Knowledge-Based Systems 120 (2017) 34–42

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A less-greedy two-term Tsallis Entropy Information Metric approach for decision tree classification

Yisen Wang[a,b], Shu-Tao Xia[a,b], Jia Wu[c,*]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[b] Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[c] Faculty of Engineering and IT, University of Technology Sydney, Sydney 2007, Australia

ARTICLE INFO

Article history:
Received 19 January 2016
Revised 14 December 2016
Accepted 20 December 2016
Available online 21 December 2016

Keywords:
Decision trees
Attribute split criterion
Tree construction
Classification

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, are based on one-term that lack adaptability to different datasets. To address the above issues, we propose a less-greedy two-term 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-term Tsallis conditional entropy, which is better than conventional one-term split criteria. Secondly, the new tree construction is based on a two-stage approach that reduces the greediness and avoids local optimum to a certain extent. The TEIM algorithm takes advantages of the generalization ability of two-term 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 and is more robust to noise.

1. 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 [1–3]. 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. Decision trees are most commonly induced in the following top-down manner. A given dataset is partitioned into left and right subsets by a split criterion test on attributes. The highest scoring partition which reduces the average uncertainty mostly is selected. Then the dataset is partitioned accordingly into two child nodes, growing the tree by making the node be the parent of the two newly created child nodes. This procedure is applied recursively until some stopping conditions, e.g. maximum tree depth or minimum leaf size, are reached.

Generally speaking, split criterion and construction method of decision trees are two fundamental issues in the induction of decision trees. As for the split criterion, a series of papers have analyzed its importance [4,5]. They demonstrate that different split criteria have a substantial influence on the generalization error of the induced decision trees. Thus, a large number of decision trees induction algorithms have been proposed based on different split criteria. For example, the Iterative Dichotomiser 3 (ID3) algorithm is based on Shannon entropy [6]; the C4.5 algorithm is based on Gain Ratio [7]; while the Classification And Regression Tree (CART) algorithm is based on Gini index [8]. 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 lack adaptability to datasets. Numerous alternatives have been proposed for the adaptive entropy estimate [9,10], but their statistical entropy estimates are too complicated to lose the simplicity and comprehensibility of decision trees. Recently, a Tsallis entropy split criterion has been proposed in [11] to unify common split criteria, i.e. Shannon entropy, Gain Ratio and Gini index. Although it provides a new perspective to enhance the performance of decision trees, its split criterion is still one-term and tree complexity is also very large, similar to the above common split criteria.

* Corresponding author.
E-mail addresses: wangys14@mails.tsinghua.edu.cn (Y. Wang), xiast@sz.tsinghua.edu.cn (S.-T. Xia), jia.wu@uts.edu.au (J. Wu).

http://dx.doi.org/10.1016/j.knosys.2016.12.021

Meanwhile, the optimal construction of decision trees has been theoretically proven to be NP-complete [12,13]. Consequently, most practical implementations of decision trees use greedy algorithms to grow trees. Such approaches, however, suffer from the flaws of local optimums. Moreover, the greediness also renders decision trees sensitive to the noise of data. Several alternatives have been proposed to overcome the issue. The ID3 algorithm with lookahead technique is presented in [14], but its complexity increases exponentially, as the degree of lookahead grows. Another alternative method to lookahead is the skewing technique [15], 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 [16], but their disadvantages are also obvious such as time-consuming computation and a large number of parameters. Dual information metric [17] is another method to reduce greediness, but its classification accuracy is worse than C4.5.

To address the above two issues, we propose a less-greedy two-term Tsallis Entropy Information Metric (TEIM) algorithm with a new split criterion and a new method for constructing decision trees. Generally, Tsallis entropy is used as the split criteria of decision trees in a one-term formula [11,18–21]. The further discussion of the Tsallis entropy split criterion is not presented, and the construction of decision trees is still greedy. As opposed to the traditional use of Tsallis entropy, we design a new split criterion M_q based on two-term Tsallis entropy rather than one-term, i.e. the summation of two symmetrical Tsallis conditional entropies. We also propose a two-stage approach to make the construction of trees less greedy and the decision trees more robust to noise. As a result, the TEIM algorithm indeed renders smaller decision trees with better performance, which is beneficial for real-time classification support in health-care systems [19,22], attack detection [23], and online classification in Big Data environment [24]. Additionally, its robustness makes it more useful in real world classification problems where noise is unavoidable [25,26]. In summary, the main contributions of the paper are as follows:

- We propose a novel decision tree algorithm, called TEIM, which uses a newly designed two-term split criterion M_q and a two-stage tree construction approach.
- The new split criterion M_q is based on Tsallis entropy with two terms, making the decision tree obtain better adaptability to datasets.
- The two-stage based tree construction method takes the influence of previous attributes and class labels into account, which reduces the greediness in decision tree induction, making the decision tree robust to noise.
- Experimental results on real-world datasets demonstrate that TEIM achieves significant performance gain, and has the better adaptability to datasets and the stronger robustness to noise, compared to the state-of-the-art decision tree algorithms.

The rest of this paper is organized as follows. Section 2 presents the background of Tsallis entropy framework. Section 3 describes the proposed TEIM algorithm. Section 4 exhibits experimental results. Section 5 summarizes the work.

2. Tsallis entropy framework

2.1. Tsallis entropy

Tsallis entropy $S_q(X)$ is one kind of generalization of Shannon entropy adding one more adjustable parameter q [27], 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 $\{x_1, \ldots, x_n\}$ 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 [28].

With respect to Shannon entropy $H(X)$ proposed in [29], 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 $q \to 1$:

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

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 $p(x_i) = 1/n, i = 1, 2, \ldots, n$. 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 [30]. 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 [31]. To be specified, data across a variety of domains exhibit a property known as the heavy tail in reality [32]. However, we cannot characterize power-law heavy-tailed distribution through maximizing Shannon entropy subject to normal mean and variance [33,34]. The reason is that Shannon entropy implicitly assumes a certain trade-off between contributions from the tails and the main mass of distribution [35]. 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 [27,36]. A well-known generalization of this concept is Tsallis entropy.

More importantly, there is a crucial difference between Shannon entropy and Tsallis entropy, i.e. additivity. For two independent random variables X and Y, Shannon entropy has the additivity property:

$$H(X, Y) = H(X) + H(Y), \tag{6}$$

however, Tsallis entropy $S_q(X)$ $(q \neq 1)$ has the pseudo-additivity (also called q-additivity) property:

$$S_q(X, Y) = S_q(X) + S_q(Y) + (1-q)S_q(X)S_q(Y). \tag{7}$$

Besides, Tsallis conditional entropy, Tsallis joint entropy and Tsallis mutual information 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 [37] are denoted by:

$$S_q(X|Y) \quad = -\sum_{x,y} p(x, y)^q \ln_q p(x|y), \ (q \neq 1) \tag{8}$$

$$S_q(X,Y) = -\sum_{x,y} p(x,y)^q \ln_q p(x,y), \ (q \neq 1). \tag{9}$$

It is remarkable that (8) can be easily reformed by

$$S_q(X|Y) = \sum_y p(y)^q S_q(X|y). \tag{10}$$

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

$$S_q(X,Y) = S_q(X) + S_q(Y|X). \tag{11}$$

Tsallis mutual information [38] is defined as the difference between Tsallis entropy and Tsallis conditional entropy:

$$I_q(X;Y) = S_q(X) - S_q(X|Y). \tag{12}$$

Moreover, the chain rule of Tsallis mutual information for random variables $X_1, \dots, X_n$ and Y holds:

$$I_q(X_1,\dots,X_n;Y) = \sum_{i=1}^{n} I_q(X_i;Y|X_1,\dots,X_{i-1}). \tag{13}$$

The relation between the conditional entropy, joint entropy and mutual information can be derived from (11) and (12):

$$S_q(Y|X) + S_q(X|Y) = S_q(X,Y) - I_q(X;Y). \tag{14}$$

2.2. Tsallis entropy criterion

Tsallis entropy criterion unifies Shannon entropy, Gain Ratio and Gini index in a parametric framework [11]. The relations between Tsallis entropy and other split criteria are shown as follows.

Tsallis entropy converges to Shannon entropy for $q \to 1$ as shown in (3). Besides, Gini index is exactly a specific case of Tsallis entropy with $q = 2$:

$$\begin{aligned}\{S_q(X)\}_{q=2} &= \underbrace{\frac{1}{1-q}\left(\sum_{i=1}^{n} p(x_i)^q - 1\right)}_{q=2} \\ &= 1 - \sum_{i=1}^{n} p(x_i)^2 \\ &= \text{Gini index}.\end{aligned} \tag{15}$$

As for the Gain Ratio (GR) which adds a normalization factor compared with standard Information Gain based on Shannon entropy, it can be seen that if Shannon entropy is replaced by Tsallis entropy, Gain Ratio is generalized to Tsallis Gain Ratio (Tsallis GR). Similar to (3), Tsallis Gain Ratio also converges to Gain Ratio as $q \to 1$:

$$\begin{aligned}\lim_{q\to 1} \text{Tsallis GR} &= \lim_{q\to 1} \frac{S_q(D) - \frac{|D'|}{|D|}S_q(D') - \frac{|D''|}{|D|}S_q(D'')}{S_q\left(\frac{|D'|}{|D|}, \frac{|D''|}{|D|}\right)} \\ &= \frac{\overbrace{H(D) - \frac{|D'|}{|D|}H(D') - \frac{|D''|}{|D|}H(D'')}^{\text{Information Gain}}}{H\left(\frac{|D'|}{|D|}, \frac{|D''|}{|D|}\right)} \\ &= \text{Gain Ratio (GR)},\end{aligned} \tag{16}$$

where D' and D'' are two child subsets if D is split in a binary manner.

In summary, Tsallis entropy unifies three kinds of split criteria, e.g. Shannon entropy, Gain Ratio and Gini index, and generalizes the split criterion of decision trees. Besides, the parameter q enables the adaptability and flexibility of Tsallis entropy criterion. More importantly, Tsallis entropy provides a new approach to enhance the performance of decision trees through a tunable parameter q in a unified framework.

3. 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.

3.1. Problem statement

Given a dataset $\mathcal{D}_n$ with n instances, each instance (X, Y) has attributes A_j $(j \in \{1, 2, \dots, d\})$ and class label $Y \in \{1, 2, \dots, K\}$. The training and predicting procedure of decision trees, from a perspective of information theory, is the procedure of maximizing the mutual information between the selected attributes and the corresponding class label [9]. The mutual information $I_q(A_1, A_2, \dots, A_d; Y)$ between a list of attributes $A_j (j = 1, 2, \dots, d)$ and the corresponding class label Y can be formulated in the chain role of conditional mutual information. According to (13) and (12), we can obtain:

$$\begin{aligned}I_q(A_1,A_2,\dots,A_d;Y) &= \sum_{j=1}^{d} I_q\big(A_j;Y\big|A_{j-1},\dots,A_1\big) \\ &= \sum_{j=1}^{d} S_q\big(A_j\big|A_{j-1},\dots,A_1\big) \\ &\quad - \sum_{j=1}^{d} S_q\big(A_j\big|A_{j-1},\dots,A_1,Y\big).\end{aligned} \tag{17}$$

The above equation provides a new perspective to comprehend the attribute selection procedure in the construction of decision trees. The first term of (17), i.e. $S_q(A_j|A_{j-1},\dots,A_1)$, represents the Tsallis conditional entropy of the jth attribute given the former $j-1$ attributes. It can be viewed as a measure of the orthogonality among the attributes independently on the class label. The second term of (17), i.e. $S_q(A_j|A_{j-1},\dots,A_1,Y)$, represents the Tsallis conditional entropy of the jth attribute given the former $j-1$ attributes and the class label Y. It can be considered as a measure of the uncertainty in each attribute about the class label, given the preselected attributes.

Since Tsallis entropy is non-negative, in order to maximize the mutual information, one needs to maximize the first summation term and minimize the second summation term at the same time according to (17). Inspired by this, we get the idea of a two-stage approach in the TEIM algorithm. However, before that, we need to establish a metric to measure the relations between attributes and class labels.

3.2. The two-term information metric $\mathrm{M_q}$

One key issue in the procedure of decision tree induction is the split criterion. At every step, the decision tree chooses one pair of attribute and cutting point which makes the maximal impurity decrease to split the data and grow the tree. Therefore, the pair of attribute and cutting point chosen to split significantly affects the structure of decision trees and further influences the classification performance.

Compared with one-term formula of Tsallis entropy criterion in [11], we propose a new information metric M_q in a two-term 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:

$$M_q(A,Y) = S_q(Y|A) + S_q(A|Y), \tag{18}$$

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 can conclude that in order to obtain maximal impurity decrease one need to minimize the M_q between attributes

Table 1
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

and class labels. Besides, it is important to note that M_q follows the required mathematical properties of a metric [39], namely:

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

$$\begin{cases} M_q(E,F) = 0 \text{ iff } E = F \\ M_q(E,F) = M_q(F,E) \\ M_q(E,G) \leq M_q(E,F) + M_q(F,G) \end{cases} \tag{19}$$

M_q has a 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 the small illustrative example in Table 1. 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 simplify, we assume $q = 1$, then S_q converges to H. Consequently, 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 $H(A_2|Y) = 0$. In aiming to minimize M_q, one prefers attribute A_2 to A_1, and yet, for a binary split, attribute A_2 only needs one split to classify the dataset completely, while attribute A_1 requires multiple splits. In terms of tree complexity, attribute A_2 is decidedly better than A_1.

The popular algorithms for decision trees, such as ID3 or TEC [11], take into account the uncertainty $S_q(Y|A_i)$ in the class label Y following the selection of attribute A_i. That is to say, they only consider the first term $S_q(Y|A_i)$ in M_q. Note from (18) that our proposed metric M_q considers both $S_q(Y|A_i)$ and $S_q(A_i|Y)$. In the above example, ID3 randomly chooses the attribute A_1 or A_2 to split, while M_q chooses A_2. The example in Table 1 shows that M_q with its two-term formula performs better than the original one-term split criteria. This is because that M_q prefers to choose fewer, but more efficient attributes, that partition the dataset as closely as possible to the class while avoiding unnecessary splits.

3.3. The two-stage based tree construction

Although the optimal induction of decision trees is NP-complete, the efficient construction of near-optimal decision trees remains an open issue. Inspired by the two-term spirits of Tsallis mutual information in (17), we propose a two-stage approach for efficient construction of decision trees.

As stated in the Section 3.2, 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 pr**e**vious one step selected attribute is A_e and the c**u**rrent to be selected attribute is A_u, the object of the two-stage approach is to select the best attribute A_u which minimizes $L(A_u)$:

$$\begin{aligned} L(A_u) &= M_q(A_u, Y) - M_q(A_e, A_u) \\ &= S_q(A_u|Y) + S_q(Y|A_u) - S_q(A_e|A_u) - S_q(A_u|A_e). \end{aligned} \tag{20}$$

In order to minimize $L(A_u)$, we need to minimize the first term $M_q(A_u, Y)$ and maximize the second term $M_q(A_e, A_u)$, because 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 of the relevance between attributes and class labels, the more information on 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. Greater orthogonality among attributes means that the two-stage approach chooses fewer redundant but more 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.

To be specific, given a dataset $\mathcal{D}_n$ with n instances, each instance (X, Y) has attributes A_j $(j \in \{1, 2, \dots, d\})$ and class label $Y \in \{1, 2, \dots, K\}$. 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 in a binary split manner. For an attribute A_j, we obtain

$$\begin{aligned} L(A_j(C_j)) &= S_q(A_j(C_j)|Y) + S_q(Y|A_j(C_j)) \\ &\quad - S_q(A_e(C_e)|A_j(C_j)) - S_q(A_j(C_j)|A_e(C_e)), \end{aligned} \tag{21}$$

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_n, it can be used to predict for new unlabeled instances. The decision tree makes the prediction in a majority vote manner. For the unlabeled instance x, the probability of each class $k \in \{1, 2, \dots, K\}$ is

$$\eta^{(k)}(x) = \frac{1}{N(A_n(x))} \sum_{(X,Y) \in A_n(x)} \mathbb{I}(Y = k), \tag{22}$$

where $A_n(x)$ denotes the leaf containing x and $N(A_n(x))$ denotes the number of instances in $A_n(x)$. $\mathbb{I}(e)$ 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:

$$\hat{y} = g_n(x) = \arg\max_k \{\eta^{(k)}(x)\}. \tag{23}$$

3.4. TEIM algorithm

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

Taking the influence of previous attributes and class labels into account, the TEIM algorithm with a two-stage approach indeed reduces the greediness in the induction of decision trees. Besides, the TEIM algorithm adopts a better two-term criterion M_q than original one-term 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, robustness and performance.

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:             $D' \leftarrow \{X \in D | A_j(X) \leq C_j\}$
11:             $D'' \leftarrow \{X \in D | A_j(X) > C_j\}$
12:             // (X, Y) is one instance in the node D
13:             // D', D'' are the two child nodes
14:             Compute $L(A_j(C_j))$ according to (21)
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 Section 3.3
23: **end while**
24: **Return** A decision tree
25: // Tree is built by nodes from the root to the leaf

4. Experiments

This section is divided into three parts to evaluate the proposed TEIM algorithm. The first one is to present the influence of parameter q in M_q. The second one is to exhibit the performance enhancement of TEIM algorithm. The third one is to demonstrate the robustness of TEIM algorithm to noise.

4.1. Evaluation metric

In order to quantitatively compare the trees obtained by different methods, we employ Accuracy (*ACC*) and Area Under the ROC Curve (*AUC*) to evaluate the effectiveness of the tree. The ACC, calculated by the percentage of successful predictions, has been well used on domain specific problems, such as graph mining [40–42]. In addition to the accuracy, some data mining applications also require accurate rankings, so we also collect the AUC in our experiments.

$$ACC = \frac{1}{N(\mathcal{D}_t)} \sum_{(X,Y) \in \mathcal{D}_t} \mathbb{I}(Y = g_n(X)), \tag{24}$$

where $\mathcal{D}_t$ is the test data and $N(\mathcal{D}_t)$ is the number of instance in $\mathcal{D}_t$. $\mathbb{I}(e)$ is the indicator function that takes 1 if e is true and 0 for other cases. Besides, g_n is the decision tree classifier.

The extension of the standard two-class ROC for multi-class problems [43,44] is denoted by:

$$AUC = \frac{2}{K(K-1)} \sum_{\{k_i, k_j\}} AUC(k_i, k_j), \tag{25}$$

where K is the number of classes and $AUC(k_i, k_j)$ is the area under the two-class ROC curve involving the classes k_i and k_j.

As for the measure of the tree complexity, we employ the total number of the tree nodes (*Nodes*).

Table 2
Datasets from UCI.

Dataset	Instances	Attributes	Classes
Hayes	160	5	3
Wine	178	13	3
Glass	214	10	7
Haberman	306	3	2
Monks	432	7	2
Scale	625	4	3
Vehicle	946	18	4
Cmc	1,473	9	3
Yeast	1,484	8	10
Car	1,728	6	4
Image	2,310	19	7
Chess	3,196	36	2
EEG	14,980	15	2
Letter	20,000	16	26

4.2. Datasets

As shown in Table 2, the 14 UCI datasets [45] are used to evaluate the proposed algorithm. These datasets are ranked by the number of instances. Note that the number of instances, attributes and classes are varied, and are sufficiently representative to demonstrate the performance of TEIM.

4.3. The influence of parameter q in TEIM

As illustrated in Section 3.2, M_q is defined in a two-term formula, i.e. the summation of two Tsallis conditional entropies. The parameter q in Tsallis entropy can be tuned for datasets, which enables the adaptability and flexibility of M_q. In the following, we will see the influence of q in the classification accuracy (*ACC*), area under the ROC curve (*AUC*) and tree complexity (*Nodes*).

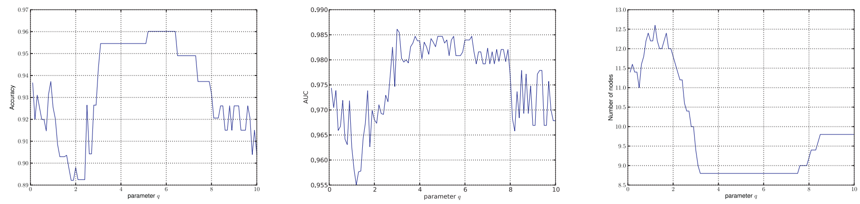
The TEIM algorithm of decision trees is implemented in Python. To exhibit the influence of q roundly, we traverse the parameter q in a step of 0.1 in the range [0.1, 10.0]. For each selected q, we perform a 10 times 10-fold cross-validation to evaluate the performance. Besides, the minimum leaf size is set to 5 to avoid overfitting.

Figs. 1–4 give intuitive exhibitions of the influence of different values of q in M_q on Wine (178 Instances, 13 Attributes), Yeast (1484 Instances, 8 Attributes), Car (1728 Instances, 6 Attributes) and EEG (14980 Instances, 15 Attributes) datasets, respectively. We can see that *ACC, AUC* and *Nodes* are all sensitive to the change of q. Most importantly, our proposed TEIM decision tree can obtain high *ACC*, high *AUC* and low *Nodes* at the same time (*e.g.* q in [5.1, 6.2] for Wine dataset, [0.5, 2.1] for Yeast dataset, [1.0, 2.0] for Car dataset, and [5.5, 6.1] for EEG dataset). In summary, experimental results show that the parameter q indeed has an effect on the classification accuracy, area under the ROC curve and tree complexity. Moreover, we can achieve different goals through selecting different q, e.g. highest accuracy or lowest complexity or trade-off, which also reflects the adaptability and flexibility of the TEIM algorithm.

4.4. The TEIM performance analysis

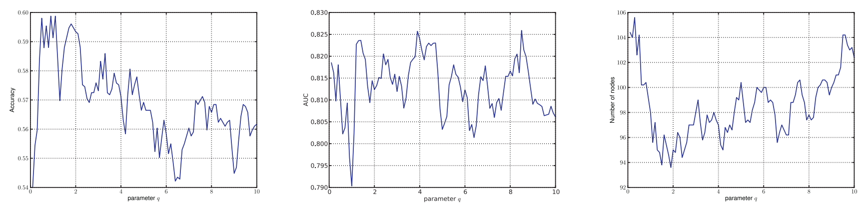
The TEIM algorithm combines advantages of the two-term Tsallis information metric M_q and two-stage tree construction, which can enhance the performance and reduce the greediness in the construction of decision trees. Thus, we conduct a series of experiments on datasets in Table 2 to test the performance of the proposed TEIM algorithm.

With respect to the algorithms for comparison, we choose two categories of decision tree algorithms based on one-term and two-term split criterion respectively. The one-term algorithms include



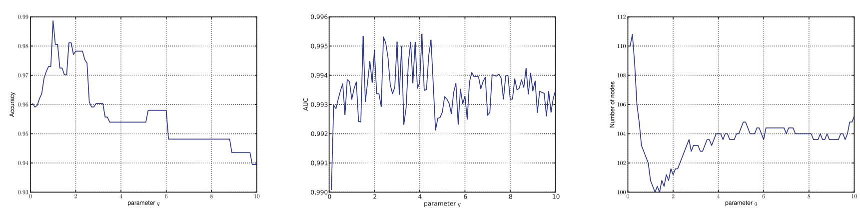
(a) Classification Accuracy (b) Area under the ROC curve (c) Tree Complexity

Fig. 1. Influence of parameter *q* in *ACC*, *AUC* and *Nodes* for the Wine dataset.



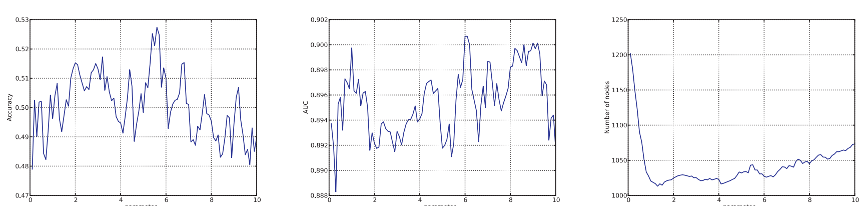
(a) Classification Accuracy (b) Area under the ROC curve (c) Tree Complexity

Fig. 2. Influence of parameter *q* in *ACC*, *AUC* and *Nodes* for the Yeast dataset.



(a) Classification Accuracy (b) Area under the ROC curve (c) Tree Complexity

Fig. 3. Influence of parameter *q* in *ACC*, *AUC* and *Nodes* for the Car dataset.



(a) Classification Accuracy (b) Area under the ROC curve (c) Tree Complexity

Fig. 4. Influence of parameter *q* in *ACC*, *AUC* and *Nodes* for the EEG dataset.

the state-of-the-art decision trees with Tsallis entropy (TE) and Tsallis Gain Ratio (TGR) criteria in [11] which have been shown to achieve better performance than ID3, C4.5 and CART. For two-term algorithms, we replace Tsallis entropy in TEIM with Shannon entropy (SEIM) and Renyi entropy (REIM) as baselines.

All the above decision trees algorithms are implemented in Python. In each dataset, we do a grid search using 10-fold cross-validation to determine the values of q and the value of α parameterizing Renyi entropy in REIM. 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.

Tables 3–6 report the comparisons of TEIM against other decision tree algorithms on various datasets. Firstly, in *ACC* as well as *AUC*, as shown in Tables 3 and 4, on each dataset, the best performance is in boldface, and the statistical significance analysis is also conducted, marked by “a”. We can see that TEIM outperforms TE, TGR, SEIM and REIM almost on all datasets. It is worth men-

Table 3
Classification accuracy (*ACC*%) of different decision tree algorithms on different datasets.

Dataset	TE	TGR	SEIM	REIM	TEIM	(Optimal q)
Hayes	82.3 [a]	81.5 [a]	**82.5**	**82.5**	**82.5**	8.6
Wine	93.1 [a]	92.9 [a]	93.0 [a]	94.5 [a]	**96.0**	5.1
Glass	60.6 [a]	53.1 [a]	55.1 [a]	60.7 [a]	**62.9**	2.6
Harberman	74.2 [a]	74.8	74.5 [a]	74.7 [a]	**75.2**	7.1
Monks	57.3 [a]	54.9 [a]	57.0 [a]	58.1 [a]	**60.9**	8.9
Scale	78.2 [a]	78.5 [a]	78.8 [a]	79.6 [a]	**82.2**	3.1
Vehicle	73.8 [a]	73.4 [a]	74.2	**74.8**	74.6	0.6
Cmc	52.0 [a]	47.8 [a]	50.3 [a]	53.4 [a]	**54.6**	1.2
Yeast	56.9 [a]	51.2 [a]	54.3 [a]	58.8 [a]	**59.8**	1.4
Car	98.3 [a]	98.4	98.3 [a]	98.7	**98.8**	0.8
Image	95.6 [a]	95.1 [a]	95.5 [a]	96.0 [a]	**96.6**	0.3
Chess	92.8 [a]	92.4 [a]	93.0 [a]	93.3 [a]	**93.8**	6.0
EEG	50.7 [a]	49.2 [a]	50.4 [a]	51.6 [a]	**52.8**	5.6
Letter	86.1 [a]	85.5 [a]	86.3 [a]	86.8 [a]	**87.3**	0.3

[a] TEIM is significantly better at the 0.05 significance level.

Table 4
Area Under the ROC Curve (*AUC*) of different decision tree algorithms on different datasets.

Dataset	TE	TGR	SEIM	REIM	TEIM
Hayes	0.965	0.965	0.965	0.966	**0.967**
Wine	0.968 [a]	0.962 [a]	0.965 [a]	0.973 [a]	**0.985**
Glass	0.813 [a]	0.808 [a]	0.827 [a]	0.835 [a]	**0.854**
Harberman	0.640 [a]	0.646 [a]	0.653 [a]	0.661 [a]	**0.677**
Monks	0.598 [a]	0.597 [a]	0.610 [a]	0.620 [a]	**0.641**
Scale	0.820 [a]	0.812 [a]	0.826 [a]	0.840 [a]	**0.865**
Vehicle	0.863 [a]	0.855 [a]	0.857 [a]	0.871 [a]	**0.877**
Cmc	0.679 [a]	0.671 [a]	0.683 [a]	0.690 [a]	**0.701**
Yeast	0.800 [a]	0.753 [a]	0.790 [a]	0.812 [a]	**0.823**
Car	0.992	0.992	0.993	0.994	**0.995**
Image	0.968 [a]	0.966 [a]	0.968 [a]	0.974 [a]	**0.979**
Chess	0.996	0.994	**0.997**	**0.997**	**0.997**
EEG	0.891	0.888 [a]	**0.899**	0.895	0.896
Letter	0.948	0.945 [a]	0.947	0.949	**0.951**

[a] TEIM is significantly better at the 0.05 significance level.

Table 5
Tree complexity (*Nodes*) of different decision tree algorithms on different datasets.

Dataset	TE	TGR	SEIM	REIM	TEIM
Hayes	19.5 [a]	**19.2** [a]	21.2	23.8	23
Wine	9.6	9.2	10.0 [b]	9.4	**8.8**
Glass	52.6 [b]	51.5 [b]	44.2 [b]	44.8 [b]	**27.6**
Harberman	33.2	33.0	33.0	34.8	36.4
Monks	89.6 [b]	88.0 [b]	87.2 [b]	86.7 [b]	**84.8**
Scale	104.6 [b]	99.6 [b]	98.5 [b]	95.7 [b]	**92.7**
Vehicle	111.0 [b]	135.7 [b]	107.6 [b]	103.0 [b]	**80.2**
Cmc	264.2 [b]	242.1 [b]	227.8 [b]	162.7 [b]	**72.8**
Yeast	195.8 [b]	197.1 [b]	156.2 [b]	139.5 [b]	**94.0**
Car	106.2 [b]	106.6 [b]	103.6 [b]	102.1 [b]	**100.0**
Image	90.0 [b]	84.4	84.8	84.0	**83.3**
Chess	59.8	56.0	**54.2** [a]	58.6	57.7
EEG	1200.0 [b]	1072.9 [b]	1026.0 [b]	1041.7 [b]	**1018.8**
Letter	3033.3 [b]	2895.8 [b]	2958.3 [b]	2866.7 [b]	**2812.5**

[a] TEIM constructs significantly smaller tree at the 0.05 significance level.
[b] TEIM constructs significantly bigger tree at the 0.05 significance level.

Table 6
Runing time (ms) of different decision tree algorithms on different datasets.

Dataset	TE	TGR	SEIM	REIM	TEIM
Hayes	3.98	4.00	4.51	4.60	4.60
Wine	13.48	13.61	14.35	14.46	14.20
Glass	12.87	12.85	12.70	13.00	12.87
Harberman	6.19	6.52	7.34	8.05	9.12
Monks	6.14	6.20	7.50	7.23	6.84
Scale	7.28	7.51	8.62	8.37	8.03
Vehicle	237.13	290.31	261.35	255.72	246.53
Cmc	30.23	28.14	100.65	80.76	53.53
Yeast	87.74	89.32	139.99	91.50	89.62
Car	12.94	14.05	25.25	24.88	24.37
Image	496.27	467.39	641.67	649.78	644.37
Chess	49.96	46.85	72.45	78.33	78.27
EEG	2043.52	1827.08	2620.81	2660.92	2602.42
Letter	4535.23	4329.65	5750.02	5571.98	5466.63

tioning that, for one-term based algorithms, the superiorities of TE and TGR have already been demonstrated compared with the classical algorithms ID3, C4.5 and CART [11], as does our proposed TEIM algorithm. Meanwhile, for two-term based algorithms, TEIM significantly outperforms SEIM because of its tunable parameter q. Moreover, TEIM also achieves significantly better performance than REIM. To summarize the reasons for this improvement, TEIM uses M_q as a split criterion that is based on the summation of two Tsallis conditional entropies, while the TE or TGR algorithms use one-term Tsallis entropy directly. As previously discussed, the two-term split criterion M_q is better than one-term Tsallis entropy split criterion. Although SEIM and REIM are also based on two-term split criterion, Tsallis entropy has better properties than Shannon and Renyi entropies - Tsallis entropy unifies common split criterion in a parametric framework as discussed in Section 2.2 while Renyi entropy does not [11,46]). Further, the two-stage approach is a maximal-orthogonality-maximal-relevance method that considers the previously selected attributes and class labels simultaneously, and this is also responsible for the improvement in performance. TEIM also benefits from Tsallis entropy and its adjustable parameter q. All these advantages combined result in the TEIM algorithm performing significantly better than the comparisons.

Focusing on tree complexity then, as reported in Table 5, the smallest tree complexity is in bold, and the statistical significance analysis is denoted by "a" or "b". As expected, TEIM results in a smaller tree on almost all datasets when compared to TE, TGR, SEIM and REIM. The remarkable decline in tree complexity is indicated on the Cmc dataset, whose tree nodes decrease from 264.2 to 72.8. A similar result also occurs on the Yeast dataset whose tree nodes decrease from 197.1 to 94.0. This decrease in complexity is caused by the two-term 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). Note that the resulting trees constructed by TEIM are not as complex as in REIM, demonstrating that Tsallis entropy can provide better generalization performance than Renyi entropy to a certain extent, as reflected in Tables 3 and 4.

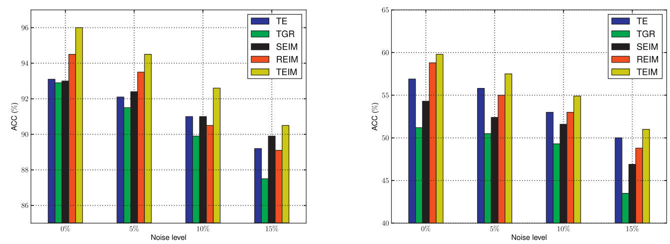
Another aspect of TEIM is its algorithm complexity. In general, the running time to construct a balanced binary tree is $O(nd\log(n))$ and query time is $O(\log(n))$. Assuming that the subtrees remain approximately balanced, the cost at each node consists of searching through $O(d)$ to find the attribute that offers the largest uncertainty reduction. Through a presorting technique, the cost at each node is $O(d\log(n))$, so that the total cost over the entire tree (by summing the cost at each node) is $O(nd\log(n))$. Although TEIM uses two-term and two-stage method, it only adds the constant algorithm complexity, which is still $O(nd\log(n))$. Using a MacBook Pro (2.8 GHz Intel Core i5, 16GB 1600 MHz DDR3), we test the running time of different decision tree algorithms and show the results in Table 6. We can see that TEIM does not consume much more time than other four algorithms (i.e., TE, TGR, SEIM and REIM). This is because TEIM requires more computation for each node, but less nodes to construct a tree. In addition, among the two-term split criterion algorithms (i.e., SEIM, REIM and TEIM), TEIM sometimes takes less time, again because it requires less nodes to construct decision trees.

Regarding the optimal value of q, though it is obtained by cross-validation in this paper, we find a fuzzy trend from Table 3. The more of class number, the smaller value of q. 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.

In summary, the TEIM algorithm utilizes the two-term based Tsallis entropy split criterion and the two-stage based tree construction to enhance the performance as well as reduce the greediness. So the TEIM algorithm can select fewer but efficient attributes for the tree construction. In other words, the proposed TEIM algorithm can construct smaller trees while maintaining higher classification performance.

4.5. The robustness analysis with respect to noise

As noted above, the two-stage approach in TEIM algorithm reduces the greediness in the construction of decision trees, which makes the TEIM algorithm avoid the local optimum to a certain



(a) Wine (178 Instances, 13 Attributes)

(b) Yeast (1484 Instances, 8 Attributes)

Fig. 5. The robustness of TEIM compared with TE and TGR.

extent. That is to say, the two-stage approach in TEIM algorithm is not only beneficial for the promotion of performance but also for the robustness to noise. In this part, we evaluate the TEIM algorithm with respect to the robustness of classification when attributes are noised.

The decision trees share the same parameters in Section 4.4. The only difference lies in the datasets which are noised manually. In order to corrupt each attribute A_j with a noise level of $l\%$, the $l\%$ of the instances in the dataset are chosen approximately and every value of A_j in these instances is assigned a random value between the minimum and maximum values of that attribute, following a uniform distribution. The noised datasets are supported by KEEL [47,48] and we choose the scheme of noisy training-clean test. A 10 times 10-fold cross-validation is also carried out to evaluate the performance.

The classification accuracy of different algorithms at different noise level is illustrated in Fig. 5. It is natural that the classification accuracy decreases as the noise level increases. At the same noise level, TEIM still achieves the highest accuracy. Moreover, at the same accuracy, e.g. around 93.0% on Wine dataset, TE, TGR and SEIM need almost no noise environment; REIM can bear about 5% level of noise; while TEIM can bear almost 10% level of noise. The results show that, compared with TE, TGR, SEIM and REIM algorithms, TEIM algorithm is more robust to noise. It is an appealing characteristic of the proposed TEIM algorithm in the real world where noise is common in datasets.

5. Conclusions

In this paper, we address two fundamental issues of decision trees, i.e. the split criterion and tree construction. We define a new two-term based 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 inspired by Tsallis mutual information. Combining all the strengths of Tsallis entropy, M_q and two-stage method together, a less-greedy two-term based 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, area under the ROC curve and tree complexity. Besides, the TEIM algorithm has the better adaptability to datasets and stronger robustness to noise, compared with the state-of-the-art decision trees algorithms.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under grant No. 61371078, and the R&D Program of Shenzhen under grant Nos. JCYJ20140509172959977, JSGG20150512162853495, ZDSYS20140509172959989, JCYJ20160331184440545.

References

[1] A. Gutierrez-Rodriguez, J.F. Martinez-Trinidad, M. Garcia-Borroto, J. Carrasco-Ochoa, Mining patterns for clustering on numerical datasets using unsupervised decision trees, Knowl. Based Syst. 82 (2015) 70–79.
[2] X. Li, H. Zhao, W. Zhu, A cost sensitive decision tree algorithm with two adaptive mechanisms, Knowl. Based Syst. 88 (2015) 24–33.
[3] I. Ibarguren, J.M. Perez, J. Muguerza, I. Gurrutxaga, O. Arbelaitz, Coverage-based resampling: building robust consolidated decision trees, Knowl. Based Syst. 79 (2015) 51–67.
[4] W. Buntine, T. Niblett, A further comparison of splitting rules for decision-tree induction, Mach. Learn. 8 (1) (1992) 75–85.
[5] W.Z. Liu, A.P. White, The importance of attribute selection measures in decision tree induction, Mach. Learn. 15 (1) (1994) 25–41.
[6] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[7] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
[8] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC press, 1984.
[9] 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.
[10] M. Serrurier, 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.
[11] Y. Wang, C. Song, S.-T. Xia, Unifying decision trees split criteria using tsallis entropy, arXiv preprint arXiv:1511.08136 (2015).
[12] L. Hyafil, R.L. Rivest, Constructing optimal binary decision trees is np-complete, Inf. Process. Lett. 5 (1) (1976) 15–17.
[13] S.K. Murthy, Automatic construction of decision trees from data: a multi-disciplinary survey, Data Min. Knowl. Discov. 2 (4) (1998) 345–389.
[14] S. Esmeir, 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.
[15] D. Page, 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.
[16] R.C. Barros, M.P. Basgalupp, A.C. De Carvalho, A. Freitas, et al., A survey of evolutionary algorithms for decision-tree induction, Syst. Man Cybern. Part C IEEE Trans. 42 (3) (2012) 291–312.
[17] I. Ben-Gal, A. Dana, N. Shkolnik, G. Singer, Efficient construction of decision trees by the dual information distance method, Qual. Technol. Quant. Manag. 11 (1) (2014) 133–147.
[18] K. Gajowniczek, T. Zabkowski, A. Orłowski, Comparison of decision trees with rényi and tsallis entropy applied for imbalanced churn dataset, in: Computer Science and Information Systems (FedCSIS-15), 2015 Federated Conference on, IEEE, 2015, pp. 39–44.
[19] A.H. Al-nuaimi, E. Jammeh, L. Sun, E. Ifeachor, Tsallis entropy as a biomarker for detection of alzheimer's disease, in: Engineering in Medicine and Biology Society (EMBC-15), 2015 37th Annual International Conference of the IEEE, IEEE, 2015, pp. 4166–4169.
[20] A. Bhandari, A. Kumar, G. Singh, Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms, Expert Syst. Appl. 42 (22) (2015) 8707–8730.
[21] V.P. Singh, H. Cui, Modeling sediment concentration in debris flow by tsallis entropy, Physica A 420 (2015) 49–58.
[22] S. Zheng, W. Liu, An experimental comparison of gene selection by lasso and dantzig selector for cancer classification, Comput. Biol. Med. 41 (11) (2011) 1033–1040.
[23] I. Basicevic, S. Ocovaj, M. Popovic, Use of tsallis entropy in detection of syn flood dos attacks, Secur. Commun. Netw. 8 (18) (2015) 3634–3640.
[24] 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.
[25] S. García, J. Luengo, F. Herrera, Dealing with noisy data, in: Data Preprocessing in Data Mining, Springer, 2015, pp. 107–145.
[26] S. He, H. Chen, Z. Zhu, D.G. Ward, H.J. Cooper, M.R. Viant, J.K. Heath, X. Yao, Robust twin boosting for feature selection from high-dimensional omics data with label noise, Inf. Sci. 291 (2015) 1–18.
[27] C. Tsallis, Possible generalization of boltzmann-gibbs statistics, J. Stat. Phys. 52 (1–2) (1988) 479–487.
[28] C. Tsallis, Generalizing what we learnt: nonextensive statistical mechanics, in: Introduction to Nonextensive Statistical Mechanics, Springer, 2009, pp. 37–106.
[29] C. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.
[30] S. Umarov, C. Tsallis, S. Steinberg, On a q-central limit theorem consistent with nonextensive statistical mechanics, Milan J. Math. 76 (1) (2008) 307–328.
[31] C. Tsallis, Introduction to Nonextensive Statistical Mechanics, Springer, 2009.
[32] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[33] J. Lisman, M. Van Zuylen, Note on the generation of most probable frequency distributions, Stat. Neerl. 26 (1) (1972) 19–23.
[34] S.Y. Park, A.K. Bera, Maximum entropy autoregressive conditional heteroskedasticity model, J. Econom. 150 (2) (2009) 219–230.

[35] T. Maszczyk, 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.
[36] A. Rényi, On a new axiomatic theory of probability, Acta Math. Hungarica 6 (3–4) (1955) 285–335.
[37] S. Abe, A. Rajagopal, Nonadditive conditional entropy and its significance for local realism, Physica A 289 (1) (2001) 157–164.
[38] T. Yamano, Information theory based on nonadditive information content, Phys. Rev. E 63 (4) (2001) 046105.
[39] S. Furuichi, Information theoretical properties of tsallis entropies, J. Math. Phys. 47 (2) (2006) 023302.
[40] J. Wu, S. Pan, X. Zhu, C. Zhang, X. Wu, Positive and unlabeled multi-graph learning, IEEE Trans. Cybern. PP (99) (2016) 1–12, doi:10.1109/TCYB.2016.2527239.
[41] J. Wu, S. Pan, X. Zhu, Z. Cai, Boosting for multi-graph classification, IEEE Trans. Cybern. 45 (3) (2015) 416–429.
[42] J. Wu, X. Zhu, C. Zhang, P.S. Yu, Bag constrained structure pattern mining for multi-graph classification, IEEE Trans. Knowl. Data Eng. 26 (10) (2014) 2382–2396.
[43] D.J. Hand, R.J. Till, A simple generalisation of the area under the roc curve for multiple class classification problems, Mach. Learn. 45 (2) (2001) 171–186.
[44] J. Wu, S. Pan, X. Zhu, P. Zhang, C. Zhang, Sode: self-adaptive one-dependence estimators for classification, Pattern Recognit. 51 (2016) 358–377.
[45] M. Lichman, UCI machine learning repository, 2013, (http://archive.ics.uci.edu/ml).
[46] A. Pattanaik, S. Mishra, D. Rana, Comparative study of edge detection using renyi entropy and differential evolution, International Journal of Engineering Research and Technology, vol. 4, ESRSA Publications, 2015.
[47] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, et al., Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[48] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2–3) (2010) 255–287.