
3: **return** a single-node root, with label = the most common class in $\mathcal{Y}$
4: Find the attribute a_i that classifies examples $\mathcal{D}$ to achieve the maximum split scoring function (information gain in ID3, gain ratio in C4.5). Add new tree branches below root corresponding to split examples
5: **for** k in new branches **do**
6: Assign split examples as a new dataset $\mathcal{D}_k = \{(\boldsymbol{x_{i_k}}, y_{i_k})\}$, and update $\mathcal{A}_k = \mathcal{A} - \{a_i\}$, $\mathcal{Y}_k = unique(\{y_{i_k}\})$
7: Below this new branch add the subtree Construct_tree(Dataset $\mathcal{D}_k$, Attributes $\mathcal{A}_k$, Class label $\mathcal{Y}_k$)
8: **return** root

Let $\mathcal{T}$ denote all the subsets created from splitting $\mathcal{D}$ by attribute a_i. And $T \in \mathcal{T}$ is the random variable denoting the subset to which a sample is assigned. Let $Y \in \mathcal{Y}$ be a random variable denoting all the class labels to split. For a random variable A obeying a discrete probability distribution $P = (p_1, p_2, ..., p_S)$, the Shannon entropy can be expressed as

$$H(A) = \sum_{i=1}^{S} p_i \log \frac{1}{p_i} \tag{1}$$

To measure mutual dependence between two discrete random variables A and B, mutual information is defined as

$$I(A;B) = H(A) + H(B) - H(A,B) = H(A) - H(A|B) \tag{2}$$

In ID3, information gain is used as split scoring function. It is a measure of the difference in entropy from before to after the dataset $\mathcal{D}$ is split on some attribute a_i. The information gain is exactly the mutual information between the variable T and Y.

$$\text{InfoGain}(a_i, \mathcal{D}) = I(Y;T) = H(Y) - H(Y|T) = H(Y) - \sum_{t \in \mathcal{T}} p(t) H(Y|T=t) \tag{3}$$

In C4.5, gain ratio is put forward as an alternative for information gain, which can be viewed as normalized information gain.

$$\text{GainRatio}(a_i, \mathcal{D}) = \frac{\text{InfoGain}(a_i, \mathcal{D})}{H(T)} \tag{4}$$

So the estimation of information gain and gain ratio reduces to the estimation of $H(Y)$, $H(T)$, and $H(Y|T=t)$.

2.2 Estimation of functionals of discrete distributions

According to the above discussion on decision tree learning, we focus on the estimation of Shannon entropy. Given n independant samples $(y_1, y_2, ..., y_n)$ from a random variable Y with unknown discrete probability distribution $P = (p_1, p_2, ..., p_S)$ and unknown support size S, we want to estimate the entropy $H(Y)$ empirically from samples $(y_1, y_2, ..., y_n)$. Estimating the Shannon entropy with finite samples is challenging [12], and we refer the readers to [13] for a recent survey.

From a sample complexity perspective, the maximum likelihood approach, which simply plugs-in the empirical distribution P_n in the entropy functional, requires $n \gg S$ samples to achieve consistent estimation. Recent advances of statistics and machine learning provide us with minimax rate-optimal approaches to estimate the Shannon entropy. It was first shown in [14] that one must have $n \gg \frac{S}{\ln S}$ in order to consistently estimate the entropy. Valiant and Valiant have proposed improved entropy estimators that achieve the optimal sample complexity [14] [15] [16]. Wu and Yang [17] and Jiao et al. [13] independently applied the idea of best polynomial approximation to entropy estimation, and obtained estimators that achieve the minimax rates. Concretely, the performance of the minimax rate-optimal estimators with n samples is essentially that of the MLE with $n \ln n$ samples. It has been empirically validated in [13] that the minimax rate-optimal entropy estimators lead to consistent and significant performance boosts in learning tree graphical models compared to the classical Chow–Liu algorithm.

3 Theoretical results

We show through the following construction that optimal splitting scoring function estimation leads to optimal tree construction.

Let (C, A_1, A_2) be jointly distributed random variables, where $C \in \mathcal{C}$ denotes the class label, $A_1 \in \mathcal{A}_1$ and $A_2 \in \mathcal{A}_2$ are two features. Suppose one relies on estimating the mutual information $I(C; A_i), i = 1, 2$ to determine which feature to split in the decision tree construction. We declare a node splitting decision to be a "mistake" if the algorithm chooses feature A_i to split while the true mutual information satisfies $I(C; A_i) < I(C; A_{3-i}) - \delta$ for some fixed constant $\delta > 0$. We have the following result. The proof is included in Appendix A.

Theorem 1 *Suppose $|\mathcal{A}_1| = |\mathcal{A}_2|$ (denoted as $|\mathcal{A}|$), $|\mathcal{A}| \to \infty, |\mathcal{C}| \to \infty$. If one uses the minimax rate-optimal estimator for mutual information (e.g. [13][17]), for any $\delta > 0$ we have*

$$\sup_P \mathbb{P}(I(C; A_{\hat{i}}) < I(C; A_{3-\hat{i}}) - \delta) \to 0 \tag{5}$$

when $n \gg \frac{|\mathcal{A}||\mathcal{C}|}{\log|\mathcal{A}|+\log|\mathcal{C}|}$.

The same statement is true for the maximum likelihood approach if $n \gg |\mathcal{A}||\mathcal{C}|$. Furthermore, if $n \lesssim |\mathcal{A}||\mathcal{C}|$, there exists some universal constant $\delta > 0$ such that

$$\sup_P \mathbb{P}(I(C; A_{\hat{i}^{\mathrm{MLE}}}) < I(C; A_{3-\hat{i}^{\mathrm{MLE}}}) - \delta) > 0 \tag{6}$$

where $\hat{i}^{\mathrm{MLE}}$ is given by the maximum likelihood approach.

Theorem 1 shows the necessity of replacing the maximum likelihood approach by employing the minimax rate-optimal estimator for mutual information: the maximum likelihood approach requires at least $n \gg |\mathcal{A}||\mathcal{C}|$ samples to succeed, while the latter one only need $n \gg \frac{|\mathcal{A}||\mathcal{C}|}{\log|\mathcal{A}|+\log|\mathcal{C}|}$ samples.

4 Experiments

We illustrate the accuracy of the (near) optimal approaches in estimating the splitting scoring function by comparing the thirteen estimators on UCI datasets [18].

4.1 Setup

We intentionally do not compare against other classification methods, because we want to assess only the contribution of improved entropy estimates. Cross comparison between ID3 and C4.5 is also omitted for this reason. For experiments on different datasets, we follow the recommendations in [19]. The decision trees with different split scoring functions are implemented in Python. In each tree splitting step, we split the data into a left subtree and right subtree. Thus a binary tree is constructed for classification. To prove that substitution of MLE could lead to improvement in real practice and to avoid over-fitting, we adopt reduced error pruning technique. Each dataset is split into 10 folds. 8 folds are used for training, 1 fold is used for reduced error pruning, and 1 fold for test. We repeat the procedure 100 times for each dataset and calculate the mean accuracy. All models are trained on exactly the same datasets using the same parameters. The node is considered as a leaf if all instances on that node belong to a same class or less than 2 instances remain in the node.

We use 19 datasets with different data types from UCI. Table 1 shows the basic information of these datasets.

4.2 Results

We test ID3 and C4.5 with in total 13 different entropy estimators on the 19 datasets.* The results are shown in Table 2 and 3. Here we use classification accuracy to evaluate the effectiveness and the total number of the tree nodes to measure the tree complexity.

*Experiments on dataset abalone, australian, heart, letter and pendigits are still not totally completed. In this paper we only provide the results for other 14 datasets. When calculating the p-value, we are using the results for all 19 datasets for MLE, JVHW and APML estimator since they are all finished.

Table 1: Basic information of test datasets

Dataset	Type	# of instance (n)	# of attribute	# of class
car	categorical	1728	6	4
iris	numeric	150	4	3
haberman	numeric	306	3	2
scale	categorical	625	4	3
abalone	mixed	4139	8	18
cmc	mixed	1473	9	3
wine	numeric	4898	12	11
transfusion	numeric	748	5	2
hepatitis	mixed	155	19	2
soybean	categorical	307	35	19
forest	mixed	326	27	4
chess	categorical	3196	36	2
monks	categorical	432	7	2
glass	numeric	214	10	7
bridges	mixed	108	13	6
australian	mixed	690	14	2
heart	mixed	270	13	2
letter	numeric	20000	16	26
pendigits	numeric	10992	16	10

The 13 estimators are the maximum likelihood estimator, Miller-Madow bias corrected estimator [20], the Jackknifed MLE [21], the unseen estimator by Valiant and Valiant [16], the coverage adjusted estimator (CAE) [22], the best upper bound estimator (BUB) [12], the shrinkage estimator [23], the Grassberger estimator [24], the Dirichlet-smoothed plug-in estimator [25], the Bayes estimator under Dirichlet prior [26], the Nemenman-Shafee-Bialek estimator (NSB) [27], Approximate Profile Maximum Likelihood (APML) estimator [28] and Jiao–Venkat–Han–Weissman (JVHW) estimator [13]. We calculate the p-value for rejecting the null hypothesis that one certain entropy estimator performs equally well as MLE estimator using Wilcoxon signed rank test. The result is shown in Table 4.

4.3 Discussion

From the three tables we can see the minimax-rate-optimal estimator JVHW brings significant improvement in accuracy with less complicated tree structure. This results from the accurate estimation of split scoring function of discrete distribution. The difference between the two estimators may be small for some datasets but statistically significant and we can conclude that improved entropy estimation yields improved classification trees. We further illustrate the relative accuracy improvement for JVHW compared with MLE in Fig. 1. JVHW is superior to MLE in 18 out of 19 datasets, and has brought relative improvement up to 3.3 percent.

Note that we are estimating the mutual information in each step, the support size S is indeed the number of class labels times the number of branches. Here we are doing binary splitting, so the

Table 2: Accuracy and number of nodes comparison of ID3 decision tree between 13 estimators.

Dataset	MLE		JVHW		APML		Valiant		Jackknifed	
	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes
car	94.79	74.37	94.77	73.76	94.82	74.47	94.86	73.78	95.04	71.80
iris	78.53	28.07	79.90	27.65	77.99	27.88	78.73	29.02	84.74	29.09
haberman	64.63	33.69	66.38	24.87	65.22	33.05	65.77	32.96	67.99	17.02
scale	77.11	65.82	76.93	60.72	77.49	63.79	76.03	61.32	76.00	55.68
cmc	44.45	204.35	46.01	146.45	44.77	187.52	44.30	196.67	47.29	133.29
wine	37.18	29.37	37.32	24.82	36.96	28.68	38.29	32.36	38.38	20.60
transfusion	72.45	54.45	73.25	40.65	72.46	51.71	71.23	52.00	72.65	41.67
hepatitis	81.13	7.59	81.06	7.02	81.12	7.30	80.61	6.78	81.73	4.67
soybean	71.80	38.58	72.26	37.76	71.82	38.23	73.59	38.67	74.29	36.89
forest	67.42	90.60	66.79	91.33	67.29	91.07	65.96	93.00	65.82	76.56
chess	98.90	48.56	98.90	48.00	98.90	48.67	98.92	47.93	98.88	47.51
monks-3	98.18	8.08	98.19	8.04	98.18	8.07	98.15	8.07	98.16	8.02
glass	43.44	44.83	43.83	41.92	43.17	43.26	43.76	46.56	42.38	20.67
bridges	52.43	21.26	53.07	20.05	53.17	20.98	52.12	19.20	52.13	16.68

Dataset	CAE		BUB		Dirichlet		Grassberger		Bayes	
	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes
car	94.67	74.07	94.57	72.78	94.67	71.24	92.48	60.73	94.52	69.13
iris	79.71	32.51	78.71	31.11	78.03	33.62	75.77	40.07	78.20	34.22
haberman	70.73	37.20	61.80	99.33	69.08	33.87	71.62	8.49	60.41	103.20
scale	75.34	49.11	77.72	71.78	75.54	54.78	75.68	45.44	76.40	57.44
cmc	44.22	208.73	43.72	342.44	44.17	324.84	46.52	110.98	43.49	439.64
wine	40.36	54.38	40.43	52.18	40.17	52.58	45.56	55.04	45.00	66.27
transfusion	75.50	75.89	72.09	170.22	75.47	68.44	74.66	19.22	72.94	141.22
hepatitis	76.80	21.56	76.80	21.67	78.05	20.56	77.99	18.11	78.05	20.67
soybean	79.54	43.89	73.27	44.11	78.91	42.78	82.19	43.22	78.80	46.33
forest	67.05	233.00	68.62	212.67	65.66	238.44	68.85	195.89	66.93	232.67
chess	98.96	54.16	98.89	51.93	98.95	50.64	90.25	105.98	98.68	50.22
monks-3	98.15	8.01	98.14	8.06	98.15	8.01	92.04	9.40	98.16	8.02
glass	34.55	125.56	35.78	115.22	39.08	110.33	37.85	40.22	37.23	129.00
bridges	49.23	39.67	54.78	35.44	50.90	31.89	47.91	24.33	51.90	34.67

Dataset	NSB		Shrinkage		bcMLE	
	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes
car	94.91	73.60	87.25	53.04	94.89	73.49
iris	78.63	28.16	78.98	28.16	78.37	28.40
haberman	65.55	35.13	68.76	13.82	67.94	30.91
scale	77.46	67.81	76.08	47.31	77.17	62.21
cmc	44.65	218.33	45.80	100.96	45.99	164.60
wine	39.37	29.91	35.62	10.58	38.24	27.78
transfusion	70.50	53.67	72.30	22.78	71.81	46.89
hepatitis	80.61	6.89	81.84	7.44	81.23	6.44
soybean	73.30	39.00	63.27	34.22	73.83	37.78
forest	65.24	87.56	64.03	72.33	65.90	93.67
chess	98.90	48.98	86.62	23.98	98.92	48.29
monks-3	98.15	8.10	98.15	7.80	97.86	7.80
glass	41.59	45.56	42.13	37.44	44.52	49.89
bridges	52.53	21.49	50.21	17.75	51.58	20.52

support size should be twice the number of class labels. Generally, JVHW performs much better on datasets with large number of classes (i.e. large support size) and small number of instances. On datasets with above properties, e.g. soybean, JVHW estimator brings up higher improvement.

For datasets with small S and large n, as the tree grows larger, the remaining sample size will

Table 3: Accuracy and number of nodes comparison of C4.5 decision tree between 13 estimators.

Dataset	MLE		JVHW		APML		Valiant		Jackknifed	
	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes
car	94.57	73.96	94.61	72.30	94.56	73.68	94.63	73.96	94.86	70.82
iris	77.09	31.67	77.97	31.13	77.15	31.47	78.42	31.98	81.23	30.64
haberman	70.80	39.83	71.29	30.22	71.19	39.23	69.84	31.89	71.12	25.47
scale	77.52	64.97	77.16	60.72	78.11	63.22	75.49	60.22	75.30	51.11
cmc	43.77	292.51	44.58	145.03	43.92	265.86	43.47	257.93	45.50	194.80
wine	37.60	57.32	37.74	54.71	37.95	53.71	40.23	54.69	37.85	39.87
transfusion	74.84	79.60	74.87	57.00	74.79	79.09	74.79	107.44	75.53	65.11
hepatitis	78.57	21.48	78.59	21.39	78.57	21.48	77.24	21.67	76.80	21.67
soybean	77.48	43.78	77.92	43.56	77.12	43.68	73.72	45.44	82.13	41.67
forest	67.45	227.14	68.16	216.72	67.33	226.19	68.58	227.11	65.03	214.00
chess	98.90	53.42	98.92	52.77	98.90	53.31	98.90	52.96	98.90	52.51
monks-3	98.17	8.06	98.19	8.03	98.17	8.04	98.14	8.05	98.16	8.02
glass	37.75	113.19	38.21	106.27	37.69	118.10	36.32	117.22	40.18	59.33
bridges	50.06	33.21	50.60	31.22	49.76	34.53	51.50	32.00	52.14	27.89

Dataset	CAE		BUB		Dirichlet		Grassberger		Bayes	
	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes
car	94.67	74.07	94.57	72.78	94.67	71.24	92.48	60.73	94.52	69.13
iris	79.71	32.51	78.71	31.11	78.03	33.62	75.77	40.07	78.20	34.22
haberman	70.73	37.20	61.80	99.33	69.08	33.87	71.62	8.49	60.41	103.20
scale	75.34	49.11	77.72	71.78	75.54	54.78	75.68	45.44	76.40	57.44
cmc	44.22	208.73	43.72	342.44	44.17	324.84	46.52	110.98	43.49	439.64
wine	40.36	54.38	40.43	52.18	40.17	52.58	45.56	55.04	45.00	66.27
transfusion	75.50	75.89	72.09	170.22	75.47	68.44	74.66	19.22	72.94	141.22
hepatitis	76.80	21.56	76.80	21.67	78.05	20.56	77.99	18.11	78.05	20.67
soybean	79.54	43.89	73.27	44.11	78.91	42.78	82.19	43.22	78.80	46.33
forest	67.05	233.00	68.62	212.67	65.66	238.44	68.85	195.89	66.93	232.67
chess	98.96	54.16	98.89	51.93	98.95	50.64	90.25	105.98	98.68	50.22
monks-3	98.15	8.01	98.14	8.06	98.15	8.01	92.04	9.40	98.16	8.02
glass	34.55	125.56	35.78	115.22	39.08	110.33	37.85	40.22	37.23	129.00
bridges	49.23	39.67	54.78	35.44	50.90	31.89	47.91	24.33	51.90	34.67

Dataset	NSB		Shrinkage		bcMLE	
	Accuracy	Nodes	Accuracy	Nodes	Accuracy	Nodes
car	94.66	73.82	87.07	51.27	94.70	72.73
iris	78.52	32.38	78.61	28.16	78.79	32.80
haberman	69.39	47.00	69.56	28.20	69.90	32.42
scale	77.51	65.44	76.42	46.89	77.80	60.11
cmc	43.74	299.18	42.56	142.60	43.38	257.91
wine	42.98	53.11	33.02	13.56	40.23	54.69
transfusion	74.48	81.44	75.50	50.67	75.53	64.89
hepatitis	76.80	21.67	77.70	20.67	76.80	21.67
soybean	76.80	42.33	66.63	34.78	83.12	45.33
forest	68.30	225.44	53.40	174.00	67.99	226.67
chess	98.92	53.69	86.17	28.40	98.90	53.27
monks-3	98.17	8.19	98.15	7.80	98.15	8.01
glass	36.89	112.78	35.54	56.11	37.30	115.33
bridges	56.29	35.33	44.05	25.56	49.33	34.52

become smaller. And we will be eventually in the data scarce regime. Even when n and S are both very small and n is comparable to S, JVHW estimator still works quite well. This explains the reason that JVHW remains its superiority for abundant instances and small S, like iris and haberman.

Table 4: P-values for rejecting the null hypothesis that one certain entropy estimator performs equally as MLE estimator using permutation test.

Estimator	ID3	C4.5
JVHW	0.0126	0.0012
APML	0.2954	0.1961
Valiant	0.9250	0.5936
Jackknife	0.2209	0.1771
CAE	0.6832	0.9249
BUB	0.1578	0.5098
Dirichlet	0.3627	0.5098
Grassberger	0.5098	0.6378
Bayes	0.6378	0.5098
NSB	0.6832	0.8613
Shrinkage	0.0843	0.0063
bcMLE	0.3305	0.6378

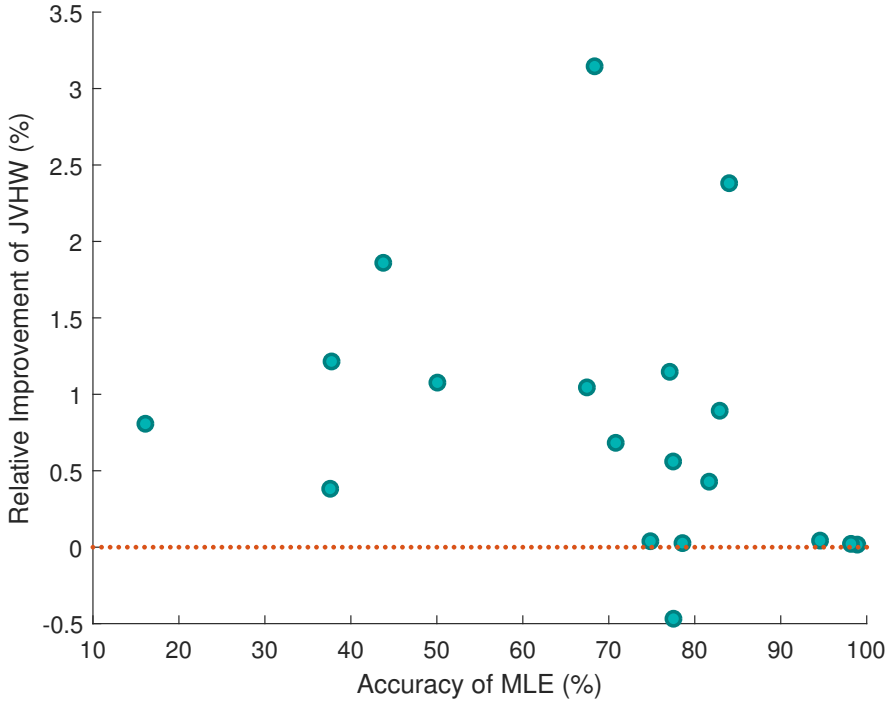

Figure 1: The relative accuracy improvement of JVHW compared with MLE for C4.5 decision tree. Here x-axis is the accuracy for MLE-based decision tree, and y-axis is the relative improvement, i.e. $(Accuracy_{JVHW} - Accuracy_{MLE})/Accuracy_{MLE}$.

To summarize, JVHW estimator works well not only in the asymptotic case when n and S are both very large and n is comparable to S, but even in the small n, S regime. It is a very valuable property for practice.

5 Conclusion

In this paper, we propose the use of minimax rate-optimal estimators to estimate the splitting scoring function in decision tree learning. We extensively experimented with the classification trees using information gain and gain ratio as split scoring function. With the minimax rate-optimal estimator, one can build up the decision tree more precisely with less nodes and the same time complexity. The code is available upon request.

References

[1] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[2] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[3] Zhi-Hua Zhou and Ji Feng. Deep forest: Towards an alternative to deep neural networks. *CoRR*, abs/1702.08835, 2017.

[4] Hyafil Laurent and Rivest Ronald. Constructing optimal binary decision tress is np-complete. *Information Processing Letters*, 1976.

[5] Antonio Criminisi, Jamie Shotton, Ender Konukoglu, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[6] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[7] J. Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[8] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[9] Wray Buntine and Tim Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8(1):75–85, 1992.

[10] Wei Zhong Liu and Allan P. White. The importance of attribute selection measures in decision tree induction. *Machine Learning*, 15(1):25–41, 1994.

[11] Sebastian Nowozin. Improved information gain estimates for decision tree induction. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 571–578. Omnipress, 2012.

[12] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

[13] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *IEEE Transactions on Information Theory*, 61(5):2835–2885, 2015.

[14] Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log(n)-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694. ACM, 2011.

[15] Gregory Valiant and Paul Valiant. The power of linear estimators. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 403–412. IEEE, 2011.

[16] Paul Valiant and Gregory Valiant. Estimating the unseen: improved estimators for entropy and other properties. In *Advances in Neural Information Processing Systems*, pages 2157–2165, 2013.

[17] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016.

[18] M. Lichman. UCI machine learning repository, 2013.

[19] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[20] G. Miller. Note on the bias of information estimates. In *Information Theory in Psychology; Problems and Methods Ii-B*, 1955.

[21] Rupert G Miller. The jackknife-a review. *Biometrika*, 61(1):1–15, 1974.

[22] Anne Chao and Tsung Jen Shen. Nonparametric estimation of shannon's index of diversity when there are unseen species in sample. *Environmental and Ecological Statistics*, 10(4):429–443, 2003.

[23] Jean Hausser and Korbinian Strimmer. Entropy inference and the james-stein estimator, with application to nonlinear gene association networks. *Statistical Applications in Genetics and Molecular Biology*, 10(3):1469–1484, 2009.

[24] P. Grassberger. Entropy estimates from insufficient samplings. *Physics*, 2003.

[25] S Schober. Some worst-case bounds for bayesian estimators of discrete distributions. In *IEEE International Symposium on Information Theory Proceedings*, pages 2194–2198, 2013.

[26] David R. Wolf and David H. Wolpert. Estimating functions of probability distributions from a finite set of samples. 1993.

[27] Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited. *Adv.neural Inf.proc.syst*, pages 471–478, 2001.

[28] Jiantao Jiao Dmitri Pavlichin and Tsachy Weissman. Approximate profile maximum likelihood.

A Proof of Theorem 1

For the first claim, let $\hat{I}(A_i;C)$ be any minimax rate-optimal estimator in estimating the mutual information $I(A_i;C)$. Now define our algorithm $\hat{i}(\mathcal{D})$ as follows:

$$\hat{i}(\mathcal{D}) = \arg\max_{i\in\{1,2\}} \hat{I}(A_i;C). \tag{7}$$

Since $n \gg \frac{|\mathcal{A}||\mathcal{C}|}{\log|\mathcal{A}|+\log|\mathcal{C}|}$, by [17, 13] we know that for any $i \in \{1,2\}$ and any $\epsilon > 0$, we have

$$\sup_P \mathbb{P}(|\hat{I}(A_i;C) - I(A_i;C)| > \epsilon) \to 0 \tag{8}$$

as $n \to \infty$. Hence, by the triangle inequality and applying the union bound, we have

$$\begin{aligned}
&\sup_P \mathbb{P}(I(C;A_{\hat{i}}) < I(C;A_{3-\hat{i}}) - \delta) \\
&\quad\le \sup_P \mathbb{P}(|\hat{I}(A_1;C) - I(A_1;C)| > \frac{\delta}{2}) + \sup_P \mathbb{P}(|\hat{I}(A_2;C) - I(A_2;C)| > \frac{\delta}{2}) &(9)\\
&\quad\to 0, &(10)
\end{aligned}$$

as desired.

For the second claim on the plug-in approach, the achievability part follows from the same argument and the result in [13] that when $n \gg |\mathcal{A}||\mathcal{C}|$, we have

$$\sup_P \mathbb{P}(|\hat{I}_{\text{plug-in}}(A_i;C) - I(A_i;C)| > \epsilon) \to 0 \tag{11}$$

for any $\epsilon > 0$ as $n \to \infty$. For the converse part, without loss of generality we may assume that $n \ge 15|\mathcal{A}||\mathcal{C}| \gg \max\{|\mathcal{A}|,|\mathcal{C}|\}$, for more samples will make the plug-in estimator become more accurate. As a result, [13] shows that for any $i \in \{1,2\}$ and $\epsilon > 0$, we have

$$\sup_P \max\{\mathbb{P}(|\hat{H}_{\text{plug-in}}(A_i) - H(A_i)| > \epsilon), \mathbb{P}(|\hat{H}_{\text{plug-in}}(C) - H(C)| > \epsilon)\} \to 0 \tag{12}$$

as $n \to \infty$. Now consider two discrete distributions:

$$P_1 = (\frac{1}{|\mathcal{A}|}, \frac{1}{|\mathcal{A}|}, \cdots, \frac{1}{|\mathcal{A}|}) \tag{13}$$

$$P_2 = (\frac{\eta}{|\mathcal{A}|}, \frac{\eta}{|\mathcal{A}|}, \cdots, \frac{\eta}{|\mathcal{A}|}, 0, \cdots, 0) \tag{14}$$

where $\eta > 1$ is some parameter to be chosen later. Let the joint distribution $P_{CA_1A_2}(c,a_1,a_2) = P_0(c)P_1(a_1)P_2(a_2)$ be mutually independent, where $P_0(\cdot)$ is the uniform distribution on $\mathcal{C}$, we have $I(C,A_1) = I(C,A_2) = 0$. Moreover, by [13, Lemma 5, Lemma 15], we know that

$$H(A_1,C) - \mathbb{E}\hat{H}_{\text{plug-in}}(A_1,C) \ge \frac{|\mathcal{A}||\mathcal{C}|-1}{2n} + \frac{1}{n^2}\left(\frac{|\mathcal{A}|^2|\mathcal{C}|^2}{20} - \frac{1}{12}\right), \tag{15}$$

$$|\mathbb{E}\hat{H}_{\text{plug-in}}(A_2,C) - H(A_2,C)| \le \frac{5|\mathcal{A}||\mathcal{C}|\log 2}{n\eta}. \tag{16}$$

Since the variance of the plug-in approach is of the order $O(\frac{(\ln n)^2}{n})$ [13], the previous inequalities imply that with high probability, we have

$$\hat{H}_{\text{plug-in}}(A_1, C) \leq H(A_1, C) - \frac{|\mathcal{A}||\mathcal{C}|}{4n}, \tag{17}$$

$$\hat{H}_{\text{plug-in}}(A_2, C) \geq H(A_2, C) - \frac{10|\mathcal{A}||\mathcal{C}|\log 2}{n\eta}. \tag{18}$$

As a result, choosing $\eta = 80\log 2$ and $\epsilon = \frac{|\mathcal{A}||\mathcal{C}|}{40n}$ in (12), with high probability we have

$$\begin{aligned}
\hat{I}_{\text{plug-in}}(A_1, C) &= \hat{H}_{\text{plug-in}}(A_1) + \hat{H}_{\text{plug-in}}(C) - \hat{H}_{\text{plug-in}}(A_1, C) && (19)\\
&\geq H(A_1) - \epsilon + H(C) - \epsilon - H(A_1, C) + 10\epsilon && (20)\\
&= 8\epsilon && (21)\\
&= H(A_2) + H(C) - H(A_2, C) + 8\epsilon && (22)\\
&\geq \hat{H}_{\text{plug-in}}(A_1) - \epsilon + \hat{H}_{\text{plug-in}}(C) - \epsilon - (\hat{H}_{\text{plug-in}}(A_1, C) + 4\epsilon) + 8\epsilon && (23)\\
&= \hat{I}_{\text{plug-in}}(A_2, C) + 2\epsilon && (24)
\end{aligned}$$

i.e., although the true mutual information is zero in both cases, with high probability the plug-in approach $\hat{I}_{\text{plug-in}}(A_1, C)$ will be larger than $\hat{I}_{\text{plug-in}}(A_2, C)$, with a constant additive gap $\epsilon = \frac{|\mathcal{A}||\mathcal{C}|}{40n} \asymp 1$ since $n \lesssim |\mathcal{A}||\mathcal{C}|$. Now by disturbing the joint distribution $P_{CA_1A_2}$ by a small constant amount, we can construct examples with $I(A_1, C) \leq I(A_2, C) - \Theta(1)$, while $\hat{I}_{\text{plug-in}}(A_1, C) \geq \hat{I}_{\text{plug-in}}(A_2, C)$ with a strictly positive probability. In this case, the plug-in approach fails with a positive probability, as desired.