


# Levels of Confidence and Utility for Binary Classifiers

Zhiyi Zhang 

The University of North Carolina at Charlotte, Charlotte, NC 28223, USA; z Zhang@uncc.edu

**Abstract:** Two performance measures for binary tree classifiers are introduced: the level of confidence and the level of utility. Both measures are probabilities of desirable events in the construction process of a classifier and hence are easily and intuitively interpretable. The statistical estimation of these measures is discussed. The usual maximum likelihood estimators are shown to have upward biases, and an entropy-based bias-reducing methodology is proposed. Along the way, the basic question of appropriate sample sizes at tree nodes is considered.

**Keywords:** binary classifiers; tree classifiers; level of confidence; level of utility; entropies; the binomial distributions; the entropic binomial distributions; the maximum likelihood estimators; the entropic maximum likelihood estimators

## 1. Introduction

The main objective of this article is to propose two performance measures for binary classifiers, the level of confidence and the level of utility. There is no lack of performance measures for tree classifiers in the literature of data science. However, the two proposed measures are probabilities of two desirable events associated with a binary classifier and, as such, imply simple and clear meanings. Furthermore, the proposed measures provide statistical support for considerations in the process of developing a binary classifier in the sense of classic probability and statistics.

Decision trees are a tool of central importance in modern data science, of which the binary decision trees are an emblematic case. The discussion in this article has relevance to multinomial decision trees. However, for simplicity and clarity, the primary focus of the discussion below is on binary classifiers at nodes in a tree structure. The construction of a decision tree may be approached with different cultures and logics, each with pros and cons. Interested readers may refer to [1,2] for in-depth discussions on different cultures of such undertakings. The two main different cultures are often termed data science and statistics, respectively, albeit not without overlapping domains. Regardless of the variation in the logic of the construction effort, the core task remains the same. Consider a Bernoulli random variable,  $B = B(p_x)$ , where  $x \in \mathcal{X}^*$  and  $\mathcal{X}^*$  is a sample space for a random co-variate element  $X$ . One of the simplest binary tree classifiers may be developed according to the following model.

1. An identically and independently distributed (*iid*) sample of size  $n$  is taken,  $\{(B_i, X_i); 1 \leq i \leq n\}$ , where  $X_i$  is a random element on  $\mathcal{X}^*$  according to some distribution and  $B_i$  is, conditioning on  $X = x$ , a Bernoulli random variable with  $p_x$ . To construct a binary tree classifier is to find, based on the sample, a partition of  $\mathcal{X}^*$ , denoted  $\mathcal{X} = \{x_j; 1 \leq j \leq J\}$ , such that in each sub-group indexed by  $x_j$ ,  $B_{x_j}$ , or more simply  $B_j$ , is a Bernoulli random variable, conditioning on  $X = x_j$  with  $p_{x_j} = p_j$  and  $q_j = 1 - p_j$ . Let  $N_j = \sum_{i=1}^n 1_{[X_i=x_j]}$ .  $\{N_j; 1 \leq j \leq J\}$  is a multinomial vector of size  $n$  with its realization  $\{n_j; 1 \leq j \leq J\}$ . The first sample of size  $n$ ,  $\{(B_i, X_i); 1 \leq i \leq n\}$ , may be thought of as a pair  $(B_i, X_i)$ , where  $X_i$  is a random element on  $\mathcal{X} = \{x_j; 1 \leq j \leq J\}$  with probability distribution  $\lambda = \{\lambda_j; 1 \leq j \leq J\}$ , and  $B_i$  is conditionally Bernoulli with  $p_j$  given  $X_i = x_j$ . Let  $Y_j = \sum_{i=1}^n B_i 1_{[X_i=x_j]}$  and



**Citation:** Zhang, Z. Levels of Confidence and Utility for Binary Classifiers. *Stats* **2024**, *7*, 1209–1225. <https://doi.org/10.3390/stats7040071>

Academic Editor: Wei Zhu

Received: 20 September 2024

Revised: 7 October 2024

Accepted: 11 October 2024

Published: 17 October 2024



**Copyright:** © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

- $\hat{p}_j = Y_j/N_j$  be the frequency and the relative frequency of the sample of size  $n$  in the  $j$ th sub-group. Another *iid* element, denoted  $(B_{n+1}, X_{n+1})$ , is to be taken.
2. A tree classifier is defined as follows: given  $X_{n+1} = x \in \mathcal{X}$ , (a) if  $\hat{p}_x > 0.5$ ,  $B_{n+1}$  is projected to be 1 (a success); (b) if  $\hat{p}_x < 0.5$ ,  $B_{n+1}$  is projected to be 0 (a failure); or (c) if  $\hat{p}_x = 1 - \hat{p}_x = 0.5$ , a fair coin is tossed to determine the classification of  $B_{n+1}$ .

There is a long list of issues involved with constructing a classifier as described above, some of which are fundamental and some are technical. To see a comprehensive discussion, one may refer to, for example, [3]. The volume of methodologies for developing classifiers has increased rapidly in recent decades, but mostly in the realm of data science rather than statistics. There are good reasons why much of the development of classifiers is on the side of data science. One of the most distinctive characteristics of data science, as opposed to statistics, is the highly non-parametric nature of the associated methodologies. Unlike many traditional statistical models, which usually have a low-dimensional data space, data science models are more general, more flexible and more complex. As such, they have a tendency to be over-zealous in dynamically searching and establishing features based on the sample in the data space. This phenomenon is sometimes known as “heat seeking” to data scientists, which may be thought of as over-fitting in the usual statistical terminologies. On top of the said “heating seeking”, there exists a fact that is exacerbating the situation: many important quantities of interest in developing and evaluating a classifier depend on the parameter  $p_V = \max\{p, 1 - p\}$ , and the usual and natural estimator  $\hat{p}_V = \max\{\hat{p}, 1 - \hat{p}\}$  of  $p_V$  has an upward bias. This fact may be plainly seen in a very simple setting.

1. Let the binary alphabet be denoted  $\mathcal{L} = \{\ell_1, \ell_2\}$  and associated with a probability distribution  $P(\ell_1) = p$  and  $P(\ell_2) = 1 - p$ .
2. Let  $p_V = \max\{p, 1 - p\}$  and  $p_\wedge = \min\{p, 1 - p\}$ , and assume  $p_V > p_\wedge$ .
3. Let the letter, corresponding to probability  $p_V$ , be denoted  $\ell_V$ , that is,  $\ell_V = \arg \max_{\ell \in \mathcal{L}} P(\ell)$ . Letter  $\ell_V$  is also referred to as the true letter.

Any reasonable performance measure of a simple classifier based on an *iid* Bernoulli sample is basically a function of  $p_V$  (and  $p_\wedge$ ). Therefore, the quality of an estimator of  $p_V$  becomes essential to the quality of the estimator of such a performance measure, which could in turn guide the entire process of constructing a binary classifier. However, a good estimator of  $p$  or  $q = 1 - p$  does not necessarily imply a good estimator of  $p_V = \max\{p, 1 - p\}$ . For example, the relative frequencies  $\hat{p}$  and  $\hat{q} = 1 - \hat{p}$  based on an *iid* Bernoulli sample are uniform minimum variance unbiased estimators of  $p$  and  $q = 1 - p$ ; but  $\hat{p}_V = \max\{\hat{p}, 1 - \hat{p}\}$  is upwardly biased since the function  $f(p) = \max\{p, 1 - p\}$  is a convex function, and hence, by Jensen’s inequality,  $E(\max\{\hat{p}, 1 - \hat{p}\}) \geq \max\{E(\hat{p}), E(1 - \hat{p})\} = p_V$ . In fact, the bias could be quite significant when the sample size  $n$  is small. This simple observation has profound implications in the process of constructing a binary classifier, or more generally a binary tree classifier as every node of a tree resembles a simple binary classifier. The upward bias tends to overestimate  $p_V$ , hence exaggerating the confidence in selecting  $\hat{\ell}_V = \arg \max_{\ell \in \mathcal{L}} \hat{P}(\ell)$  as the likely true letter.

In this article, several relevant results are presented in the subsections of Section 2. These results may be thought to belong to two categories. The first contains motivational arguments leading to the definitions of the level of confidence and the level of utility of a binary classifier. Along the way, a general entropy and a notion of entropic objects, including an entropic binomial distribution, are defined. The second contains some consideration of the estimation of the levels of confidence and utility, which includes an introduction to the notion of an entropic maximum likelihood estimator (*emle*), as opposed to the maximum likelihood estimator (*mle*). The article then proposes a weighted average of the *mle* and the *emle* of  $p_V$  as a bias-alleviating estimator of  $p_V$ . Several numerical calculations and simulation studies are also reported in the same section. Finally, the article ends with a few concluding remarks in the last section, including several recommendations to practitioners on how to incorporate the findings of this article into practice.

## 2. Main Results

### 2.1. Entropies and Entropic Objects

Consider a general countable alphabet,  $\mathcal{L} = \{\ell_k; k \geq 1\}$ , along with an associated probability distribution,  $\mathbf{p} = \{p_k; k \geq 1\}$ . Let  $\mathbf{p}_\downarrow = \{p_{(k)}; k \geq 1\}$  be the non-increasingly re-arranged  $\mathbf{p}$ , that is, for every  $k, k \geq 1, p_{(k)} \geq p_{(k+1)}$ . A general notion of entropy was first given in [4], but is given below for a self-contained presentation.

**Definition 1.** A function  $f(\mathbf{p})$  is referred to as an entropy if  $f(\mathbf{p})$  depends on  $\mathbf{p}$  only through  $\mathbf{p}_\downarrow$ , that is,  $f(\mathbf{p}) = f(\mathbf{p}_\downarrow)$ .

Definition 1 not only defines general entropies but also implies a notion of label-independence. An entropy is a measure that is invariant with regard to the labels of the underlying alphabet. Many well-known entropies studied in the existing literature include Shannon’s entropy  $H_s = -\sum_{k \geq 1} p_k \ln p_k$  as in [5], Rényi’s entropy  $H_r = \ln(\sum_{k \geq 1} p_k^\alpha) / (1 - \alpha)$  for some  $\alpha$  where  $0 < \alpha < \infty$  and  $\alpha \neq 1$  as in [6], the Tsallis entropy  $H_t = (1 - \sum_{k \geq 1} p_k^\alpha) / (\alpha - 1)$  for any  $\alpha > 1$  as in [7], and the generalized Simpson’s entropy  $H_{gs} = \sum_{k \geq 1} p_k^u (1 - p_k)^v$  for any pair of integers  $u \geq 1$  and  $v \geq 0$ , as in [8,9]. It may be interesting to note that  $p_{(k)}$  for any  $k$  is an entropy, and in particular,  $p_\vee = p_{(1)}$  is an entropy.

In the spirit of Definition 1, the adjective “entropic” is adopted to describe objects that are label-independent. For example, a sample of size  $n$  from a countable alphabet may be summarized by a multinomial random array  $\mathbf{Y} = \{Y_k; k \geq 1\}$ , which may be re-arranged non-increasingly into  $\mathbf{Y}_\downarrow = \{Y_{(k)}; k \geq 1\}$  and referred to as the entropic statistics associated with  $\mathbf{Y}$ . Similarly, while the elements of  $\mathbf{p} = \{p_k; k \geq 1\}$  are multinomial parameters, these of  $\mathbf{p}_\downarrow$  may be referred to as entropic multinomial parameters. It may be interesting to note that, by the same token, a classifier, or a decision tree, is also entropic in nature and that an exercise of developing a classifier is also entropic.

### 2.2. Entropic Binomial Distributions

Consider a Bernoulli population with probability  $p$  and an iid sample of size  $n$  taken from it. The sample may be summarized by a binomial random variable  $Y \sim B(n, p)$  with probability distribution

$$P(y) = P(Y = y) = \frac{n!}{y!(n - y)!} p^y (1 - p)^{n - y} \tag{1}$$

for integer  $y, 0 \leq y \leq n$ . Let  $Y_\vee = \max\{Y, n - Y\}$ . The probability distribution of  $Y_\vee$  is

$$\begin{aligned} P_\vee(y) &= P(Y_\vee = y) \\ &= \begin{cases} P(Y_\vee = y), & \text{for } y > n - y \text{ or } y > n/2; \\ P(Y_\vee = y), & \text{for } y = n - y \text{ or } y = n/2. \end{cases} \\ &= \begin{cases} \frac{n!}{y!(n - y)!} [p_\vee^y (1 - p_\vee)^{n - y} + p_\vee^{n - y} (1 - p_\vee)^y], & \text{for } n/2 < y \leq n; \\ \frac{n!}{y!(n - y)!} [p_\vee (1 - p_\vee)]^{n/2}, & \text{for } y = n/2. \end{cases} \end{aligned} \tag{2}$$

$P_\vee(y)$  of (2) is referred to as the entropic binomial distribution. It is to be noted that the entropic binomial distribution is parameterized by the entropic parameter,  $p_\vee$ , and not by the binomial parameter  $p$ . Also to be noted is the fact that all the probabilities in (2) are entropies by Definition 1. Furthermore, it is to be noted that the binomial probability of (1) is defined on a binomial sample space, while (2) is defined on an aggregated binomial sample space. The difference between the two is an important point to be exploited in this article.

Consider a mixture of several Bernoulli populations, each of which has probability  $p_j$ , for integers  $1 \leq j \leq J$ , with non-negative mixing weights,  $\lambda_j$ , such that  $\sum_{j=1}^J \lambda_j = 1$ . An iid sample of size  $n$  may be summarized into  $\{(N_j, Y_j); 1 \leq j \leq J\}$ , where  $\{N_j; 1 \leq j \leq J\}$

is a multinomial vector with size  $n$  and category probabilities  $\{\lambda_j; 1 \leq j \leq J\}$ , and, given  $\{N_j = n_j; 1 \leq j \leq J\}$ ,  $\{Y_j|N_j = n_j; 1 \leq j \leq J\}$  is a vector of independent binomial random variables with probabilities  $\{p_j; 1 \leq j \leq J\}$ . It then follows that the probability distribution of  $\{(N_j, Y_j); 1 \leq j \leq J\}$  is as given below. Writing  $\mathbf{N} = \{N_j; 1 \leq j \leq J\}$ ,  $\mathbf{n} = \{n_j; 1 \leq j \leq J\}$  as a realization of  $\mathbf{N}$ ,  $\mathbf{Y} = \{Y_j; 1 \leq j \leq J\}$  and  $\mathbf{y} = \{y_j; 1 \leq j \leq J\}$  as a realization of  $\mathbf{Y}$ ,

$$\begin{aligned}
 P((\mathbf{N}, \mathbf{Y}) = (\mathbf{n}, \mathbf{y})) &= P(\mathbf{Y} = \mathbf{y} | \mathbf{N} = \mathbf{n}) P(\mathbf{N} = \mathbf{n}) \\
 &= \left[ \prod_{j=1}^J \frac{n_j!}{y_j!(n_j - y_j)!} p_j^{y_j} (1 - p_j)^{n_j - y_j} \right] \left( \frac{n!}{n_1!n_2! \dots n_J!} \prod_{j=1}^J \lambda_j^{n_j} \right) \\
 &= \frac{n!}{\prod_{j=1}^J [y_j!(n_j - y_j)!]} \prod_{j=1}^J [\lambda_j^{n_j} p_j^{y_j} (1 - p_j)^{n_j - y_j}]. \tag{3}
 \end{aligned}$$

Let  $Y_{j,\vee} = \max\{Y_j, N_j - Y_j\}$  and let  $y_{j,\vee} = \max\{y_j, n_j - y_j\}$  be a realization of  $Y_{j,\vee}$  for every  $j$ ,  $1 \leq j \leq J$ . The probability distribution of  $\{(N_j, Y_{j,\vee}); 1 \leq j \leq J\}$  is as given in (4) below. Let  $\mathbf{Y}_\vee = \{Y_{j,\vee}; j = 1, \dots, J\}$  where  $Y_{j,\vee} = \max\{Y_j, n_j - y_j\}$ , let  $\mathbf{y}_\vee = \{y_{j,\vee}; 1 \leq j \leq J\}$  be a realization of  $\mathbf{Y}_\vee$ , and let  $p_{j,\vee} = \max\{p_j, 1 - p_j\}$  for every  $j$ ,  $1 \leq j \leq J$ . It is easily seen that, by way of (3), assuming  $p_j \neq 0.5$  for every  $j$  and for  $y_j$ ,  $n_j/2 \leq y_j \leq n_j$  for all  $j$ ,  $1 \leq j \leq J$ ,

$$\begin{aligned}
 P((\mathbf{N}, \mathbf{Y}_\vee) = (\mathbf{n}, \mathbf{y}_\vee)) &= P(\mathbf{Y}_\vee = \mathbf{y}_\vee | \mathbf{N} = \mathbf{n}) P(\mathbf{N} = \mathbf{n}) \\
 &= \frac{n!}{\prod_{j=1}^J [y_j!(n_j - y_j)!]} \prod_{j=1}^J \left\{ \lambda_j^{n_j} \left[ p_{j,\vee}^{y_j} (1 - p_{j,\vee})^{n_j - y_j} + p_{j,\vee}^{n_j - y_j} (1 - p_{j,\vee})^{y_j} \mathbf{1}_{[y_j > n_j/2]} \right] \right\}. \tag{4}
 \end{aligned}$$

**Remark 1.** The probability in (4) is not an entropy in the sense of Definition 1 but a product of entropies, each of which is defined with respect to a Bernoulli sub-population indexed by  $j$ ,  $1 \leq j \leq J$ .

### 2.3. Levels of Confidence and Utility

In the two-stage contemplation of constructing a tree classifier described in Section 1, there are two desirable events as follows, both of which pertain to the  $(n + 1)$  th observation  $(B_{n+1}, X_{n+1})$ .

1.  $C = X_{n+1}$  falls into a sub-population (or a tree node), say  $X_{n+1} = x_j$  for some  $j$ , where the classifier correctly identifies the true letter based on the sample of size  $n$ ,  $\{(B_i, X_i); 1 \leq i \leq n\}$ .
2.  $U = B_{n+1}$  is correctly predicted based on the sample of size  $n$ ,  $\{(B_i, X_i); 1 \leq i \leq n\}$ .

**Definition 2.** The probability of  $C$ ,  $P(C)$ , is referred to as the level of confidence, and the probability of  $U$ ,  $P(U)$ , is referred to as the level of utility.

To be instructive, consider first the levels of confidence and utility in the case of a single Bernoulli population. Let  $e(n)$  be the indicator function that  $n$  is an even integer.

$$\begin{aligned}
 P(C) &= P(\hat{\ell}_\vee = \ell_\vee) = P(Y > n - Y) + e(n) P(Y = n/2)/2 \\
 &= 1 - P(Y \leq n/2) + e(n) P(Y = n/2)/2 \\
 &= \sum_{n/2 < y \leq n} \frac{n!}{y!(n - y)!} p_\vee^y (1 - p_\vee)^{n - y} + \frac{n!e(n)}{2[(n/2)!]^2} [p_\vee(1 - p_\vee)]^{n/2},
 \end{aligned} \tag{5}$$

$$P(U) = p_\vee P(C) + (1 - p_\vee)(1 - P(C)), \tag{6}$$

where  $Y \sim B(n, p_\vee)$ . It may be interesting to note that both (5) and (6) are entropies. The proof of the following fact is trivial.

**Fact 1.** Assuming  $p_\vee > 1 - p_\vee$ ,  $\lim_{n \rightarrow \infty} P(C) = 1$  and  $\lim_{n \rightarrow \infty} P(U) = p_\vee$ .

Tables 1 and 2 below give the level of confidence and the level of utility, according to (5) and (6), for several combinations of sample size  $n$  and  $p_V$ . Given a desired level of confidence or a level of utility, an appropriate sample size may be found at every level of  $p_V$ . For example, in Table 1, at a desired confidence level of 95% and  $p_V = 0.75$ , the minimum sample size is  $n = 9$ . Similarly to reach a utility level of 0.70 with  $p_V = 0.75$ , a minimum sample of size  $n = 5$  is required. In practice, however,  $p_V$  is unknown, and therefore, either an empirical value exists or it needs to be estimated to make a judgment as to whether a sample size is adequate. The estimation of  $p_V$  is discussed in the next subsection.

**Table 1.** Confidence levels,  $P(C)$ , as a function of  $n$  and  $p_V$ .

$p_V$	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 1$	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 2$	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 3$	0.50	0.57	0.65	0.72	0.78	0.84	0.90	0.94	0.97	0.99	1.00
$n = 4$	0.50	0.57	0.65	0.72	0.78	0.84	0.90	0.94	0.97	0.99	1.00
$n = 5$	0.50	0.59	0.68	0.76	0.84	0.90	0.94	0.97	0.99	1.00	1.00
$n = 6$	0.50	0.59	0.68	0.76	0.84	0.90	0.94	0.97	0.99	1.00	1.00
$n = 7$	0.50	0.61	0.71	0.80	0.87	0.93	0.97	0.99	1.00	1.00	1.00
$n = 8$	0.50	0.61	0.71	0.80	0.87	0.93	0.97	0.99	1.00	1.00	1.00
$n = 9$	0.50	0.62	0.73	0.83	0.90	0.95	0.98	0.99	1.00	1.00	1.00
$n = 10$	0.50	0.62	0.73	0.83	0.90	0.95	0.98	0.99	1.00	1.00	1.00
$n = 15$	0.50	0.65	0.79	0.89	0.95	0.98	1.00	1.00	1.00	1.00	1.00
$n = 20$	0.50	0.67	0.81	0.91	0.97	0.99	1.00	1.00	1.00	1.00	1.00
$n = 25$	0.50	0.69	0.85	0.94	0.98	1.00	1.00	1.00	1.00	1.00	1.00
$n = 30$	0.50	0.71	0.86	0.95	0.99	1.00	1.00	1.00	1.00	1.00	1.00
$n = 35$	0.50	0.72	0.89	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00
$n = 40$	0.50	0.74	0.90	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 45$	0.50	0.75	0.91	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 50$	0.50	0.76	0.92	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 60$	0.50	0.78	0.94	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 70$	0.50	0.80	0.95	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 80$	0.50	0.81	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 90$	0.50	0.83	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 100$	0.50	0.84	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 200$	0.50	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$n = 300$	0.50	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

In the case of a tree structure, where there are at least  $J \geq 2$  nodes, the forms of  $P(C)$  and  $P(U)$  are slightly more complex. First, let it be noted that  $\mathbf{N} = \{N_1, \dots, N_J\}$  is a multinomial random vector with fixed size  $n$  and multinomial probabilities,  $\lambda = \{\lambda_1, \dots, \lambda_J\}$ , that is,

$$P(\mathbf{N} = \{n_1, \dots, n_J\}) = \binom{n}{n_1, \dots, n_J} \prod_{j=1}^J \lambda_j^{n_j}. \tag{7}$$

However, the marginal distribution of each  $N_j$  is a binomial, that is,

$$P(N_j = n_j) = \binom{n}{n_j} \lambda_j^{n_j} (1 - \lambda_j)^{n - n_j} \tag{8}$$

subject to  $0 \leq n_j \leq n$ . Therefore, for each  $j$ ,  $1 \leq j \leq J$ , letting  $C_j = \{\hat{\ell}_{j,V} = \ell_{j,V}\}$  be the event that the true letter at the  $j$ th node is correctly identified,

$$\begin{aligned}
 P(C_j) &= P(\hat{\ell}_{j,\vee} = \ell_{j,\vee}) \\
 &= \sum_{m=1}^n P(\hat{\ell}_{j,\vee} = \ell_{j,\vee} | N_j = m) P(N_j = m) \\
 &= \sum_{m=1}^n \left\{ [1 - P(Y_j \leq m/2) + e(m) P(Y_j = m/2)/2] \left[ \frac{n!}{m!(n-m)!} \lambda_j^m (1 - \lambda_j)^{n-m} \right] \right\} \quad (9)
 \end{aligned}$$

where  $Y_j \sim B(m, p_{j,\vee})$ ;

$$P(C) = \sum_{j=1}^J \lambda_j P(C_j) \quad \text{and} \quad (10)$$

$$P(U) = \sum_{j=1}^J \lambda_j [p_{j,\vee} P(C_j) + (1 - p_{j,\vee})(1 - P(C_j))] \quad (11)$$

where  $P(C_j)$  is as in (9).

**Table 2.** Utility levels,  $P(U)$ , as a function of  $n$  and  $p_{\vee}$ .

$p_{\vee}$	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 1$	0.50	0.51	0.52	0.55	0.58	0.63	0.68	0.75	0.82	0.91	1.00
$n = 2$	0.50	0.51	0.52	0.55	0.58	0.63	0.68	0.75	0.82	0.91	1.00
$n = 3$	0.50	0.51	0.53	0.57	0.61	0.67	0.74	0.81	0.88	0.94	1.00
$n = 4$	0.50	0.51	0.53	0.57	0.61	0.67	0.74	0.81	0.88	0.94	1.00
$n = 5$	0.50	0.51	0.54	0.58	0.63	0.70	0.77	0.83	0.89	0.95	1.00
$n = 6$	0.50	0.51	0.54	0.58	0.63	0.70	0.77	0.83	0.89	0.95	1.00
$n = 7$	0.50	0.51	0.54	0.59	0.65	0.71	0.78	0.84	0.90	0.95	1.00
$n = 8$	0.50	0.51	0.54	0.59	0.65	0.71	0.78	0.84	0.90	0.95	1.00
$n = 9$	0.50	0.51	0.54	0.60	0.66	0.73	0.79	0.85	0.90	0.95	1.00
$n = 10$	0.50	0.51	0.54	0.60	0.66	0.73	0.79	0.85	0.90	0.95	1.00
$n = 15$	0.50	0.52	0.56	0.62	0.68	0.74	0.80	0.85	0.90	0.95	1.00
$n = 20$	0.50	0.52	0.56	0.62	0.69	0.75	0.80	0.85	0.90	0.95	1.00
$n = 25$	0.50	0.52	0.56	0.63	0.69	0.75	0.80	0.85	0.90	0.95	1.00
$n = 30$	0.50	0.52	0.57	0.64	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 35$	0.50	0.52	0.58	0.64	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 40$	0.50	0.52	0.58	0.64	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 45$	0.50	0.53	0.58	0.64	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 50$	0.50	0.53	0.58	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 60$	0.50	0.53	0.59	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 70$	0.50	0.53	0.59	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 80$	0.50	0.53	0.59	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 90$	0.50	0.53	0.59	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 100$	0.50	0.53	0.59	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 200$	0.50	0.54	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$n = 300$	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00

**Example 1.** Suppose a binary tree classifier has  $J = 2$  nodes. The three parameters of the binary classifier are  $\lambda$ , and  $p_{1,\vee}$  and  $p_{2,\vee}$ , where  $\lambda$  is the partition weights,  $\lambda_1 = \lambda$  and  $\lambda_2 = 1 - \lambda$ , and  $p_{1,\vee}$  and  $p_{2,\vee}$  are the maximum probabilities in two partitions, respectively. By (9),

$$\begin{aligned}
 P(C_1) &= \sum_{m=1}^n \left\{ [1 - P(Y_1 \leq m/2) + e(m) P(Y_1 = m/2)/2] \left[ \frac{n!}{m!(n-m)!} \lambda^m (1-\lambda)^{n-m} \right] \right\} \\
 P(C_2) &= \sum_{m=1}^n \left\{ [1 - P(Y_2 \leq m/2) + e(m) P(Y_2 = m/2)/2] \left[ \frac{n!}{m!(n-m)!} (1-\lambda)^m \lambda^{n-m} \right] \right\}. \\
 P(C) &= \lambda P(C_1) + (1-\lambda) P(C_2), \tag{12}
 \end{aligned}$$

$$\begin{aligned}
 P(U) &= \lambda [p_{1,\vee} P(C_1) + (1-p_{1,\vee})(1-P(C_1))] \\
 &+ (1-\lambda) [p_{2,\vee} P(C_2) + (1-p_{2,\vee})(1-P(C_2))]. \tag{13}
 \end{aligned}$$

Tables 3–8 show calculated levels of confidence and utility for several combined values of the underlying parameters,  $\{\lambda, p_{1,\vee}, p_{2,\vee}\}$ , according to (12) and (13).

**Table 3.** Confidence level,  $P(C)$ , with  $J = 2$  and  $\lambda = 0.5$ .

$(p_{1,\vee}, p_{2,\vee})$	(0.6, 0.6)	(0.6, 0.7)	(0.6, 0.8)	(0.6, 0.9)	(0.7, 0.7)	(0.8, 0.8)	(0.9, 0.9)
$n = 5$	0.5358	0.5752	0.6107	0.6419	0.6145	0.6855	0.7480
$n = 10$	0.6045	0.6577	0.7017	0.7360	0.7109	0.7989	0.8675
$n = 20$	0.6574	0.7252	0.7726	0.8017	0.7930	0.8879	0.9460
$n = 30$	0.6941	0.7691	0.8136	0.8351	0.8441	0.9331	0.9761
$n = 40$	0.7232	0.8015	0.8410	0.8561	0.8798	0.9588	0.9890
$n = 50$	0.7475	0.8268	0.8609	0.8712	0.9061	0.9742	0.9949
$n = 100$	0.8303	0.8999	0.9137	0.9151	0.9696	0.9971	0.9999
$n = 200$	0.9130	0.9546	0.9565	0.9565	0.9961	1.0000	1.0000
$n = 300$	0.9524	0.9759	0.9762	0.9762	0.9994	1.0000	1.0000

**Table 4.** Confidence level,  $P(C)$ , with  $J = 2$  and  $\lambda = 0.75$ .

$(p_{1,\vee}, p_{2,\vee})$	(0.6, 0.6)	(0.6, 0.7)	(0.6, 0.8)	(0.6, 0.9)	(0.7, 0.7)	(0.8, 0.8)	(0.9, 0.9)
$n = 5$	0.5392	0.5500	0.5601	0.5694	0.6341	0.7159	0.7820
$n = 10$	0.6215	0.6384	0.6536	0.6670	0.7447	0.8347	0.8921
$n = 20$	0.6925	0.7159	0.7356	0.7515	0.8371	0.9159	0.9514
$n = 30$	0.7338	0.7614	0.7832	0.7993	0.8819	0.9448	0.9688
$n = 40$	0.7650	0.7956	0.8183	0.8337	0.9093	0.9596	0.9781
$n = 50$	0.7900	0.8230	0.8459	0.8602	0.9275	0.9686	0.8941
$n = 100$	0.8679	0.9062	0.9259	0.9339	0.9664	0.9882	0.9963
$n = 200$	0.9314	0.9686	0.9792	0.9811	0.9877	0.9978	0.9998
$n = 300$	0.9565	0.9884	0.9935	0.9939	0.9944	0.9996	1.0000

**Table 5.** Confidence level,  $P(C)$ , with  $J = 2$  and  $\lambda = 0.90$ .

$(p_{1,\vee}, p_{2,\vee})$	(0.6, 0.6)	(0.6, 0.7)	(0.6, 0.8)	(0.6, 0.9)	(0.7, 0.7)	(0.8, 0.8)	(0.9, 0.9)
$n = 5$	0.5902	0.5917	0.5931	0.5944	0.7156	0.8140	0.8785
$n = 10$	0.6579	0.6611	0.6641	0.6669	0.8047	0.8906	0.9261
$n = 20$	0.7432	0.7487	0.7537	0.7581	0.8952	0.9466	0.9577
$n = 30$	0.7955	0.8024	0.8085	0.8138	0.9336	0.9633	0.9699
$n = 40$	0.8311	0.8391	0.8459	0.8517	0.9515	0.9702	0.9762
$n = 50$	0.8573	0.8660	0.8737	0.8795	0.9606	0.9740	0.9801
$n = 100$	0.9249	0.9365	0.9453	0.9514	0.9743	0.9833	0.9895
$n = 200$	0.9621	0.9764	0.9852	0.9898	0.9830	0.9917	0.9964
$n = 300$	0.9715	0.9868	0.9944	0.9974	0.9880	0.9956	0.9986

**Table 6.** Utility level,  $P(U)$ , with  $J = 2$  and  $\lambda = 0.5$ .

$(p_{1,\vee}, p_{2,\vee})$	(0.6, 0.6)	(0.6, 0.7)	(0.6, 0.8)	(0.6, 0.9)	(0.7, 0.7)	(0.8, 0.8)	(0.9, 0.9)
$n = 5$	0.5072	0.5265	0.5592	0.6028	0.5458	0.6113	0.6984
$n = 10$	0.5209	0.5526	0.6001	0.6574	0.5844	0.6794	0.7940
$n = 20$	0.5315	0.5743	0.6321	0.6942	0.6172	0.7328	0.8568
$n = 30$	0.5388	0.5882	0.6493	0.7099	0.6376	0.7600	0.8809
$n = 40$	0.5446	0.5983	0.6560	0.7179	0.6519	0.7753	0.8912
$n = 50$	0.5495	0.6060	0.6670	0.7227	0.6624	0.7845	0.8959
$n = 100$	0.5661	0.6269	0.6822	0.7330	0.6878	0.7983	0.9000
$n = 200$	0.5826	0.6405	0.6913	0.7413	0.6984	0.8000	0.9000
$n = 300$	0.5905	0.6451	0.6952	0.7452	0.6998	0.8000	0.9000
$n = \infty$	0.6000	0.6500	0.7000	0.7500	0.7000	0.8000	0.9000

**Table 7.** Utility level,  $P(U)$ , with  $J = 2$  and  $\lambda = 0.75$ .

$(p_{1,\vee}, p_{2,\vee})$	(0.6, 0.6)	(0.6, 0.7)	(0.6, 0.8)	(0.6, 0.9)	(0.7, 0.7)	(0.8, 0.8)	(0.9, 0.9)
$n = 5$	0.5078	0.5037	0.5035	0.5067	0.5536	0.6296	0.7256
$n = 10$	0.5243	0.5311	0.5436	0.5608	0.5979	0.7008	0.8137
$n = 20$	0.5385	0.5522	0.5731	0.5988	0.6348	0.7495	0.8611
$n = 30$	0.5468	0.5636	0.5880	0.6166	0.6528	0.7669	0.8751
$n = 40$	0.5530	0.5721	0.5988	0.6287	0.6637	0.7757	0.8825
$n = 50$	0.5580	0.5790	0.6071	0.6375	0.6710	0.7811	0.8873
$n = 100$	0.5736	0.6000	0.6305	0.6560	0.6866	0.7929	0.8970
$n = 200$	0.5863	0.6162	0.6450	0.6711	0.6949	0.7987	0.8998
$n = 300$	0.5913	0.6216	0.6485	0.6738	0.6978	0.7997	0.9000
$n = \infty$	0.6000	0.6250	0.6500	0.6750	0.7000	0.8000	0.9000

**Table 8.** Utility level,  $P(U)$ , with  $J = 2$  and  $\lambda = 0.90$ .

$(p_{1,\vee}, p_{2,\vee})$	(0.6, 0.6)	(0.6, 0.7)	(0.6, 0.8)	(0.6, 0.9)	(0.7, 0.7)	(0.8, 0.8)	(0.9, 0.9)
$n = 5$	0.5180	0.5106	0.5037	0.4973	0.5862	0.6884	0.8028
$n = 10$	0.5516	0.5277	0.5250	0.5233	0.6219	0.7343	0.8408
$n = 20$	0.5486	0.5495	0.5521	0.5564	0.6581	0.7680	0.8662
$n = 30$	0.5591	0.5622	0.5675	0.5747	0.6734	0.7780	0.8759
$n = 40$	0.5662	0.5705	0.5774	0.5861	0.6806	0.7821	0.8810
$n = 50$	0.5715	0.5765	0.5843	0.5940	0.6842	0.7844	0.8841
$n = 100$	0.5850	0.5922	0.6024	0.6140	0.6897	0.7900	0.8916
$n = 200$	0.5924	0.6019	0.6137	0.6258	0.6932	0.7950	0.8971
$n = 300$	0.5943	0.6050	0.6171	0.6287	0.6952	0.7974	0.8989
$n = \infty$	0.6000	0.6100	0.6200	0.6300	0.7000	0.8000	0.9000

Consider two possible splits of the Bernoulli population with probability  $p_{\vee} = 0.70$ , along with a no-split, as follows.

1. Split A:  $\lambda = 0.9$ ,  $p_{1,\vee} = 0.75$  (hence  $p_{2,\vee} = 0.75$ );
2. Split B:  $\lambda = 0.2$ ,  $p_{1,\vee} = 0.75$  (hence  $p_{2,\vee} = 0.6875$ );
3. No Split:  $\lambda = 1$  and  $p_{\vee} = 0.70$ .

The levels of confidence as functions of the sample size  $n$  are plotted in Figure 1, where the thick solid curve is of No Split, the thin solid curve is of Split A, and the dashed curve is of Split B. Similarly, Figure 2 shows the three curves for the levels of utility. The fact that these curves cross, converge and dominate provides a basis for contemplation in the process of constructing and evaluating classifiers.



Calculations of  $P(C)$  and  $P(U)$  for more general cases may be carried out easily according (10) and (11).

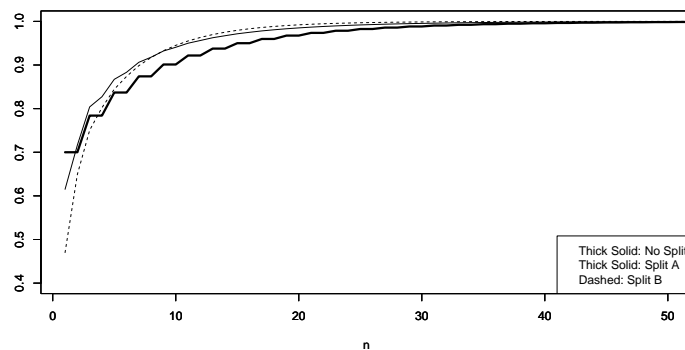


Figure 1. Confidence levels of competing splits.

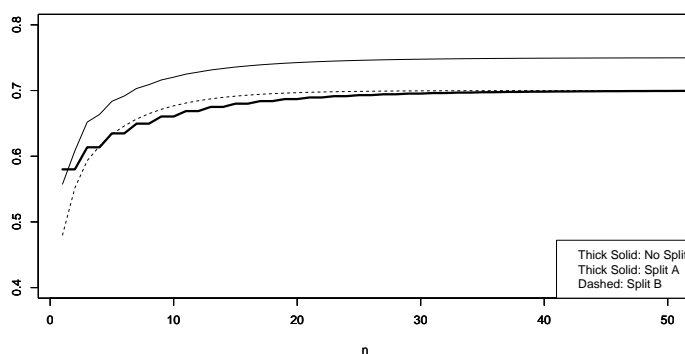


Figure 2. Utility levels of competing splits.

### 2.4. MLE and Entropic MLE

In practice, both the confidence and utility levels are unknown and therefore need to be estimated. Consider first the case of a homogeneous Bernoulli population. Since  $P(C)$  and  $P(U)$  are functions of  $p_V$ , their estimation boils down to that of  $p_V = \max\{p, 1 - p\}$ . Perhaps the most natural estimator of  $p_V$  is the maximum likelihood estimator (*mle*) under the binomial distribution in (1),

$$\hat{p}_V = \max\{\hat{p}, 1 - \hat{p}\} \tag{14}$$

where  $\hat{p} = Y/n$ , and the corresponding *mles* of  $P(C)$  and  $P(U)$  in (5) and (6) are

$$\hat{P}(C) = \sum_{y=\lfloor n/2 \rfloor + 1}^n \frac{n!}{y!(n-y)!} \hat{p}_V^y (1 - \hat{p}_V)^{n-y} + \frac{n!/2}{(n/2)!(n/2)!} [\hat{p}_V(1 - \hat{p}_V)]^{n/2} \times e(n), \tag{15}$$

$$\hat{P}(U) = 1 - \hat{P}(C)(1 - \hat{p}_V) - \hat{p}_V(1 - \hat{P}(C)). \tag{16}$$

The following three facts collectively indicate a tendency of over-estimation by (15) and (16).

**Fact 2.** Suppose  $p \in (0, 1)$ . Then  $E(\hat{p}_V) > p_V$ .

A proof of Fact 2 is given in Section 1.

**Fact 3.** The confidence level,  $P(C)$  of (5), is an increasing function of  $p_V$ .

**Proof.** It is best to prove the fact in two separate cases:  $n$  is odd and even. Assuming  $n$  is odd, rewriting (10) gives

$$\begin{aligned}
 P(C) &= \sum_{y=\lfloor n/2 \rfloor + 1}^n \frac{n!}{y!(n-y)!} p_v^y (1-p_v)^{n-y} + \frac{n!/2}{(n/2)!(n/2)!} [p_v(1-p_v)]^{n/2} \times 1_{[n \text{ is even}]} \\
 &= \sum_{y=\lfloor n/2 \rfloor + 1}^n \frac{n!}{y!(n-y)!} p_v^y (1-p_v)^{n-y} = \sum_{y=(n+1)/2}^n \frac{n!}{y!(n-y)!} p_v^y (1-p_v)^{n-y} \\
 &= P(Y \geq (n+1)/2) = P(2Y \geq n+1) = P(Y \geq (n-Y) + 1) \\
 &= P(Y > n-Y) \tag{17}
 \end{aligned}$$

where  $Y$  is a binomial random variable with distribution  $B(n, p_v)$ . Noting that  $p_v > 0.5$  by assumption and that the event  $\{Y > n - Y\}$  is the event of “more successes than failures in a sample of size  $n$ ”, it follows that  $P(C)$  increases as  $p_v$  does.

Similarly assuming  $n$  is even,

$$\begin{aligned}
 P(C) &= P(Y \geq (n+1)/2) + P(Y = n/2) - \frac{n!/2}{(n/2)!(n/2)!} [p_v(1-p_v)]^{n/2} \\
 &= P(Y \geq n/2) - \frac{n!/2}{(n/2)!(n/2)!} [p_v(1-p_v)]^{n/2} \\
 &= P(2Y \geq n) - \frac{n!/2}{(n/2)!(n/2)!} [p_v(1-p_v)]^{n/2} \\
 &= P(Y \geq n - Y) - \frac{n!/2}{(n/2)!(n/2)!} [p_v(1-p_v)]^{n/2} \tag{18}
 \end{aligned}$$

where  $Y$  is a binomial random variable with distribution  $B(n, p_v)$ . Noting that  $p_v > 0.5$  by assumption and that the event  $\{Y > n - Y\}$  is the event of “no fewer successes than failures”, it follows that  $P(Y \geq n - Y)$  increases as  $p_v$  does. On the other hand, the negative term in (18) is a strictly decreasing function of  $p_v$  for  $p_v \in [0.5, 1)$ . It follows that  $P(C)$  is increasing in  $p_v$ . □

**Fact 4.** The utility level,  $P(U)$  of (11), is an increasing function of  $p_v$ .

**Proof.** Noting  $P(U) = P(C)p_v + (1 - P(C))(1 - p_v)$  of (11), taking the derivative of  $P(U)$  with respect to  $p_v$ , and letting  $P'(C)$  denote the derivative of  $P(C)$  with respect to  $p_v$ ,

$$\begin{aligned}
 P'(U) &= P'(C)p_v + P(C) - P'(C)(1 - p_v) - (1 - P(C)) \\
 &= P'(C)(2p_v - 1) + (2P(C) - 1). \tag{19}
 \end{aligned}$$

Noting  $P'(C) > 0$  as shown in Fact 3,  $p_v > 0.5$  (and hence  $2p_v > 1$ ) and  $P(C) \geq 0.5$  (and hence  $2P(C) \geq 1$ ), it follows that  $P'(U) > 0$  for  $p_v \in (0.5, 1)$ . □

On the other hand, let the maximizing value of  $p_v$  in the likelihood of the entropic binomial distribution (2) be referred to as the entropic maximum likelihood estimator (*emle*), denoted  $\tilde{p}_v$ , and let  $g(\tilde{p}_v)$ , for any function  $g(\cdot)$ , be referred to as the *emle* of  $g(p_v)$ .  $\tilde{p}_v$  tends to underestimate  $p_v$  with smaller samples. However, it provides an opportunity to offset the upward bias of the *mle*,  $\hat{p}_v$ , in various ways, for example, by means of a weighted average

$$\hat{p}_v^b = w\hat{p}_v + (1 - w)\tilde{p}_v \tag{20}$$

where  $w, 0 \leq w \leq 1$ , may be data-based. More specifically, the following is the proposed estimator of  $p_v$  of this article, with  $w = \hat{p}_v$ .

$$\hat{p}_v^b = \hat{p}_v \hat{p}_v + (1 - \hat{p}_v)\tilde{p}_v = \tilde{p}_v + \hat{p}_v(\hat{p}_v - \tilde{p}_v), \tag{21}$$

which may be viewed as an under-estimator,  $\tilde{p}_v$ , with a non-negative correction term,  $\hat{p}_v(\hat{p}_v - \tilde{p}_v)$ .

**Example 2.** Suppose an iid Bernoulli sample yields  $Y = 5$  and  $n - Y = 2$ . The likelihood of the binomial distribution (1) is proportional to  $p^y(1 - p)^{n-y}$ , the dashed curve given in Figure 3, and the mle of  $p_V$  is  $\hat{p}_V = 5/7 = 0.7134$ , as the dashed arrow points to, and the mles of  $P(C)$  and  $P(U)$ , by (15) and (16), are, respectively,

$$\hat{P}(C) = \sum_{y=4}^7 \frac{7!}{y!(7-y)!} (5/7)^y (2/7)^{7-y} = 0.8917,$$

$$\begin{aligned} \hat{P}(U) &= 1 - 0.8917(1 - 5/7) - (5/7)(1 - 0.8917) = 1 - 0.8917(2/7) - (5/7)(0.1083) \\ &= 0.6679. \end{aligned}$$

The entropic likelihood of the entropic binomial distribution (2) is represented by the solid curve in Figure 3. The entropic maximum likelihood estimator (emle) of  $p_V$  is  $\tilde{p}_V = 0.6667$ , as the solid arrow points at, and the emles of  $P(C)$  and  $P(U)$ , by (15) and (16), are, respectively,

$$\tilde{P}(C) = \sum_{y=4}^7 \frac{7!}{y!(7-y)!} (0.6667)^y (1 - 0.6667)^{7-y} = 0.8267$$

$$\tilde{P}(U) = 1 - 0.8267(1 - 0.6667) - (0.6667)(1 - 0.8267) = 0.6089$$

By (21), (15) and (16),  $\hat{p}_V^b = 0.7000$ , and

$$\hat{P}^b(C) = \sum_{y=4}^7 \frac{7!}{y!(7-y)!} (0.7)^y (1 - 0.7)^{7-y} = 0.8740$$

$$\hat{P}^b(U) = 1 - 0.8740(1 - 0.7) - (0.7)(1 - 0.8740) = 0.6496.$$

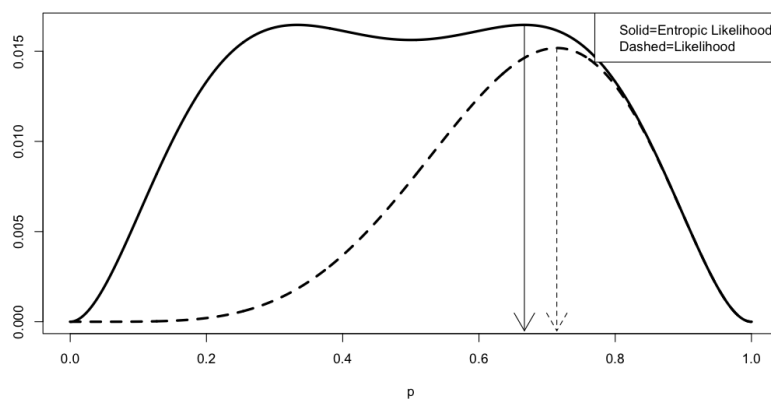
It may be interesting to note that, in this case,

$$\tilde{p}_V \leq \hat{p}_V^b \leq \hat{p}_V \tag{22}$$

$$\tilde{P}(C) \leq \hat{P}^b(C) \leq \hat{P}(C), \text{ and} \tag{23}$$

$$\tilde{P}(U) \leq \hat{P}^b(U) \leq \hat{P}(U). \tag{24}$$

It is to be mentioned that the qualitative difference between the two likelihood functions in Figure 3 remains the same in general: the values of emles are lower than those of mles. It is important to understand that the binomial distribution leads to an mle of  $p$ , and then one of  $p_V = \max\{p, 1 - p\}$ , while the entropic binomial distribution estimates  $p_V$  directly via the likelihood of  $Y_V$ . The inequalities in (22)–(24) are deterministically true in general, which gives an opportunity for a reduction in the biases of  $\hat{p}_V$ ,  $\hat{P}(C)$  and  $\hat{P}(U)$ .



**Figure 3.** Likelihood and entropic likelihood.

2.5. Several Numerical Studies

Several simulation studies were conducted, and the results are presented in Tables 16 and 17 for the bias and the mean squared errors (MSE) of  $\hat{p}_V$ ,  $\tilde{p}_V$ , and  $\hat{p}_V^b$  respectively. Each simulated case is based on ten thousand random samples. Results are tabulated for combinations of four  $p_V$  values {0.6, 0.7, 0.8, 0.9} crossed with ten values of  $n$  from 5 to 50. The MSE values are quite stable across the three estimators in question. Therefore, the comparison is mainly of the biases of the estimators. It is observed that the simulated bias of *emle*,  $\tilde{p}_V$ , is significantly lower than that of the *mle*,  $\hat{p}_V$ , for smaller values of  $p_V$  and with smaller samples, and the bias of the proposed weighted average estimator,  $\hat{p}_V^b$ , snugs in between. This is a fact well suggested by Facts 2–4. The weighted average,  $\hat{p}_V^b$ , seems to do better than  $\hat{p}_V$  across the board. In the study range of  $p_V$ , the bias of  $\hat{p}_V^b$  seems to be controlled when the sample size reaches  $n = 20$  or  $n = 30$ , as evidenced by the fact that the simulated bias is controlled under 2% and 1%, while the bias of  $\hat{p}_V$  becomes controlled when the sample size reaches  $n = 40$  to  $n = 50$ . The observed difference between the required sample sizes is the advantage of the proposed estimator. However, it is also observed that the required sample size to reach a reasonable precision, say the bias is under 2% or 1%, respectively, depends on the value of  $p_V$ . The smaller the  $p_V$ , the larger the sample required. In that sense, the biases tabulated in Tables 16 and 17 do not tell the whole story. In fact, the first part of Table 9 contains the biases for a very small value of  $p_V = 0.51$ , which shows that the *mle*,  $\hat{p}_V$ , needs  $n = 200$  to have bias under 2% and  $n = 500$  to have a bias under 1%. It is once again seen that the proposed estimator,  $\hat{p}_V^b$ , performs much better.

**Table 9.** Biases with very small  $p_V$ .

Bias of	$\hat{p}_V$	$\tilde{p}_V$	$\hat{p}_V^b$	$\hat{P}(C)$	$\hat{P}^b(C)$	$\hat{P}(U)$	$\hat{P}^b(U)$
$p_V$	0.51	0.51	0.51	0.51	0.51	0.51	0.51
$n = 100$	0.0307	0.0121	0.0220	0.1847	0.1242	0.0268	0.0207
$n = 200$	0.0192	0.0062	0.0130	0.1518	0.0915	0.0186	0.0141
$n = 300$	0.0142	0.0042	0.0094	0.1284	0.0711	0.0148	0.0113
$n = 400$	0.0112	0.0023	0.0069	0.1094	0.0502	0.0124	0.0092
$n = 500$	0.0093	0.0016	0.0056	0.0951	0.0375	0.0108	0.0080
$n = 600$	0.0080	0.0009	0.0046	0.0826	0.0257	0.0096	0.0071
$n = 700$	0.0069	0.0005	0.0038	0.0710	0.0146	0.0086	0.0063
$n = 800$	0.0061	0.0001	0.0032	0.0608	0.0049	0.0079	0.0057

Several simulation results for estimating the levels of confidence and utility in the case of one single Bernoulli population are given in Tables 18 and 19. It is observed that the biases are quite large across the board for smaller values of  $\hat{p}_V$  and with smaller samples, albeit the biases are much smaller for the estimators of  $P(U)$  than for those of  $P(C)$ . For example, when  $p_V = 0.6$ , for the biases in Table 18 to be under 1%, a node size of  $n = 300$  is needed. When  $p_V = 0.7$ , for the biases in Table 18 to be under 1%, a node size of  $n = 70$  is needed. It may be interesting to note that, in Table 19, in estimating  $P(U)$  when  $p_V = 0.6$  for a bias under 1%, a node size of only  $n = 70$  is needed. When  $p_V = 0.7$ , for a bias under 1%, a node size of only  $n = 20$  is needed. An increasing trend in bias, as  $p_V$  decreases, is clearly observed. To give a more complete picture, the second part of Table 9 gives biases of the *mle* and the proposed estimators, of  $P(C)$  and  $P(U)$ , at  $p_V = 0.51$ . It is clearly seen that the *mle*,  $\hat{P}_V(C)$  of  $P(C)$  performs very poorly, while the proposed estimator,  $\hat{P}_V^b(C)$  does much better though is not necessarily satisfactory. It may be interesting to note that, in estimating  $P(U)$ , the *mle* and the proposed estimator are comparable in bias as the sample size varies. One could say that, for lack of a better term,  $P(U)$  is easier to estimate than  $P(C)$ .

The simulation studies reported above give a glimpse of how  $p_V$ , the level of confidence and the level of utility may be estimated on a very limited scope, mostly focusing on

a single Bernoulli population. However, more general and more complex situations may be easily carried out in similar manners.

**Example 3.** One of the most popular illustrative examples of a decision tree in data science involves predicting whether a randomly selected golfer goes to play the game under a set of weather conditions. A sample of  $n = 14$  is given in Table 10.

**Table 10.** Golfing and weather.

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Many questions about this data set may be asked. For illustration purposes, let the question be, if only one covariate is used, which of the four, among Outlook, Temp, Humidity and Wind, is the best predictor. Obviously, the sample size is too small to convey any meaningful reliability of the results and therefore is ignored. Several key statistics for each of the four factors are tabulated in Tables 11–14. The statistics include  $n_j, y_j, \hat{\lambda}_j, \hat{p}_{j,\nu}, \hat{P}(C_j),$  and  $\hat{P}(U_j)$  for each  $j$ , specifically noting that (9)–(11) are the basis of the plug-in estimators.

**Table 11.** Outlook with  $J = 3$ .

Outlook	Rainy	Overcast	Sunny
$n_j$	5	4	5
$y_j$	2	4	3
$\hat{\lambda}_j$	5/14	4/14	5/14
$\hat{p}_{j,\nu}$	3/5	4/4	3/5
$\hat{P}(C_j)$	0.8418	0.7397	0.8418
$\hat{P}(U_j)$	0.5684	0.7397	0.5684

**Table 12.** Temperature with  $J = 3$ .

Temp	Hot	Mild	Cool
$n_j$	4	6	4
$y_j$	2	4	3
$\hat{\lambda}_j$	4/14	6/14	4/14
$\hat{p}_{j,\nu}$	2/4	4/6	3/4
$\hat{P}(C_j)$	0.6426	0.9408	0.7268
$\hat{P}(U_j)$	0.5000	0.6470	0.6134

**Table 13.** Humidity with  $J = 2$ .

Humidity	High	Normal
$n_j$	7	7
$y_j$	3	6
$\hat{\lambda}_j$	7/14	7/14
$\hat{p}_{j,\mathcal{V}}$	4/7	6/7
$\hat{P}(C_j)$	0.9246	0.9915
$\hat{P}(U_j)$	0.5606	0.8511

**Table 14.** Wind with  $J = 2$ .

Windy	False	True
$n_j$	8	6
$y_j$	6	3
$\hat{\lambda}_j$	8/14	6/14
$\hat{p}_{j,\mathcal{V}}$	6/8	3/6
$\hat{P}(C_j)$	0.9916	0.8547
$\hat{P}(U_j)$	0.7458	0.5000

The estimated overall levels of confidence and utility are tabulated for each of the four covariates in Table 15. For comparison, the estimated Gini’s information impurities for the respective covariates are also tabulated. It is clear that for the estimated levels of confidence and utility, Humidity is the best predictor, followed by Wind and then Outlook, and Temp is the worst. Incidentally, the estimated Gini’s information impurities also support the same ranking (Tables 16–19).

**Table 15.** Estimated levels of confidence and utility.

Weather	Outlook	Temp	Humidity	Windy
$\hat{P}(C)$	0.8126	0.7945	0.9581	0.9329
$\hat{P}(U)$	0.6173	0.5954	0.7059	0.6405
$\hat{g}_\lambda$	0.7143	0.7976	0.3673	0.4286

**Table 16.** Biases in estimating  $p_{\mathcal{V}}$ .

Bias of	$\hat{p}_{\mathcal{V}}$				$\tilde{p}_{\mathcal{V}}$				$\hat{p}_{\mathcal{V}}^{\mathcal{V}}$			
	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9
$n = 5$	0.1021	0.0468	0.0158	0.0037	0.0468	−0.0014	−0.0141	−0.0077	0.0781	0.0284	0.0048	−0.0001
$n = 6$	0.0743	0.0278	0.0076	0.0025	−0.0016	−0.0370	−0.0364	−0.0136	0.0490	0.0063	−0.0070	−0.0028
$n = 7$	0.0740	0.0282	0.0073	0.0020	0.0231	−0.0111	−0.0160	−0.0055	0.0544	0.0137	−0.0008	−0.0004
$n = 8$	0.0578	0.0183	0.0048	0.0013	0.0056	−0.0212	−0.0165	−0.0036	0.0385	0.0038	−0.0029	−0.0004
$n = 9$	0.0578	0.0184	0.0040	0.0008	−0.0202	−0.0434	−0.0301	−0.0070	0.0292	−0.0036	−0.0078	−0.0018
$n = 10$	0.0471	0.0125	0.0024	0.0008	0.0058	−0.0161	−0.0101	−0.0013	0.0310	0.0016	−0.0022	0.0000
$n = 20$	0.0212	0.0031	0.0010	0.0007	−0.0138	−0.0140	−0.0019	0.0006	0.0067	−0.0039	−0.0002	0.0006
$n = 30$	0.0122	0.0015	0.0007	0.0005	−0.0098	−0.0052	0.0002	0.0027	−0.0013	0.0005	0.0005	0.0005
$n = 40$	0.0078	0.0009	0.0008	0.0005	−0.0125	−0.0031	0.0007	0.0005	−0.0011	−0.0008	0.0007	0.0005
$n = 50$	0.0052	0.0007	0.0006	0.0004	−0.0093	−0.0009	0.0006	0.0004	−0.0012	0.0000	0.0006	0.0004

**Table 17.** Mean squared errors in estimating  $p_V$ .

MSE of	$\hat{p}_V$				$\tilde{p}_V$				$\hat{p}_V^b$			
	$p_V$	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8
$n = 5$	0.0250	0.0237	0.0232	0.0162	0.0320	0.0360	0.0358	0.0223	0.0282	0.0277	0.0276	0.0183
$n = 6$	0.0274	0.0237	0.0224	0.0144	0.0274	0.0386	0.0413	0.0244	0.0230	0.0263	0.0271	0.0171
$n = 7$	0.0193	0.0192	0.0189	0.0122	0.0209	0.0270	0.0271	0.0159	0.0197	0.0219	0.0217	0.0133
$n = 8$	0.0183	0.0191	0.0180	0.0109	0.0215	0.0292	0.0273	0.0138	0.0181	0.0218	0.0209	0.0118
$n = 9$	0.0150	0.0163	0.0157	0.0097	0.0200	0.0321	0.0146	0.0308	0.0147	0.0202	0.0198	0.0111
$n = 10$	0.0147	0.0161	0.0147	0.0088	0.0175	0.0233	0.0199	0.0100	0.0150	0.0184	0.0165	0.0093
$n = 20$	0.0076	0.0093	0.0078	0.0044	0.0114	0.0143	0.0092	0.0044	0.0086	0.0111	0.0083	0.0044
$n = 30$	0.0056	0.0067	0.0052	0.0029	0.0086	0.0054	0.0053	0.0029	0.0064	0.0074	0.0029	0.0029
$n = 40$	0.0045	0.0051	0.0039	0.0022	0.0071	0.0064	0.0040	0.0022	0.0054	0.0056	0.0040	0.0022
$n = 50$	0.0038	0.0041	0.0031	0.0018	0.0057	0.0046	0.0032	0.0018	0.0045	0.0043	0.0031	0.0018

**Table 18.** Biases in estimating  $P(C)$ .

Bias of	$\hat{P}(C)$				$\tilde{P}(C)$				$\hat{P}^b(C)$			
	$p_V$	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8
$n = 20$	-0.0004	-0.0610	-0.0213	-0.0014	-0.1192	-0.1183	-0.0311	-0.0016	-0.0479	-0.0834	-0.0251	-0.0015
$n = 30$	-0.0318	-0.0460	-0.0067	-0.0001	-0.1149	-0.0712	-0.0082	-0.0001	-0.0685	-0.0562	-0.0073	-0.0001
$n = 40$	-0.0479	-0.0309	-0.0020	-0.0000	-0.1423	-0.0489	-0.0023	-0.0000	-0.0872	-0.0382	-0.0022	-0.0000
$n = 50$	-0.0561	-0.0199	-0.0006	0.0000	-0.1301	-0.0279	-0.0007	0.0000	-0.0877	-0.0232	-0.0007	0.0000
$n = 60$	-0.0594	-0.0125	-0.0002	-0.0000	-0.1258	-0.0169	-0.0002	-0.0000	-0.0878	-0.0143	-0.0002	-0.0000
$n = 70$	-0.0599	-0.0079	-0.0000	-0.0000	-0.1245	-0.0104	-0.0000	-0.0000	-0.0875	-0.0089	-0.0000	-0.0000
$n = 80$	-0.0595	-0.0049	-0.0000	-0.0000	-0.1116	-0.0060	-0.0000	-0.0000	-0.0820	-0.0054	-0.0000	-0.0000
$n = 90$	-0.0570	-0.0031	-0.0000	-0.0000	-0.1023	-0.0036	-0.0000	-0.0000	-0.0766	-0.0033	-0.0000	-0.0000
$n = 100$	-0.0531	-0.0020	-0.0000	-0.0000	-0.0993	-0.0025	-0.0000	-0.0000	-0.0729	-0.0022	-0.0000	-0.0000
$n = 200$	-0.0222	-0.0000	-0.0000	0.0000	-0.0337	-0.0000	-0.0000	0.0000	-0.0272	-0.0000	-0.0000	0.0000
$n = 300$	-0.0074	-0.0000	-0.0000	0.0000	-0.0096	-0.0000	-0.0000	0.0000	-0.0084	-0.0000	-0.0000	0.0000

**Table 19.** Biases in estimating  $P(U)$ .

Bias of	$\hat{P}(U)$				$\tilde{P}(U)$				$\hat{P}^b(U)$			
	$p_V$	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8
$n = 20$	0.0408	0.0024	-0.0043	0.0000	0.0252	-0.0065	-0.0062	0.0000	0.0302	-0.0033	-0.0054	0.0000
$n = 30$	0.0250	-0.0030	-0.0011	0.0006	0.0129	-0.0077	-0.0015	0.0006	0.0180	-0.0055	-0.0013	0.0006
$n = 40$	0.0158	-0.0033	0.0002	0.0005	0.0055	-0.0058	0.0001	0.0005	0.0090	-0.0048	0.0002	0.0005
$n = 50$	0.0098	-0.0024	0.0004	0.0005	0.0017	-0.0036	0.0004	0.0004	0.0048	-0.0030	0.0004	0.0005
$n = 60$	0.0060	-0.0014	0.0006	0.0004	-0.0014	-0.0021	0.0006	0.0004	0.0016	-0.0018	0.0006	0.0004
$n = 70$	0.0033	-0.0008	0.0005	0.0005	-0.0028	-0.0012	0.0005	0.0005	-0.0006	-0.0010	0.0005	0.0005
$n = 80$	0.0013	-0.0004	0.0005	0.0004	-0.0038	-0.0006	0.0005	0.0004	-0.0038	-0.0005	0.0005	0.0004
$n = 90$	-0.0001	-0.0000	0.0005	0.0004	-0.0046	-0.0001	0.0005	0.0004	-0.0028	-0.0001	0.0005	0.0004
$n = 100$	-0.0007	-0.0002	0.0004	0.0004	-0.0046	-0.0003	0.0004	0.0004	-0.0032	-0.0003	0.0004	0.0004
$n = 200$	-0.0021	-0.0002	0.0002	0.0002	-0.0030	-0.0002	0.0002	0.0002	-0.0027	-0.0002	0.0002	0.0002
$n = 300$	-0.0010	-0.0003	0.0002	0.0002	-0.0011	-0.0003	0.0002	0.0002	-0.0011	-0.0003	0.0002	0.0002

### 3. Summary

This article proposes two performance measures,  $P(C)$  and  $P(U)$ , that are linked to probabilities of two desirable label-invariant events in the sampling/developing process of a binary tree construction. They are referred to as the level of confidence and the level of utility of a binary classifier. A core component of these measures is the larger of the probabilities in a Bernoulli trial, that is,  $p_V = \max\{p, 1 - p\}$ . Several properties of  $p_V$ ,  $P(C)$

and  $P(U)$  are discussed. Also discussed is the estimation of these quantities. However, let it be noted that, although  $P(C)$  and  $P(U)$  are the measures of central interest in this article, the estimation of  $p_V$  is important in its own right, since it could be a key element in evaluating many aspects of a binary classifier beyond those considered in this article.

One of the most distinct features identified in this article is the upward bias of the usual *mle* of  $p_V$ , namely  $\hat{p}_V$ . This bias may be significant and increases as  $p_V$  decreases toward 0.5 with a fixed  $n$ . Because of that, the biases of the *mles* of  $P(C)$  and  $P(U)$ , namely  $\hat{P}(C)$  and  $\hat{P}(U)$ , have the same issues though to different extents. To control the said biases to within a reasonable bound, for example, 1% or 2%, a required sample size may need to be very large.

In terms of practice, several recommendations are made below, which may provide some useful guidance.

1. Small sample size considerations are important because, in developing a tree classifier, the perpetual question is whether to go further into the next layer, regardless of the macro modeling logic one may use. At the end of splitting, the sample size races toward zero. No matter what macro logic is employed in construction, a tree always comes to nodes to be developed with samples of smaller sizes. One of the most important questions is whether the sample size is sufficiently large to be statistically meaningful. To answer this question, the best approach is to have a prior empirical judgment on the range for  $p_V$ . If a range is judged as reasonable, say  $[p_a, p_b]$ , where  $0.5 < p_a < p_b \leq 1$ , then that  $p_a$  may be used to determine the appropriate sample size via Formulas (5) and (6) at a given desired level, say 95% for  $P(C)$  and another practically chosen level for  $P(U)$ , noting that  $P(U)$  has a ceiling, that is,  $P(U) \leq p_V$ , according to Fact 1.
2. If no sufficient prior knowledge exists for  $p_V$ , then a preliminary estimate for it is needed. The proposed estimator in (20),  $\hat{p}_V^b$ , is preferred to the usual *mle*,  $\hat{p}_V$ . The estimated  $p_V$  is then used in Formulas (5) and (6) to produce estimated levels of confidence and utility, which in turn could give baseline information for further adjustments, such as pruning or further splitting. Of course, in such estimation, a reasonable sample size is needed. A recommended initial minimum sample size, according to Table 17 with a reasonable range  $[0.6, 0.9]$ , is  $n = 40$  if  $\hat{p}_V$  is used and  $n = 20$  if  $\hat{p}_V^b$  is used.
3. For a given binary tree classifier with  $J \geq 2$  leaves or nodes, both the level of confidence and the level of utility after  $p_{j,V}$  is estimated for each and every  $j$ ,  $2 \leq j \leq J$ . The formulas of (10) and (11) may be used, with the *mle* of  $\lambda$ ,  $\hat{\lambda}_j = n_j/n$ , and the *mle* or the proposed estimator of  $P(C_j)$  and  $P(U_j)$  for each and every  $j$ ,  $1 \leq j \leq J$ , to produce estimates of overall levels of confidence and utility. Noting that both (10) and (11) are  $\lambda$ -weighted averages, the overall level of confidence or the overall level of utility may be negatively affected if individual nodes have particularly low  $P(C_j)$  or  $P(U_j)$  for some  $j$ ,  $1 \leq j \leq J$ . If individual nodes are found to be low in confidence or utility, some repair or adjustment may be called for.

The main objective of this paper is to add two measures of performance to the literature of development and evaluation of a binary tree classifier. The measures have intuitive and simple probabilistic meanings. As such, some basic questions, like the relationship between the parameters and sample sizes, may be naturally considered and described in a style of classic statistics. However, it must be noted that it is not meant to replace or take away anything from the collection of methodologies in modern data science. It is hoped that the discussion of this article serves as a starting point for much more to come as data science advances and evolves.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.



**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The author declares no conflicts of interest.

## References

1. Breiman, L. Statistical modeling: The two cultures (with discussion). *Stat. Sci.* **2001**, *16*, 199–231. [[CrossRef](#)]
2. Kass, R.E. The Two Cultures: Statistics and Machine Learning in Science. *Obs. Stud.* **2021**, *7*, 135–144. [[CrossRef](#)]
3. Tan, P.-N.; Steinbach, M.; Karpatne, A.; Kumar, V. *Introduction to Data Mining*, 2nd ed.; Pearson: London, UK, 2018.
4. Zhang, Z. Entropy-Based Statistics and Their Applications. *Entropy* **2023**, *25*, 936. [[CrossRef](#)]
5. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423, 623–656. [[CrossRef](#)]
6. Rényi, A. On measures of information and entropy. *Berkeley Symp. Math. Stat. Probab.* **1961**, *4.1*, 547–561.
7. Tsallis, C. Possible generalization of Boltzmann-Gibbs statistics. *J. Stat. Phys.* **1988**, *52*, 479–487. [[CrossRef](#)]
8. Simpson, E.H. Measurement of diversity. *Nature* **1949**, *163*, 688. [[CrossRef](#)]
9. Zhang, Z.; Zhou, J. Re-parameterization of multinomial distribution and diversity indices. *J. Stat. Plan. Inference* **2010**, *140*, 1731–1738. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.