Improving Decision Trees by Tsallis Entropy Information Metric Method

Yisen Wang*†‡, Chaobing Song†, Shu-Tao Xia*†
*Department of Computer Science and Technology, Tsinghua University, Beijing, China
†Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
‡ Tsinghua National Laboratory for Information Science and Technology, Beijing, China
Email: wangys14@mails.tsinghua.edu.cn, songchaobin@126.com, xiast@sz.tsinghua.edu.cn

***Abstract*—The construction of efficient and effective decision trees remains a key topic in machine learning because of their simplicity and flexibility. A lot of heuristic algorithms have been proposed to construct near-optimal decision trees. Most of them, however, are greedy algorithms that have the drawback of obtaining only local optimums. Besides, conventional split criteria they used, e.g. Shannon entropy, Gain Ratio and Gini index, cannot select informative attributes efficiently. To address the above issues, we propose a novel Tsallis Entropy Information Metric (TEIM) algorithm with a new split criterion and a new construction method of decision trees. Firstly, the new split criterion is based on two terms of Tsallis conditional entropy, which is better than conventional split criteria. Secondly, the new construction method is based on a two-stage approach that avoids local optimum to a certain extent. The TEIM algorithm takes advantages of the generalization ability of Tsallis entropy and the low greediness property of two-stage approach. Experimental results on UCI datasets indicate that, compared with the state-of-the-art decision trees algorithms, the TEIM algorithm yields statistically significantly better decision trees in classification accuracy as well as tree complexity.**

I. Introduction

The decision trees method is a non-parametric supervised learning method used for classification and regression. Although the decision trees method has been one of the first machine learning approaches, it remains an actively researched domain in machine learning. It is not only simple to understand and interpret, but also offers relatively good results, efficiency and flexibility. The general idea of decision trees is to predict unknown input instances by learning simple decision rules inferred from several known training instances.

Generally speaking, split criterion and construction method of decision trees are two fundamental issues in decision trees. A large number of decision trees algorithms have been proposed based on different split criteria. For example, the Iterative Dichotomiser 3 (ID3) algorithm is based on Shannon entropy [1]; the C4.5 algorithm is based on Gain Ratio [2]; while the Classification And Regression Tree (CART) algorithm is based on Gini index [3]. However, among these algorithms, no one algorithm always gets the best results on various datasets. Actually, it reflects one drawback of this kind of split criteria that they cannot select informative attributes efficiently in the tree construction procedure. Numerous alternatives have been proposed to address this issue, for example, the adaptive entropy estimate [4], [5], but their statistical entropy estimates are too complicated to lose the simplicity and comprehensibility of decision trees. In addition, a series of papers have analyzed the importance of the split criterion [6], [7]. They demonstrate that different split criteria have a substantial influence on the generalization error of the decision trees.

Meanwhile, the optimal construction of decision trees has been theoretically proven to be NP-complete [8], [9]. Consequently, most practical implementations of decision trees use greedy algorithms to grow trees. Such approaches, however, suffer from the flaws of local optimums. Several alternatives have been proposed to overcome the issue. The ID3 algorithm with lookahead technique is presented in [10], but its complexity increases exponentially, as the degree of lookahead grows. Another alternative method to lookahead is the skewing technique [11], but it can only apply to datasets with less than 7 attributes. Evolutionary Algorithms (EAs) are another kinds of alternatives which replace the local search with the global search to escape from the local optimum [12], but their disadvantages are also obvious such as time-consuming computation and a large number of parameters. Dual information metric [13] is another method to reduce greediness, but its classification accuracy is worse than C4.5.

To address the above two issues, inspired by the Tsallis entropy split criterion in [14], we propose a novel Tsallis Entropy Information Metric (TEIM) algorithm with a new split criterion and a new construction method of decision trees in this paper. In [14], Tsallis entropy split criterion simply replaces the three common split criteria and is only explored as one term formula, but the further discussion of Tsallis entropy split criterion is not presented, and the construction of decision trees is also in a greedy manner. However, in this paper, we design a new split criterion M_q based on two terms of Tsallis entropy, i.e. the summation of two symmetrical Tsallis conditional entropies. Also, we propose a two-stage approach to make the construction of trees less greedy. Empirical results show that the TEIM algorithm is indeed an efficient and effective construction method of decision trees.

The rest of this paper is organized as follows. Section II presents the background of Tsallis entropy framework. Section III describes the proposed TEIM algorithm. Section IV exhibits experimental results. Section V summarizes the work.

978-1-5090-0620-5/16/$31.00

II. TSALLIS ENTROPY FRAMEWORK

Tsallis entropy $S_q(X)$ is one kind of generalization of Shannon entropy adding one more adjustable parameter q [15], which is defined by:

$$S_q(X) = \frac{1}{1-q}\left(\sum_{i=1}^{n} p(x_i)^q - 1\right), \quad q \in \mathbb{R}, \tag{1}$$

where X is a random variable taking values $\{x_1, ..., x_n\}$ and $p(x_i)$ is the corresponding probability of x_i. For $q < 0$, Tsallis entropy is convex. For $q = 0$, Tsallis entropy is non-convex and non-concave. For $q > 0$, Tsallis entropy is concave [16].

With respect to Shannon entropy $H(X)$ proposed in [17], it is a measure of the uncertainty associated with a random variable X:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \ln p(x_i). \tag{2}$$

Tsallis entropy converges to Shannon entropy when $q \to 1$:

$$\begin{aligned}\lim_{q\to 1} S_q(X) &= \lim_{q\to 1} \frac{1}{1-q}\left(\sum_{i=1}^{n} p(x_i)^q - 1\right) \\ &= -\sum_{i=1}^{n} p(x_i) \ln p(x_i) \\ &= H(X). \end{aligned} \tag{3}$$

Besides, Tsallis entropy also covers Gini index and Gain Ratio with specific values of q. For a more comprehensive review we refer the reader to [14].

Moreover, Tsallis entropy has some properties similar to Shannon entropy. For instance, for $q > 0$, $S_q \geq 0$ and S_q is maximal at the uniform distribution $p(x_i) = 1/n, i = 1, 2, ..., n$. The relation to Shannon entropy can be made clearer by rewriting the definition in the form:

$$S_q(X) = -\sum_{i=1}^{n} p(x_i)^q \ln_q p(x_i), \tag{4}$$

where

$$\ln_q(x) = \frac{x^{1-q} - 1}{1-q}, \quad q \neq 1, x \geq 0 \tag{5}$$

is called the q-logarithmic function [18]. And when $q \to 1$, $\ln_q(x) \to \ln(x)$.

The reason behind the proposition of Tsallis entropy is to characterize and explain some physical systems that have complex behaviors such as long-range and long-memory interactions [19]. To be specific, data across a variety of domains exhibit a property known as the heavy tail in reality [20]. However, we cannot characterize power-law heavy-tailed distribution through maximizing Shannon entropy subject to normal mean and variance [21], [22]. The reason is that Shannon entropy implicitly assumes a certain trade-off between contributions from the tails and the main mass of distribution [23]. It should be worthwhile to control this trade-off explicitly to characterize the power-law heavy-tailed distribution family. Entropy measures that depend on powers of probability, e.g. $\sum_{i=1}^{n} p(x_i)^q$, can provide such control. Thus, some parameterized entropies have been proposed [24], [15]. A well-known generalization of this concept is Tsallis entropy.

Besides, Tsallis conditional entropy and Tsallis joint entropy are also derived similarly to Shannon entropy. For the conditional probability $p(x|y) = p(X = x|Y = y)$ and the joint probability $p(x, y) = p(X = x, Y = y)$, Tsallis conditional entropy and Tsallis joint entropy [25] are denoted by:

$$S_q(X|Y) = -\sum_{x,y} p(x,y)^q \ln_q p(x|y), (q \neq 1) \tag{6}$$

$$S_q(X,Y) = -\sum_{x,y} p(x,y)^q \ln_q p(x,y), (q \neq 1). \tag{7}$$

It is remarkable that (6) can be easily reformed by

$$S_q(X|Y) = \sum_{y} p(y)^q S_q(X|y). \tag{8}$$

The relation between the conditional entropy and joint entropy is given by:

$$S_q(X,Y) = S_q(X) + S_q(Y|X). \tag{9}$$

III. TSALLIS ENTROPY INFORMATION METRIC (TEIM) ALGORITHM

In this section, we describe the proposed Tsallis Entropy Information Metric (TEIM) algorithm with a new split criterion and a new construction method of decision trees.

A. The New Information Metric M_q

Given a dataset $\mathcal{D}_n$ with n instances, each instance $(\boldsymbol{X}, Y)$ has attributes A_j $(j \in \{1, 2, \ldots, d\})$ and class label $Y \in \{1, 2, \ldots, K\}$. One key issue in the procedure of decision tree construction is the split criterion. At every step, the decision tree chooses one pair of attribute and cutting point which make the maximal impurity decrease to split the data and grow the tree. Therefore, the pair of attribute and cutting point chosen to split significantly affect the structure of decision trees and further influence the classification performance.

Compared with one term formula of Tsallis entropy criterion in [14], we propose a new information metric M_q in a two terms formula, i.e. the summation of two symmetrical Tsallis conditional entropies. The metric M_q between attribute A and class label Y is defined as follows:

$$M_q(A, Y) = S_q(Y|A) + S_q(A|Y), \tag{10}$$

where S_q is Tsallis entropy and the parameter q can be adjusted for datasets. S_q degenerates to H (Shannon entropy) when $q = 1$. From the definition, we find that in order to obtain maximal impurity decrease one need to minimize the M_q between attributes and class labels. Besides, it is important to note that M_q follows the required mathematical properties of a metric [26], namely:

For $q > 0$, M_q satisfies: (E, F, G are random variables)

(i)

$$M_q(E, F) = 0 \text{ iff } E = F,$$

(ii)

$$M_q(E,F) = M_q(F,E),$$

(iii)

$$M_q(E,G) \leq M_q(E,F) + M_q(F,G).$$

M_q has the symmetrical formula, i.e. the summation of two Tsallis conditional entropies. Unlike other split criteria of decision trees, M_q takes into account two terms for attribute selection. The logic behind this is less explicit but can be well understood through a small illustrative example in Table I. Let us look at the following six instances dataset that consists of two input attributes, A_1 and A_2. A_1 has 6 values; A_2 has 2 values and class label Y has 2 values. To be simplified, we assume $q = 1$, then S_q converges to H. As clearly seen, attribute A_1 or A_2 can classify the class completely, so $H(Y|A_1) = 0$ and $H(Y|A_2) = 0$. However, attributes A_1 and A_2 are not identified. A_1 partitions the dataset into six subsets while A_2 partitions the dataset into two subsets. The difference is reflected in the second conditional entropy of M_q, $H(A_1|Y) = 1.58$ and $H(A_2|Y) = 0$. By aiming to minimize the M_q, one will prefer attribute A_2 to A_1. In fact, for a binary split, attribute A_2 only needs one split to classify the dataset completely, however, attribute A_1 needs multiple splits. In the aspect of tree complexity, attribute A_2 is exactly better than A_1.

Popular algorithms of decision trees, such as ID3 or TE/TGR algorithm [14], take into account the uncertainty $S_q(Y|A)$ in the class label Y following the selection of attribute A. That is to say, they only consider the first term $S_q(Y|A)$ in M_q. Note from (10) that our proposed metric M_q considers not only $S_q(Y|A)$ but also $S_q(A|Y)$. In the above example, ID3 will choose the attribute A_1 or A_2 randomly to split, while M_q will choose the attribute A_2. The small illustrative example shows that M_q with the formula of two terms is better than the original one term split criteria. M_q prefers to choose fewer but efficient attributes that partition the dataset as close as possible to the class while avoiding unnecessary splits. It reduces the complexity of decision trees to a certain extent.

TABLE I
ILLUSTRATION EXAMPLE FOR M_q

A_1	A_2	Y
1	1	*
2	1	*
3	1	*
4	2	O
5	2	O
6	2	O

B. Split Attribute Evolution Criterion

Although the optimal construction of decision trees is NP-complete, the efficient construction of near-optimal decision trees remains an open issue. In this paper, we propose a two-stage approach for efficient construction of decision trees.

As stated in the subsection III-A, M_q is a metric to measure the distance between random variables. The two-stage approach is a maximal-orthogonality-maximal-relevance method for tree construction using M_q criterion. Maximal-orthogonality refers to the maximal orthogonality between the attributes, and maximal-relevance refers to the maximal relevance between the attributes and class labels. That is to say, in the procedure of attribute selection, the two-stage approach takes into consideration not only the immediate contributions to the classification but also the previous potential effects of attributes. Assuming the previous one step selected attribute is A_e and the current to be selected attribute is A_u, the object of the two-stage approach is to select the best attribute A_u which minimizes $L(A_u)$:

$$\begin{aligned} L(A_u) &= M_q(A_u, Y) - M_q(A_e, A_u) \\ &= S_q(A_u|Y) + S_q(Y|A_u) - S_q(A_e|A_u) - S_q(A_u|A_e). \end{aligned} \tag{11}$$

In order to minimize $L(A_u)$, we need to minimize the first term $M_q(A_u, Y)$ and maximize the second term $M_q(A_e, A_u)$ due to the fact that M_q is non-negative. Minimizing $M_q(A_u, Y)$ is synonymous to maximizing the relevance between the currently selected attribute and class label. The higher relevance between attributes and class labels, the more information of class labels that attributes can provide. And, maximizing $M_q(A_e, A_u)$ is identical to maximizing the orthogonality between the currently selected attribute A_u and the previous one step selected attribute A_e. The higher orthogonality among attributes leads that the two-stage approach chooses fewer, less redundant but efficient attributes to construct decision trees. In summary, the two-stage approach prefers the attribute A_u which has the maximal orthogonality to the previous attribute A_e and maximal relevance to the class label Y at the same time.

C. Tree Construction

Given the dataset $\mathcal{D}_n$ with n instances, each instance $(\boldsymbol{X}, Y)$ has attributes A_j $(j \in \{1, 2, \ldots, d\})$ and class label $Y \in \{1, 2, \ldots, K\}$. For each tree node, we search for every possible pair of attribute and cutting point to choose the optimal attribute and cutting point to grow the tree. Without loss of generality, similar to CART, we also choose the binary split manner. For an attribute A_j, we obtain

$$\begin{aligned} L(A_j(C_j)) =& S_q(A_j(C_j)|Y) + S_q(Y|A_j(C_j)) \\ &- S_q(A_e(C_e)|A_j(C_j)) - S_q(A_j(C_j)|A_e(C_e)), \end{aligned} \tag{12}$$

where $A_j(C_j)$ denotes the candidate pair of attribute as well as cutting point to be selected and $A_e(C_e)$ is the previously selected pair. Assuming D is the dataset belonging to one node to be partitioned, and then D' and D'' are two child nodes that would be created if D is partitioned by $A_j(C_j)$. The pair of attribute A_j as well as cutting point C_j which minimizes $L(A_j(C_j))$ is chosen to construct the tree.

The above procedure is applied recursively until some stopping conditions are reached. The stopping conditions consist of

three principles: (i) The classification is achieved in a subset. (ii) No attributes are left for selection. (iii) The cardinality of a subset is lower than the predefined threshold.

Once the tree has been trained by the data as a classifier g, it can be used to predict for new unlabeled instances. The decision tree makes the prediction in a majority voting manner. For the unlabeled instance $\boldsymbol{x}$, the probability of each class $k \in \{1, 2, \ldots, K\}$ is

$$\eta^{(k)}(\boldsymbol{x}) = \frac{1}{N(A(\boldsymbol{x}))} \sum_{(\boldsymbol{X},Y)\in A(\boldsymbol{x})} \mathbb{I}(Y = k), \qquad (13)$$

where $A(\boldsymbol{x})$ denotes the leaf containing $\boldsymbol{x}$ and $N(A(\boldsymbol{x}))$ denotes the number of instances in $A(\boldsymbol{x})$. $\mathbb{I}(e)$ is the indicator function that takes 1 if e is true and 0 for other cases. Then the tree prediction $\hat{y}$ is the class that maximizes this value:

$$\hat{y} = g(\boldsymbol{x}) = \arg\max_k \{\eta^{(k)}(\boldsymbol{x})\}. \qquad (14)$$

D. TEIM Algorithm

Here, we summarize our proposed Tsallis Entropy Information Metric (TEIM) algorithm in a pseudo-code format in Algorithm 1.

Algorithm 1 TEIM algorithm

1: **Input**: Data $\mathcal{D}_n$, Attributes $A \in \mathbb{R}^d$, Class Y
2: **Output**: A decision tree
3: Initialize $A_e(C_e) = \arg\min_j M_q(A_j, Y), A_j \in A$
4: **while** not satisfying the stop condition **do**
5: **for** each attribute A_j **do**
6: $S \leftarrow domain(A_j)$
7: // S is the candidate cutting point set of A_j
8: // C_j is one cutting point in the set S
9: **for** each $C_j \in S$ **do**
10: $D' \leftarrow \{\boldsymbol{X} \in D_n | A_j(\boldsymbol{X}) \leq C_j\}$
11: $D'' \leftarrow \{\boldsymbol{X} \in D_n | A_j(\boldsymbol{X}) > C_j\}$
12: // $(\boldsymbol{X}, Y)$ is one instance in Data $\mathcal{D}_n$
13: // D', D'' are the two child nodes
14: Compute $L(A_j(C_j))$ according to (12)
15: **end for**
16: **end for**
17: $A_u(C_u) \leftarrow \arg\min L(A_j(C_j))$
18: // $A_u(C_u)$ is the best pair of split attribute and cutting point
19: Grow the tree using $A_u(C_u)$ and partition the data using the binary split
20: $A_e(C_e) \leftarrow A_u(C_u)$
21: Go to line 4 for D' and D''
22: // Recursively repeat the procedure and the stop condition is presented in subsection III-C
23: **end while**
24: **Return** A decision tree
25: // Tree is built by nodes from the root to the leaf

Taking the influence of previous attributes and class labels into account, the TEIM algorithm with two-stage approach indeed reduces the greediness effect in the construction of decision trees. Besides, the TEIM algorithm adopts a better criterion M_q than original Tsallis entropy criterion. Moreover, the parameter q in M_q can be tuned depending on datasets for better adaptability and flexibility. Thus, the TEIM algorithm enables constructing decision trees with better adaptability, better robustness, higher accuracy and lower complexity.

IV. Experiments

In this section, we conduct a series of experiments to exhibit the performance enhancement of TEIM algorithm in classification accuracy and tree complexity.

A. Evaluation Metric

In order to quantitatively compare the trees obtained by different methods, we employ accuracy to evaluate the effectiveness of the tree and the total number of the tree nodes to measure the tree complexity.

B. Datasets

As shown in Table II, the 6 UCI datasets [27] are used to evaluate the proposed algorithm. These datasets consist of three types, namely numeric, categorical and mixed datasets. Also, the instance size and class number are various, which are sufficiently representative to demonstrate the performance of TEIM.

TABLE II
Detailed information of the UCI data sets

Dataset	Type	No. of instance	No. of attribute	No. of class
Yeast	numeric	1484	8	10
Glass	numeric	214	10	7
Haberman	numeric	306	3	2
Scale	categorical	625	4	3
Monks	categorial	432	7	2
Abalone	mixed	4139	8	18

C. Experimental Setup

The TEIM algorithm combines advantages of the Tsallis information metric M_q and two-stage approach, which can reduce the greediness in the construction of decision trees and enhance the performance. Thus, we conduct a series of experiments on datasets in Table II to test the performance of the proposed TEIM algorithm. With respect to the algorithms for comparison, we choose the state-of-the-art decision trees with Tsallis Entropy (TE) and Tsallis Gain Ratio (TGR) criteria in [14] which achieve better performance than ID3, C4.5 and CART. TE prefers high accuracy while TGR prefers low complexity. Intuitively, they both have strengths as well as weaknesses, and no one can beat another one absolutely, but the TEIM algorithm combines their strengths together to obtain better decision trees both in classification accuracy and tree complexity.

The decision trees with Tsallis entropy (TE) and Tsallis Gain Ratio (TGR) criteria are also implemented in Python. In each dataset, we do a grid search using 10-fold cross-validation

TABLE III
CLASSIFICATION ACCURACY OF TE, TGR AND TEIM ON DIFFERENT DATASETS

Dataset	TE		TGR		TEIM		
	Accuracy (%)	No. of nodes	Accuracy (%)	No. of nodes	Accuracy (%)	No. of nodes	optimal q
Yeast	56.9	195.8	51.2	197.1	**59.8**	**94.0**	1.4
Glass	60.6	52.6	53.1	51.5	**62.9**	**27.6**	2.6
Haberman	74.2	33.2	74.8	**33.0**	**75.2**	36.4	7.1
Scale	78.2	93.1	78.5	77.0	**82.2**	**75.0**	3.1
Monks	57.3	89.6	54.9	**88.0**	**60.9**	89.8	8.9
Abalone	26.8	86.2	25.7	85.1	**27.2**	**70.8**	0.8

to determine the values of q in M_q. Maybe the optimal q for TE, TGR or M_q is different, but for the fair comparison, we adopt the same q, e.g. optimal q for M_q. Besides, the minimum leaf size is set to 5 to avoid overfitting.

D. The Analysis of Results

Table III reports the performance of several decision trees algorithms. The highest accuracy and lowest complexity on each dataset are in boldface. As expected, the performance of TEIM outperforms TE and TGR both in classification accuracy and tree complexity. It is worth mentioning that TE and TGR have been demonstrated to outperforms the classical algorithms ID3, C4.5 and CART [14]. In this case, our proposed TEIM algorithm can also perform better than those traditional decision tree algorithms. Focusing on the tree complexity firstly, TEIM achieves the smaller tree on 4 out of 6 datasets compared with TE and TGR. The remarkable decline in tree complexity is indicated on the Yeast dataset whose number of tree nodes decrease from 197.1 to 94.0. The decreases in tree complexity are found by Wilcoxon signed ranked test [28] to be significant ($p = 0.0068$). The decrease of complexity is caused by the metric M_q and two-stage approach in the TEIM algorithm, which is similar to C4.5 (the normalization factor in Gain Ratio leads to the decrease in tree complexity). The advantages of TEIM, e.g. reducing tree complexity and constructing smaller trees, are beneficial for the online classification in Big Data environments [29], e.g. health-care systems.

Reducing the tree complexity is usually beneficial on condition that the associated accuracy is not substantially reduced. Comparing to the classification accuracy of TE and TGR, we find that TEIM is better than them on all the datasets. The TEIM algorithm uses M_q as a split criterion that is based on the summation of two Tsallis conditional entropies, while the TE or TGR algorithm uses one term of Tsallis entropy directly. As discussed before, the split criterion M_q is better than Tsallis entropy. Besides, the two-stage approach is a maximal-orthogonality-maximal-relevance method considering the previously selected attributes and class labels, which is also responsible for the improvement of the accuracy. Moreover, TEIM also benefits from Tsallis entropy and its adjustable parameter q. Therefore, the TEIM algorithm can achieve better performance due to the fact that it combines all the advantages together.

Another difference between TE/TGR and TEIM algorithms is the tree construction method. The TE/TGR algorithm still uses the classical greedy construction method of decision trees, while the TEIM algorithm utilizes the two-stage method to reduce the greediness. So the TEIM algorithm can select fewer but efficient attributes for the tree construction. Thus, the TEIM algorithm can construct smaller trees while maintaining higher accuracy.

Regarding the optimal value of q, though it is obtained by cross-validation in this paper, we find a fuzzy trend from Table III. The more of class number, the smaller value of q, e.g. for numeric type datasets from Yeast to Haberman, q is increasing while the class number is decreasing. Experimental results show that the optimal q obtained by cross-validation is usually not equal to 1 or 2 which implies the better performance through tuning the parameter q. Although the optimal q may be different for different datasets, we conjecture that the value of q is associated with the properties of datasets.

V. CONCLUSIONS

In this paper, we address two fundamental issues of decision trees, i.e. the split criterion and tree construction. We define a new split criterion M_q with the summation of two Tsallis conditional entropies, and propose a new construction method of decision trees with the two-stage approach. Combining all the strengths of Tsallis entropy, M_q and two-stage method together, a novel decision tree algorithm, i.e. Tsallis Entropy Information Metric (TEIM) algorithm, is proposed. Empirically, the TEIM algorithm promotes the performance of decision trees in accuracy and complexity, compared with the state-of-the-art decision trees algorithms.

ACKNOWLEDGMENTS

This research is supported in part by the Major State Basic Research Development Program of China (973 Program, 2012CB315803), the National Natural Science Foundation of China (61371078), and the Research Fund for the Doctoral Program of Higher Education of China (20130002110051).

REFERENCES

[1] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
[2] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[3] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[4] S. Nowozin, "Improved information gain estimates for decision tree induction," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. ACM, 2012, pp. 297–304.
[5] M. Serrurier and H. Prade, "Entropy evaluation based on confidence intervals of frequency estimates: Application to the learning of decision trees," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. ACM, 2015, pp. 1576–1584.
[6] W. Buntine and T. Niblett, "A further comparison of splitting rules for decision-tree induction," *Machine Learning*, vol. 8, no. 1, pp. 75–85, 1992.
[7] W. Z. Liu and A. P. White, "The importance of attribute selection measures in decision tree induction," *Machine Learning*, vol. 15, no. 1, pp. 25–41, 1994.
[8] L. Hyafil and R. L. Rivest, "Constructing optimal binary decision trees is np-complete," *Information Processing Letters*, vol. 5, no. 1, pp. 15–17, 1976.
[9] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, 1998.
[10] S. Esmeir and S. Markovitch, "Lookahead-based algorithms for anytime induction of decision trees," in *Proceedings of the 21st International Conference on Machine Learning (ICML-04)*. ACM, 2004, p. 33.
[11] D. Page and S. Ray, "Skewing: An efficient alternative to lookahead for decision tree induction," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003, pp. 601–612.
[12] R. C. Barros, M. P. Basgalupp, A. C. De Carvalho, A. Freitas *et al.*, "A survey of evolutionary algorithms for decision-tree induction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 3, pp. 291–312, 2012.
[13] I. Ben-Gal, A. Dana, N. Shkolnik, and G. Singer, "Efficient construction of decision trees by the dual information distance method," *Quality Technology & Quantitative Management*, vol. 11, no. 1, pp. 133–147, 2014.
[14] Y. Wang, C. Song, and S.-T. Xia, "Unifying decision trees split criteria using tsallis entropy," *arXiv preprint arXiv:1511.08136*, 2015.
[15] C. Tsallis, "Possible generalization of boltzmann-gibbs statistics," *Journal of Statistical Physics*, vol. 52, no. 1-2, pp. 479–487, 1988.
[16] C. Tsallis, "Generalizing what we learnt: Nonextensive statistical mechanics," in *Introduction to Nonextensive Statistical Mechanics*. Springer, 2009, pp. 37–106.
[17] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
[18] S. Umarov, C. Tsallis, and S. Steinberg, "On a q-central limit theorem consistent with nonextensive statistical mechanics," *Milan journal of mathematics*, vol. 76, no. 1, pp. 307–328, 2008.
[19] C. Tsallis, *Introduction to nonextensive statistical mechanics*. Springer, 2009.
[20] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
[21] J. Lisman and M. Van Zuylen, "Note on the generation of most probable frequency distributions," *Statistica Neerlandica*, vol. 26, no. 1, pp. 19–23, 1972.
[22] S. Y. Park and A. K. Bera, "Maximum entropy autoregressive conditional heteroskedasticity model," *Journal of Econometrics*, vol. 150, no. 2, pp. 219–230, 2009.
[23] T. Maszczyk and W. Duch, "Comparison of shannon, renyi and tsallis entropy used in decision trees," in *Proceedings of the 17th International Conference on Artificial Intelligence and Soft Computing (ICAISC-08)*. Springer, 2008, pp. 643–651.
[24] A. Rényi, "On a new axiomatic theory of probability," *Acta Mathematica Hungarica*, vol. 6, no. 3-4, pp. 285–335, 1955.
[25] S. Abe and A. Rajagopal, "Nonadditive conditional entropy and its significance for local realism," *Physica A: Statistical Mechanics and its Applications*, vol. 289, no. 1, pp. 157–164, 2001.
[26] S. Furuichi, "Information theoretical properties of tsallis entropies," *Journal of Mathematical Physics*, vol. 47, no. 2, pp. 109–120, 2006.
[27] M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2013.
[28] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
[29] P. Perner, "Decision tree induction methods and their application to big data," in *Modeling and Processing for Next-Generation Big-Data Technologies*. Springer, 2015, pp. 57–88.