EVALUATION METRICS IN CLASSIFICATION: A QUANTIFICATION OF DISTANCE-BIAS

RICARDO VILALTA AND DANIEL OBLINGER
IBM T.J. Watson Research Center, New York

This paper provides a characterization of bias for evaluation metrics in classification (e.g., Information Gain, Gini, χ^2, etc.). Our characterization provides a uniform representation for all traditional evaluation metrics. Such representation leads naturally to a measure for the distance between the bias of two evaluation metrics. We give a practical value to our measure by observing if the distance between the bias of two evaluation metrics correlates with differences in predictive accuracy when we compare two versions of the same learning algorithm that differ in the evaluation metric only. Experiments on real-world domains show how the expectations on accuracy differences generated by the distance-bias measure correlate with actual differences when the learning algorithm is simple (e.g., search for the best single-feature or the best single-rule). The correlation, however, weakens with more complex algorithms (e.g., learning decision trees). Our results show how interaction among learning components is a key factor to understand learning performance.

Key words: evaluation metric, bias, inductive learning, classification.

1. INTRODUCTION

The role of an evaluation metric during classification is to assess the value of potentially useful expressions (i.e., the value of single features or the combination of features). A classic scenario is to use an evaluation metric to rank all candidate expressions; the best expression is then used as a building block to model the final hypothesis. For example, top-down univariate decision-tree induction (Quinlan, 1986; Breiman, Friedman, Olshen, & Stone, 1984) recursively partitions the training set by finding at each tree node the feature (i.e, attribute, variable) that maximizes the purity, or class uniformity, of each induced example subset; similarly, cover-set rule induction (Clark & Niblett, 1989) requires evaluation of conjunction of feature values during the search for a final rule-set hypothesis.

The most common kind of evaluation metrics, *traditional* or *purity-based*, looks into the proportion of classes on each induced example subset; an optimal result is achieved if each subset comprises examples of the same class. Examples of traditional or purity-based evaluation metrics are Information Gain (Quinlan, 1986, 1994), Gain Ratio (Quinlan, 1986, 1994), G statistic (Mingers, 1989), χ^2 (Mingers, 1989; White & Liu, 1994), Laplace (Quinlan, 1995), Gini Index (Breiman et al., 1984). Although this paper studies traditional evaluation metrics exclusively, a different kind of metric measure the ability of an expression to discriminate among different classes (Hong, 1997; Kira & Rendell, 1992; Kononenko & Hong, 1997).

This paper is a study of the bias adopted by traditional evaluation metrics. We limit ourselves to the case where expressions are binary-valued. Previous work has defined bias as the tendency of a metric to favor multi-valued expressions (White & Liu, 1994; Kononenko, 1995). Instead we follow the definition of bias as the preference for a partial ordering over the space of candidate

expressions. The question we address is not how much metric M favors multi-valued expressions, but rather how metrics M_1 and M_2 differ in their ranking of all available binary-valued expressions. Under this framework, our characterization shows the bias adopted by each metric explicitly, in a uniform representation, and enables us to compare differences in bias.

We propose a measure for the distance between the bias of two evaluation metrics. A pair-wise comparison of several traditional metrics using this distance-bias measure quantifies the degree of similarity (conversely dissimilarity) between metrics. We explore the connection between the distance-bias measure and differences in predictive accuracy when the same learning algorithm is built using different evaluation metrics. Multiple experiments on real-world domains show how the expectations on accuracy differences generated by the distance-bias measure correlate with actual differences when the learning algorithm is simple: look for the best single feature or the best single rule (Pearson's correlation coefficient $r \geq 0.97$). The usefulness of our proposed measure, however, is questionable under more complex algorithms, e.g., learning decision trees (Pearson's correlation coefficient $r \geq 0.52$). We conclude interactions among components in a learning algorithm must be taken into account to fully explain accuracy performance. A shorter version of this paper is provided by Vilalta and Oblinger (2000).

The organization of this paper follows. Section 2 reviews definitions of some traditional evaluation metrics. Section 3 gives a characterization of bias when the number of induced example subsets is binary, and defines a measure for the distance in bias between two evaluation metrics. In addition, we conduct a pair-wise comparison of the distance-bias of various evaluation metrics. Section 4 details on related work. Section 5 describes our experiments on real-world domains with discussion. Finally, Section 6 extends a summary and our conclusions.

2. DEFINITIONS OF SOME EVALUATION METRICS

Classification presupposes the availability of a training set T made up of examples, $T : \{(X_i, c_i)\}_{i=1}^{N}$, where a vector X is characterized as a point in a k-dimensional feature space. X_i is labeled with class c_i according to an unknown target function C, $C(X_i) = c_i$. For simplicity, a splitting function or expression E will be restricted to a logical form: E can represent a feature value or the logical combination of feature values; target concept C will also be assumed of logical form. Section 3.6 discusses how to extend our results when these restrictions are removed.

An evaluation metric M is used to quantify the quality of the partitions induced by an expression E over a training set[1] T, such that $|T| = N$. Expression E divides T in two sets: $\{X \mid E(X) = 1\}$ and $\{X \mid E(X) = 0\}$; we say the former set is covered by E, whereas the latter set is covered by the complement E'. Similarly, set T can be divided according to the coverage of concept C, and its complement C'. Figure 1 shows the cross-classification of classes and values of E. Let n^1 and n^0 be the number of examples in T of class 1 and 0 respectively, where $n^1 + n^0 = N$. Let n_1^1 and n_1^0 be the number of examples covered by E of class 1 and 0 respectively, such that $n_1^1 + n_1^0 = n_1$, and let n_0^1 and n_0^0 represent the corresponding numbers in E', such that $n_0^1 + n_0^0 = n_0$. In addition, Figure 1 defines probabilities as estimated from the data.

The following are definitions of some traditional evaluation metrics, where the goal is to maximize the output value.

[1] Alternatively T may refer to a subset of the training set. For example, T may be a subset of examples covered by a node of a decision tree other than the root node.

(a)

	C	C'	
E	n_1^1	n_1^0	n_1
E'	n_0^1	n_0^0	n_0
	n^1	n^0	N

(b) Probabilities:

For C and C': $P^1 = \frac{n^1}{N}$ $P^0 = \frac{n^0}{N}$

For E: $P_1^1 = \frac{n_1^1}{n_1}$ $P_1^0 = \frac{n_1^0}{n_1}$ $P_1 = \frac{n_1}{N}$

For E': $P_0^1 = \frac{n_0^1}{n_0}$ $P_0^0 = \frac{n_0^0}{n_0}$ $P_0 = \frac{n_0}{N}$

FIGURE 1: (a) cross-classification of expression values and classes; (b) probabilities estimated from the data.

Information Gain

Let entropy $\mathrm{H}(x, y) = -x \log_2(x) - y \log_2(y)$

$$\mathrm{IG}(E) = \mathrm{H}(P^1, P^0) - \sum_{i=0}^{1} (P_i \ \mathrm{H}(P_i^1, P_i^0)) \tag{1}$$

Gain Ratio

$$\mathrm{GR}(E) = \frac{\mathrm{IG}(E)}{\mathrm{H}(P_0, P_1)} \tag{2}$$

G Statistic

$$\mathrm{G}(E) = 2N \ \mathrm{IG}(E) \ \ln 2 \tag{3}$$

Gini

Let $\mathrm{GI}(x, y) = 1 - (x^2 + y^2)$

$$\mathrm{gini}(E) = \mathrm{GI}(P^1, P^0) - \sum_{i=0}^{1} (P_i \ \mathrm{GI}(P_i^1, P_i^0)) \tag{4}$$

χ^2

$$\chi^2(E) = \frac{N \ (n_1^1 n_0^0 - n_0^1 n_1^0)^2}{n^1 n^0 n_1 n_0} \tag{5}$$

Laplace

$$\mathrm{L}(E) = \begin{cases} \frac{n_1^1 + 1}{n_1^1 + n_1^0 + 2} & \text{if } n_1^1 \geq n_1^0 \\ \frac{n_1^0 + 1}{n_1^1 + n_1^0 + 2} & \text{if } n_1^1 < n_1^0 \end{cases} \tag{6}$$

Observations

- Function Gini is defined as the gain obtained by comparing the average impurity on each induced subset to the impurity of the whole set (similar to the definition of information gain).
- In 2×2 contingency tables, the sampling distribution of the χ^2 statistic is approximated by the χ^2 distribution with one degree of freedom (Upton, 1982).

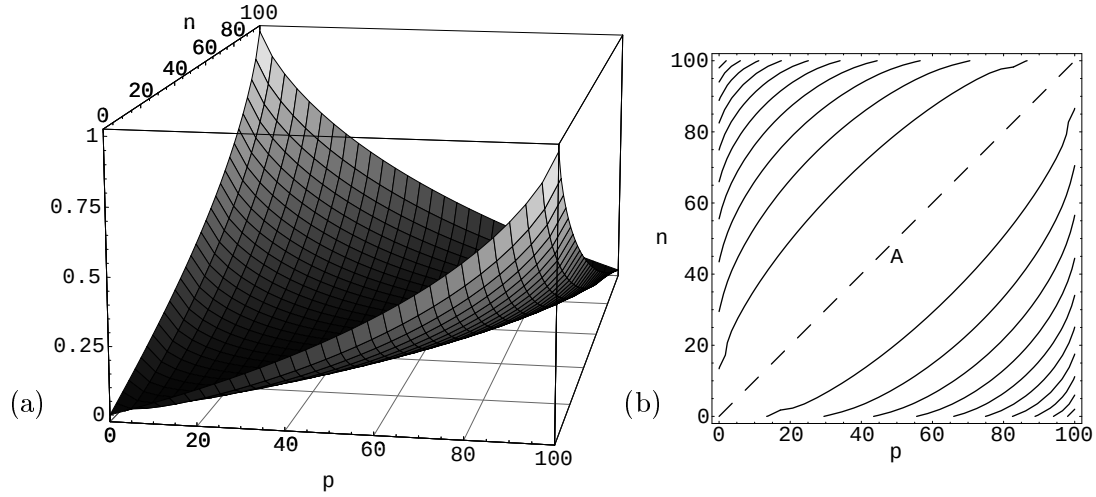

FIGURE 2: (a) Information Gain as a function of the possible coverage (number of positive and negative examples) of an expression. (b) A contour plot of (a) showing isometric lines over the coverage plane.

- The Laplace function is based on the information contained within the coverage of expression E, ignoring the information contained within the complement E'.

3. BIAS IN TRADITIONAL EVALUATION METRICS

This section extends a characterization of the bias of an evaluation metric for binary-valued expressions. The characterization will be helpful for comparison purposes. The most common approach to compare two different metrics, M_1 and M_2, is by observing the generalization accuracy obtained when the same learning model (e.g., a decision tree) is induced using M_1 first, and M_2 next. This approach ignores interactions between the evaluation metric and the inductive mechanism. A better approach is to isolate and study the bias of different evaluation metrics, independently of the learning algorithm in use.

3.1. The Coverage Plane

The metrics defined in Section 2 share a common characteristic: the value of an expression E is based on the intra-class purity of the induced example subsets. An evaluation metric M is a function of the number of positive and negative examples covered by expression E and of its complement E', $M : f(n_1^1, n_1^0, n_0^1, n_0^0)$ (Figure 1). Alternatively M could be defined as a function of the coverage of E and of the coverage of the whole set T, $f(n_1^1, n_1^0, n^1, n^0)$, since $n^1 = n_0^1 + n_1^1$ and $n^0 = n_0^0 + n_1^0$. For a given learning problem, n^1 and n^0 are fixed; by considering them as constants, we can simply express M as $f(n_1^1, n_1^0)$. For simplicity let's rename n_1^1 and n_1^0 as p and n (the positive and negative examples covered by E), such that $M : f(p, n)$. Metric M extends above the plane defined by these two variables.

Definition 1. Given an example set T, the **coverage plane** is a two-dimensional space where each point (p, n) represents the possible coverage (number of positive an negative examples) of an expression E in T.

Each of the metrics defined on Section 2 can be plotted above a coverage plane bounded by the total positive and negative examples in T, n^1 and n^0. As an example, Figure 2(a) plots

Information Gain when the number of positive and negative examples in T is the same ($n^1 = n^0 = 100$); each point (p, n) is evaluated according to equation 1. The fact that the value of $f(p, n)$ takes into account both the coverage of E and of its complement E' is reflected by the symmetry about the axis-line $((0, 0), (100, 100))$. The maximum values are attained at the extreme points $(100, 0)$ and $(0, 100)$, when the induced example subsets are class uniform.

Under the framework adopted by traditional evaluation metrics, the coverage plane is a low-dimensional representation of the space of all possible expressions $\mathcal{E}$. Two expressions covering different examples, but with the same contingency table as in Figure 2(a), would collapse into the same coordinate-pair. Hence, the bias of a traditional metric M specifies a partial ordering over all points in the coverage plane (e.g., prefer (30,0) to (40,5)). Such ordering is necessary to select the best instance(s) of $\mathcal{E}$ during learning.

Observe that although the coverage plane provides us with a uniform representation, it hides the role of the size of the training set from our definitions. For example, the values of the G and χ^2 statistics (Equations 3 and 5 respectively) depend directly on the size of the training set N. In contrast, information gain (Equation 1) depends only on relative class proportions. Since in our study N is considered fixed for each domain, we overlook this distinction[2].

3.2. Bias and Isometric Lines

We now answer the following question: how can we concisely represent the ordering imposed by an evaluation metric over all points in the coverage plane? (i.e., over all possible expressions?). To answer this question, consider first the result of projecting $M : f(p, n)$ over the coverage plane. Those points that have the same value for $f(p, n)$ project into the plane as isometric lines.

Definition 2. An isometric line indicates a constant value for $M : f(p, n)$ throughout its extent on the plane $p \times n$; the points along an isometric line, $(p_1, n_1), (p_2, n_2), \cdots, (p_s, n_s)$, are equally rewarded by M, i.e., $f(p_1, n_1) = f(p_2, n_2) = \cdots = f(p_s, n_s)$.

As an example, Figure 2(b) shows isometric lines obtained by projecting Information Gain over the coverage plane. Equivalent plots for all functions defined in Section 2 are shown in Appendix A. An isometric line comprises points having the same height on $f(p, n)$; the line actually joins such points, but has no meaning between non-integer values of p and n. The isometric line with lowest-value points, axis-line A, cuts the space in two, from $(0, 0)$ to (n^1, n^0). All other lines move parallel to A towards the extreme points $(n^1, 0)$ and $(0, n^0)$, where $n^1 = n^0 = 100$.

Consider arbitrarily the right lower half of the coverage plane shown in Figure 2(b). As long as M monotonically increases from axis-line A to the extreme point $(n^1, 0)$, then any isometric line divides this half space into two regions; the right lower region encompasses points with a preference over those points in the left upper region [3]. *Thus, the bias of an evaluation metric is determined by the shape of the isometric lines (i.e., contour lines) obtained by projecting $M : f(p, n)$ over the coverage plane.* This can be formalized in the following way:

Observation. Let $M_1 : f_1(p, n)$ and $M_2 : f_2(p, n)$ be two different evaluation metrics (i.e., there is at least one point (p, n) where $f_1(p, n) \neq f_2(p, n)$), and let both metrics be symmetrical about the lowest-valued isometric line A, going from $(0, 0)$ to (n^1, n^0). Assume both M_1 and M_2 increase

[2] Evaluation metrics that depend directly on N can be approximated by a sample distribution that is not parameterized by sample size (e.g., the X^2 distribution), which allows for measures of significance.

[3] Symmetrically, an isometric line on the left upper half of the coverage plane divides this half space into two regions; the left upper region having a preference over the right lower region.

monotonically from A to the extreme points $(n^1, 0)$ and $(0, n^0)$. If both M_1 and M_2 project the same isometric lines over the coverage plane, then no difference exists in their inductive bias.

An example of two metrics with same bias is Information Gain and the G statistic (Section 2). To prove that both metrics share the same bias, note that the G statistic is simply a constant factor times Information Gain, and thus both metrics project the same isometrics lines over the coverage plane. In general, let $M_1 : f_1(p, n)$ and $M_2 : f_2(p, n)$ be two different evaluation metrics, and let $f_2(p, n) = cf_1(p, n)$, where c is a constant factor. For any two points $R_1 = (p_1, n_1)$ and $R_2 = (p_2, n_2)$, it follows that if $f_1(R_1) \oplus_1 f_1(R_2)$ is true, then $cf_1(R_1) \oplus_1 cf_1(R_2)$ must also be true, where $\oplus_1 \in \{=, >, <\}$. Thus M_1 and M_2 have the same bias whenever M_2 is simply M_1 times a constant factor. It can be seen how the same argument must hold for any monotonic transformation of f_1.

3.3. The Distance Between The Bias of Two Evaluation Metrics

We now proceed to explain a method to quantify the distance between the bias of two evaluation metrics. To begin, let us first find a representation for a single isometric line. Let $M : f(p, n)$ be an evaluation metric and let $R_0 = (p_0, n_0)$ be a fixed point on the coverage plane. We can represent the isometric line passing through R_0 by a function that maps values of n (arbitrarily chosen as the dependent variable) onto values of p, the independent variable. We use the function, $\mathrm{C}_{M,R_0}(p)$, to denote the contour line of the metric M that passes through the point R_0. For example, for all points on the right lower half of the coverage plane (i.e., when $p > n$), the Laplace function (Section 2) is defined as follows:

$$\mathrm{L}(E) = \mathrm{L}(p, n) = \frac{p+1}{p+n+2} \tag{7}$$

For a fixed point $R_0 = \langle p_0, n_0 \rangle$, we solve this equation for n and generate the contour function for this metric:

$$\mathrm{C}_{L,R_0}(p) = n = \frac{p+1}{\mathrm{L}(\mathrm{p_0}, \mathrm{n_0})} - (p+2) \tag{8}$$

In order to ensure that this definition is valid over the entire coverage plane we take n to be 0 if the contour function returns a negative number. More precisely, n must be represented as $n : (g(p))_+$, where $(\cdot)_+$ denotes the positive threshold function, i.e., $(g(p))_+ = 0$ if $g(p) < 0$, and $(g(p))_+ = g(p)$ if $g(p) \geq 0$. Equation 8 represents a family of isometric lines for the Laplace function according to the value of R_0.

We now consider a definition for the distance between the bias of two evaluation metrics. Given a fixed point R_0 and two metrics M_1 and M_2 we define the distance between these two metrics as the area between the isometric lines from each metric passing through the point R_0. Formally the area between the two isometric lines at R_0 is written as $\delta_{R_0}(M_1, M_2)$, and it is defined as:

$$\delta_{R_0}(M_1, M_2) = \int_0^{n^1} |\mathrm{C}_{M_1,R_0}(p) - \mathrm{C}_{M_2,R_0}(p)| \; dp \tag{9}$$

The integral goes from 0 to the total number of positive examples n^1, and captures the disagreement between M_1 and M_2. As an example, Figure 3(a) shows the projection of a single isometric line for two metrics. R_0 corresponds to the point of intersection between the lines. The shaded area between both lines quantifies the difference in bias between M_1 and M_2 at R_0. Equation 9 can be interpreted in the following way: Assume E is an expression whose coverage

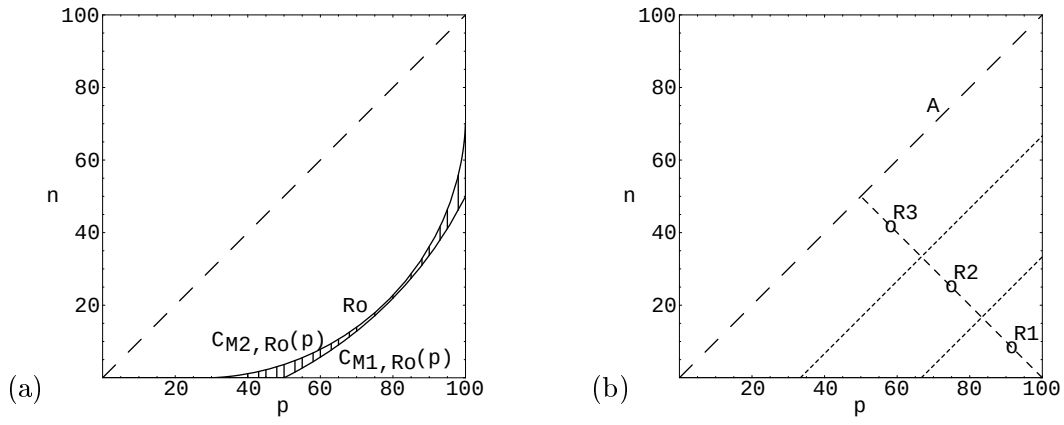

FIGURE 3: (a) The difference in bias between metrics M_1 and M_2 represented as the area between $n : g_1(p)$ and $n : g_2(p)$. (b) Three points positioned at different regions of the coverage plane.

is characterized by the coordinate R_0. Now for another expression E', the metrics will disagree on the ordering of E and E' only if the coverage coordinates of E' fall within the area between the isometric curves for the two metrics. Thus Equation 9 is a measure of dissent between the two metrics at a single point R_0.

We can naturally extend equation 9 to a general measure of dissimilarity between two metrics. This is simply the average size of the dissent region over the space of all coverage coordinates. The overall measure of dissimilarity is defined as:

$$\delta(M_1, M_2) = \frac{1}{|\mathrm{R}|} \sum_{r \in \mathrm{R}} \delta_r(M_1, M_2) \tag{10}$$

where R denotes the set of all points r in the coverage plane, $|R| = pn$. Because a numeric approximation of equation 10 is computationally expensive, in what follows we focus our attention on equation 9 exclusively.

3.4. A Quantitative Comparison

The definition for the distance between the bias of two evaluation metrics (Equation 9) varies depending on R_0. This is an important issue for analysis. The difference in bias depends on the position of the expression E under evaluation. An expression near the extreme points $(n^1, 0)$ or $(0, n^0)$ is expected to have high preference over all other points closer to axis-line A, and we expect this to be the same among most metrics. In the extreme case where the expression lies right on one of these two extreme points (i.e., the expression induces class-uniform subsets) then $\delta_{R_0}(M_1, M_2) = 0$ for every pair of metrics M_1 and M_2. But as we move towards axis-line A we expect the disagreement among different metrics to be more evident.

To test the effect of the position of R_0 in the evaluation of $\delta_{R_0}(M_1, M_2)$, we conducted a series of experiments described as follows. Figure 3(b) shows a coverage plane with equal class proportions ($n^1 = n^0 = 100$); the right-lower half of the plane is divided into three equally-spaced regions. We took three different points along the line perpendicular to A ending on the extreme point $(n^1, 0)$ (i.e., $(100, 0)$). These points are labeled as R_1, R_2, and R_3. We carried out a pair-wise comparison of all evaluation metrics defined in Section 2 by applying equation 9 at each of these three points. The goal of this experiment is to determine which metrics are more similar to each other, and what is the effect of evaluating an expression on different regions of the

TABLE 1: A pair-wise comparison of evaluation metrics.

Pair of Metrics	R_1 $(92, 8)$	R_2 $(75, 25)$	R_3 $(58, 42)$
Gain-Ratio vs Gini/χ^2	20.23 0.4%	233.8 4.6%	152.62 3.0%
Gain-Ratio vs Info-Gain/G	6.76 0.13%	178.8 3.5%	145.42 2.9%
Gain-Ratio vs Laplace	327.5 6.5%	806.5 16%	573.6 11.4%
Gini/χ^2 vs Info-Gain/G	13.46 0.27%	55.02 1.1%	7.20 0.14%
Gini/χ^2 vs Laplace	327.5 6.5%	806.5 16%	567.3 11.3%
Info-Gain/G vs Laplace	327.5 6.5%	806.5 16%	567.3 11.3%
Info-Gain vs G Gini vs χ^2	0.0 0.0%	0.0 0.0%	0.0 0.0%

coverage plane. On each comparison we used Mathematica (Wolfram, 1999) to obtain a numeric approximation of equation 9. The results are shown in Table 1. The coordinates for R_1, R_2, and R_3 are indicated on the first row. On each cell, the number on the top represents the result of evaluating equation 9; the number on the bottom shows the fraction of the area covered by the top number with respect to the area of the entire half space of the coverage plane. The area of the half space in Figure 3(b) equals $(100 \times 100)/2 = 5000$, and in general the area equals $(n^1 \times n^0)/2$.

An analysis of Table 1 shows no difference between Information Gain and the G statistic. The two metrics share the same bias since the G statistics is simply a constant factor times information gain (Section 3.2). In addition, our results show no difference between χ^2 and Gini. We leave as an open problem a proof (or disproof) that both Gini and χ^2 project the same isometric lines. Using the bottom value on each table entry, the smallest difference on average is between Gini (or χ^2) and Information Gain (or the G statistic), around 0.5%. Next in magnitude is the difference between Gain Ratio and Information Gain (or the G statistic), around 2.18%, followed by the difference between Gain Ratio and Gini (or χ^2), around 2.7%. Most distances are within 5%, except for the Laplace function showing differences up to 16%.

Notice in Table 1 that the difference between any pair of metrics increases from point R_1 to point R_2, and then decreases from R_2 to R_3. An explanation for such general behavior can be related to the ceiling and floor effects of experimental design (Cohen, 1995, Chapter 3, p. 79). A ceiling effect occurs at points around R_1 because the quality of the expressions in this region is close to optimal, leaving few room for improvement. Conversely, a floor effect occurs at points around R_3 because the proximity to axis-line A forces most metrics to agree in assigning low credit to any expression in this region. The highest difference among biases is expected along the region comprising R_2, where no upper or lower bounds restrain the degree of agreement —or disagreement— in bias[4]. Accordingly, one must be careful in attending the position of the expression under evaluation when comparing different evaluation metrics, and to relate such position to the nature of the domain under study. Learning in simple domains tends to produce expressions around point R_1; concluding that two metrics are similar is affected by the ceiling effect. A similar invalid conclusion would be derived by comparing two metrics on difficult (possibly random) domains where the induced expression lies close to axis-line A (Figure 3(b)).

[4]The location of the three points was selected to show this effect more clearly.

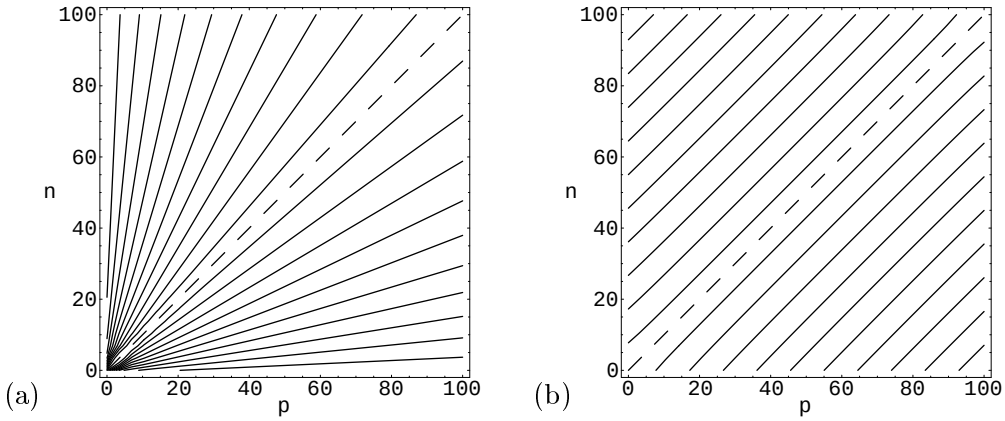

FIGURE 4: (a) A contour plot for the Laplace function (b) A contour plot for the alternate definition of the Laplace function in which both an expression E and its complement E' are taken into account.

The reason for the Laplace function yielding high differences in bias stems from the definition provided in Section 2, equation 6, that measures the (adjusted) accuracy of the coverage of an expression E independent of its complement E'. To further understand this fact, let us first define two families of functions within the set of traditional evaluation metrics: *complete evaluators*, characterized by evaluating the quality of the coverage of an expression E and of its complement E' (i.e., by looking into all numbers in the contingency table of Figure 1), and *partial evaluators*, characterized by evaluating the quality of the coverage of E alone, ignoring the complement E'. The results in Table 1 can be explained by noticing that the Laplace function belongs to the second kind of family while all others functions belong to the first family. We could consider a version of the Laplace function L_2 that belongs to the first kind of family (i.e., complete evaluators) as follows:

Laplace 2

$$\mathrm{L}_2(E) = \sum_{i=0}^{1}(P_i \times \mathrm{L}(P_i^1, P_i^0)) \tag{11}$$

The definition for L_2 is simply a weighted average of the original Laplace function on both E and E'. Figures 4(a) and (b) show isometric lines obtained by projecting both the original and alternate definition for the Laplace function over the coverage plane. The original function L pays little attention to the size of the coverage of E as long as class purity within E is guaranteed; L_2 instead balances coverage size and class purity, since both E and E' are part of the equation. Table 2 compares the two versions of the Laplace function against Information Gain. The distance between L_2 and Information Gain is similar to the distance between pairs of metrics in Table 1 (excluding occurrences with the original Laplace function). L_2, however, appears different than other metrics close to axis-line A, where we observe an increase in distance-bias from R_2 to R_3. Overall we can conclude that to explain differences in performance it is important to identify the family of functions to which an evaluation metric belongs. We expect the distance in bias between two metrics to be small if both metrics belong to the same family.

3.5. The Case of Unequal Class Proportions

Results on Table 1 are based on problem domains with equal class proportions. But, what

TABLE 2: A comparison of the two versions of the Laplace function against Information Gain.

Pair of Metrics	R_1 $(92, 8)$	R_2 $(75, 25)$	R_3 $(58, 42)$
Info-Gain vs Laplace	327.5 6.5%	806.5 16%	567.3 11.3%
Info-Gain vs Laplace2	20.12 0.4%	200.5 4.0%	238.04 4.7%

happens if class proportions are unequal? To answer this question we analyze the effect of skewed class distributions over the coverage plane. We have learned that the difference in bias between two metrics is expected to be highest when the coverage of the expression under evaluation lies half-way between one of the extreme points, $(n^1, 0)$ or $(0, n^0)$, and axis-line A; it is in this half-way region where strong disagreements on how to rank an expression may take place (point R_2, Figure 3(b)). A domain with unequal class proportions maps to a coverage plane that stretches into a wide or tall rectangle, depending on the dominating class. In either case, the result is a reduction in the size of this half-way region. More formally, consider the line perpendicular to axis line A connecting A with any of the extreme points $(n^1, 0)$ and $(0, n^0)$ (Figure 3(b)). This line attains maximum size when $n^1 = n^0$, decreasing as the population of one class grows large relative to the other class. The shorter the size of this line the more difficult to distinguish differences in bias because most expressions would lie close to either line A or one of the uniform-class points. Therefore, the degree of skewness in the class distribution is an important factor when comparing the effect of different evaluation metrics during learning. A highly skewed distribution may lead to the conclusion that two metrics yield similar generalization effects, when in fact a significant difference could be detected under equal class distributions.

3.6. The Case of Multi-Class Problems

An interesting problem is to extend the study of Section 3.3 to multi-class problems. We proceed to do this analytically, deferring the actual numeric computation of the distance measure for future work. We begin by expanding our notation. Let $n^0, n^1, \cdots, n^k$ be the number of examples of class $0, 1, \cdots, k$ in set T respectively. Again, because $n^0, n^1, \cdots, n^k$ are fixed in T, an evaluation metric M can be represented as a function of the number of examples of each class covered by an expression E, $M : f(n_1^0, n_1^1, \cdots, n_1^k)$. This assumes E is a logical expression inducing exactly two example subsets[5], because the coverage of the complement E' can be deduced from the coverage of E and T. Since the coverage of a logical expression E is a point in a k dimensional space, the space of all these points is now referred to as the coverage space. Function f can be projected over the k-dimensional coverage space to facilitate identification of isometric hyperplanes, i.e., hyperplanes where each point is rewarded equally by f. The hyperplane passing through a point R_0 can be represented by arbitrarily taking n_1^k as the dependent variable, and generating the contour function for this metric:

$$\mathrm{C}_{M,R_0}(n_1^0, n_1^1, \cdots, n_1^{k-1}) = n_1^k = g(n_1^0, n_1^1, \cdots, n_1^{k-1}) \tag{12}$$

For a fixed point R_0 and two metrics $M_1 : f_1(n_1^0, n_1^1, \cdots, n_1^k)$ and $M_2 : f_2(n_1^0, n_1^1, \cdots, n_1^k)$, let the isometric hyperplanes corresponding to each metric passing through R_0 be defined (respectively)

[5]Our results are limited to binary expressions. They cannot be extended to multi-valued expressions where the coverage of a single expression-value is insufficient to deduce the coverage of the complement, i.e, to deduce the coverage of all other values in that expression.

as $n_1^k : g_1(n_1^0, n_1^1, \cdots, n_1^{k-1})$ and $n_1^k : g_2(n_1^0, n_1^1, \cdots, n_1^{k-1})$. The distance between the two metrics, $\delta_{R_0}(M_1, M_2)$, is the region between g_1 and g_2:

$$\int_0^{n^0} \int_0^{n^1} \cdots \int_0^{n^{k-1}} |\mathrm{C}_{M_1,R_0}(n_1^0, n_1^1, \cdots, n_1^{k-1}) - \mathrm{C}_{M_2,R_0}(n_1^0, n_1^1, \cdots, n_1^{k-1})| \; d_{n_1^{k-1}} d_{n_1^{k-2}} \cdots d_{n_1^0} \quad (13)$$

Equation 12 extends equation 9 to capture the disagreement between M_1 and M_2 at point R_0 in problem domains involving k classes.

4. RELATED WORK

Early work comparing evaluation metrics for classification is reported by Mingers (1989). Mingers compares various metrics (the G statistic, χ^2, Gain Ratio, Gini, and random selection) in the context of decision tree induction. His conclusion that features selected at random produce trees that perform similar to trees with features selected using standard evaluation metrics is refuted by Liu and White (1994).

Other studies concerning the bias of an evaluation metric focus on the tendency to favor multi-valued features. White and Liu (1994) show how metrics using the χ^2 distribution (e.g., χ^2 test of goodness of fit, and the G statistic) are to be preferred because they automatically adjust for the number of feature values. Their study, however, included only random features. A more recent study by Kononenko (1995) investigates the work of White and Liu (1994) concluding that χ^2 and the G statistic are also biased towards informative multi-valued features. Overall Kononenko advocates RELIEF (Kira & Rendell, 1992; Kononenko, 1994) as a more promising measure.

Another line of research is centered on the properties of evaluation metrics that guarantee optimal multi-partitions of numeric features. Elomaa and Rousu (1999) characterize metrics to determine their usefulness when splitting numeric features. Codrington and Brodley (1999) study the convexity properties of many evaluation metrics. Morishita (1998) focuses on the convexity property of Entropy.

Our study relates to the bias associated with evaluation metrics when the number of induced partitions is binary. Related work is found by Breiman (1996) who investigates the difference between Entropy and Gini under binary partitions on multi-class problems. Breiman shows how the Gini metric tends to place the largest class on one side of the partition (to produce a pure node), and all other examples on the other side. In contrast, Entropy tends to balance the size of the two sides of the partition.

An analysis of the effects of the concavity of an evaluation metric during decision-tree learning is provided by Kearns and Mansour (1996) and Dietterich, Kearns, and Mansour (1996). Although both papers aim at fitting decision-tree learners within the framework of boosting, part of their study investigates properties of splitting functions. Their analysis uses a closely similar representation to ours when plotting evaluation metrics, where the X-axis stands for the proportion of positive examples, and the Y-axis stands for the value of the impurity measure (see papers for details). They make similar observations to us by noticing that the lowest gains are achieved in the extreme sides of the X-axis (corresponding to the extreme optimal points in the coverage plane, Figure 3b), or when the class proportions in the two induced example subsets are very similar (corresponding to axis-line A in Figure 3b).

5. EXPERIMENTS

We now turn to a series of experiments intended to measure the effects generated by selecting different evaluation metrics during learning. If two metrics, M_1 and M_2, are used to generate hypotheses h_1 and h_2 respectively, we expect the disagreement between h_1 and h_2 to increase as the distance in bias $\delta(M_1, M_2)$ increases too. Of particular interest is then to compare differences in predictive accuracy resulting from using different evaluation metrics. We wish to determine if the distance measure developed in Section 3.3 (equation 9) correlates with such performance differences. A positive correlation would improve our understanding of what makes a learning algorithm perform better (or worse) than other algorithms according to the domain of study.

We select a simple learning model that conducts a search over the space of logical expressions, each expression represented as the conjunction of feature literals (feature values or their negations). The final hypothesis is a rule of the form: **if** (f_1 & f_2 & $\cdots$ & f_p) **then** Class = c, where f_i is a boolean feature or its negation, and c is the class assigned to any example satisfying the rule antecedent. The model is justified from the fact that we wish to obtain hypotheses not always consistent with the training data (expected to underfit the target concept), and thus not necessarily lying on an optimal, class-uniform point over the coverage plane (Section 3.1). This will allow us to test the distance measure (equation 9) on different regions of the coverage plane.

The learning algorithm that outputs a rule as the final hypothesis explores the space of rule antecedents by conducting a beam search over the space of expressions represented as conjunctions or monomials. It proceeds by conjoining one literal (i.e., boolean feature or its complement) at a time to the best previously retained expressions in the beam (the beam starts with the best single literals). Adding literals extends the depth of the search; retaining the best expressions limits the width of the search. The process continues while keeping track of the best global expression according to the evaluation metric in use. The search above terminates when no more expressions are left on the beam, at which point the best expression E is output. The majority class c of E is taken as the class in the final rule hypothesis: **if** E **then** Class = c. The algorithm is detailed in Appendix B.

5.1. Experimental Methodology

Table 3 compares the performance of the learning algorithm described above when the search for the best expression is conducted using different evaluation metrics. The table reports on four different algorithm versions (columns 4-7). Information Gain and the G statistic, and Gini and χ^2, have each pair the same bias (i.e., the same generalization effect) and thus collapse into the same column.

The first column lists real-world domains extracted from the UCI database repository (Merz & Murphy, 1998), except for the star-cluster domain extracted from Table 22.1 in Andrews and Herzberg (1985).

The second column in Table 3 indicates, on average and for each domain, the location of the final expression over the coverage plane. Region 1 is the closest to the class-uniform points, region 2 is half-way between the class-uniform points and axis-line A, and finally region 3 is closest to axis-line A (Figure 3b). The region to which an expression belongs is determined as explained in Section 3.4, by dividing each of the two half-spaces of the coverage plane into three equally spaced regions. The location of the expression on the coverage plane indicates the region to which it belongs. Each entry on column 2 is the result of first averaging for each algorithm version and over all runs the region on the coverage plane where the final expression lies. We

TABLE 3: Results on real-world domains. Numbers enclosed in parentheses represent standard deviations.

Concept	Region (1-3)	Proportion (+) class	Predictive Accuracy			
			Info. Gain G Statistic	Gini χ^2	Gain Ratio	Laplace
voting	1.0	0.38	95.0 (0.26)	95.13 (0.38)	94.95 (0.39)	92.53 (1.35)
cancer	1.01	0.65	94.16 (0.64)	94.64 (0.61)	94.51 (0.36)	90.34 (1.08)
new-thyroid-hyper	1.01	0.17	96.67 (1.33)	97.10 (1.05)	96.43 (0.75)	90.42 (1.79)
new-thyroid-hypo	1.0	0.15	96.19 (1.04)	96.10 (0.90)	95.62 (0.87)	94.19 (1.98)
star cluster	1.04	0.67	96.95 (0.71)	96.81 (0.80)	96.86 (0.83)	83.90 (2.68)
promoters	1.18	0.50	76.2 (4.07)	77.8 (2.71)	78.7 (3.82)	74.2 (4.02)
mushroom	1.25	0.45	99.73 (0.02)	99.53 (0.32)	99.73 (0.02)	78.39 (0.01)
ionosphere	1.25	0.66	89.63 (1.64)	89.74 (1.71)	90.31 (1.24)	66.06 (3.42)
crx	1.38	0.45	86.01 (0.34)	86.12 (0.26)	85.94 (0.32)	62.05 (2.39)
mean region 1			92.28	92.55	92.56	81.34
hepatitis	1.58	0.18	70.2 (2.93)	78.0 (1.90)	85.9 (2.34)	64.5 (3.11)
lymphography-2	1.6	0.52	81.71 (2.22)	81.93 (2.0)	80.21 (3.13)	72.29 (3.09)
lymphography-3	1.67	0.44	79.21 (3.03)	79.43 (2.85)	76.79 (1.87)	64.86 (4.54)
zoo	1.92	0.05	85.33 (2.27)	85.44 (3.20)	85.0 (3.55)	47.0 (10.27)
credit	1.95	0.67	67.36 (3.69)	69.81 (2.50)	73.09 (3.18)	54.90 (4.25)
chess-end game	2.01	0.50	79.41 (0.86)	80.73 (0.33)	75.44 (0.09)	72.80 (0.94)
heart	2.06	0.45	71.31 (1.03)	71.82 (1.35)	71.52 (1.27)	62.38 (4.77)
mean region 2			76.36	78.17	78.28	62.68
diabetes	2.51	0.32	67.62 (2.13)	70.86 (1.80)	69.89 (0.84)	43.32 (0.95)
bupa	2.61	0.45	59.18 (2.56)	60.63 (2.57)	57.72 (2.18)	47.12 (1.35)
tic-tac-toe	2.69	0.67	58.41 (2.37)	67.88 (4.15)	73.68 (0.88)	50.77 (1.97)
mean region 3			61.74	66.46	67.10	47.07
mean overall			77.16	78.94	79.30	65.02

then take the average over all algorithm versions. Domains in Table 3 are ordered based on the values in column 2. We assign a domain to region 1 if the entry on column 2 is within $[1.0, 1.5)$, region 2 if it is within $[1.5, 2.5)$, and region 3 if it is within $[2.5, 3.0]$.

The third column in Table 3 shows the proportion of positive examples on the training set. Each entry is found by averaging the proportion of positive classes on the training set over all runs (all algorithm versions are presented the same training data).

The last four columns in Table 3 estimate, for each algorithm version, the predictive accuracy achieved on each domain by using stratified 10-fold cross-validation (Kohavi, 1995), averaged over 10 repetitions. Numbers enclosed in parentheses represent standard deviations. Previous to each run, an initial discretization step makes all features boolean (numeric features are discretized following Catlett (1991); nominal features are decomposed into a boolean feature for each nominal value). On each run, the training set is divided in two: one half conducts a beam search for the best logical expression (as mentioned above), the other half serves to validate the best current expression to avoid statistical errors from multiple comparisons (Jensen & Cohen, 1999, 1997).

Runs were performed on a RISC/6000 IBM model 7043-140.

5.2. Testing The Utility Of The Distance Measure

We now test the utility of the distance-bias measure defined in equation 9. Our first experiment uses the last row in Table 3 corresponding to the average predictive accuracy for each

algorithm over all domains. We compute the absolute difference in predictive accuracy between each pair of algorithms. We also compute the average distance in bias between pairs of evaluation metrics in Table 1 (average over the three regions). The results can be paired-up by matching average distance in bias between metrics with the corresponding average accuracy difference (e.g., the average distance bias between Gini and Laplace maps to the absolute accuracy difference of the two algorithms using Gini and Laplace). A linear regression model applied to this data yields a correlation coefficient (Pearson's coefficient) of $r = 0.97$, which points to a strong variable interdependence between the distance-bias measure and differences in accuracy performance. Results for other similar experiments are all summarized on Table 4. Each entry shows the correlation coefficient obtained from fitting a linear regression model to the data. Results are grouped in two columns. The first column uses all available domains. The second column eliminates the effect of skewed class distributions by filtering out domains with a positive-class proportion outside the range $[0.4, 0.6]$ (Section 3.5), leaving a total of eight (out of nineteen) domains. Without skewed distributions the experiment described above produces a correlation coefficient of $r = 0.99$. Such improvement indicates that the relation between the distance-bias measure and differences in predictive accuracy is more evident when the computation of equation 9 is done assuming a class distribution similar to that of the domain under analysis (Table 1 assumes equal class proportions). In addition, Table 4 shows results of experiments similar to the two above, except we group domains on three regions according to the coverage of the expression in the final rule hypothesis. The highest correlation is observed when domains are grouped into regions and skewed distributions are eliminated (bottom entries on column 3, Table 4).

5.3. Using Different Models

Our analysis is so far limited to single-rule hypotheses. One might ask if the results hold for other learning algorithms. We address the following question: does the relation between the distance-bias measure and differences in accuracy hold as the complexity of the algorithm increases? To answer this question we report on two additional experiments. The first experiment uses a simple algorithm that outputs a single-feature as the final hypothesis. After repeating the experiments reported above, correlation coefficients comparable to the first row of results on Table 4 (average over all domains with and without eliminating skewed distributions) take values of $r = 0.98$ and $r = 0.98$ respectively.

The second experiment increases the complexity of the algorithm by using a decision tree as the hypothesis. Our implementation uses an initial discretization step on all numeric features by dividing each feature domain into ten equally-sized intervals. We stop growing a tree if the number of examples on a node is less than 3 or if all examples are class uniform. The final tree is pruned using a pessimistic-pruning method (Quinlan, 1994). Repeating the experiments above, the correlation coefficients are $r = 0.52$ and $r = 0.72$.

Our results show how increasing the number of learning components in the algorithm weakens the correlation between the distance-bias measure and differences in predictive accuracy. Thus, even if the evaluation metric is the only component altered in the learning algorithm, failing to understand interactions among all other components may result in a poor understanding of performance. If two different evaluation metrics are used to grow two decision trees, the same pruning mechanism may exert different changes on the trees. Thus, a robust analysis would additionally need to consider the effects of interacting components to account for differences in performance, such as tree pruning, a continuous partitioning of the feature space, the tree-stopping criterion, etc.

TABLE 4: Correlation coefficients on data comparing the distance-bias measure vs. differences in accuracy for a single-rule hypothesis.

Experiment	Correlation Coefficient	
	All Domains	Domains Without Skewed Dist.
Overall Average	0.97	0.99
Average Region 1	0.79	1.00
Average Region 2	0.97	0.98
Average Region 3	0.92	0.98

6. SUMMARY AND CONCLUSIONS

This paper provides a characterization of bias for traditional or purity-based evaluation metrics. We show how the projection of an evaluation metric M over the coverage plane (i.e., the plane where axis i represents the number of examples of class i covered by an expression) yields isometric lines, or lines of constant value. It is the shape of these isometric lines that indicates the preference for one expression over another, i.e., indicates the bias of M (Section 3.2). The characterization above leads naturally to a measure for the distance in bias between two evaluation metrics (Section 3.3).

The distance measure simply accounts for the area between two intersecting (isometric) lines, each line corresponding to one of the evaluation metrics. The distance measure is dependent on the position of the intersecting point between the two lines over the coverage plane. The closer this point to either a class-uniform point or a maximally impure point, the shorter the distance in bias. The highest disagreement between two evaluation metrics, i.e., the largest value for the distance measure, is expected half-way between the extreme points above (Section 3.4).

Our experimental results show a correlation between the distance-bias measure and differences in predictive accuracy when the same learning model is built using different evaluation metrics (e.g., look for the best single feature or the best single rule). Our results also show how the correlation tends to weaken as the degree of interaction between the evaluation-metric component and other components embedded in the learning algorithm increases (e.g., learning decision trees, Section 5.3). From this we conclude that a key element to understand differences in performance is to take into account interactions among learning components.

Future work will extend our results by trying to characterize the distance in bias between learning models apparently too far apart in their design. For example, how can we explain the different (or similar) behavior of a decision tree and a neural network on a particular domain? How about bayesian and kernel estimators? We gather from this study that an initial step to understand differences in performance is to produce an explicit representation of model bias, i.e., of the partial ordering imposed over the space of hypotheses. Such representation, can then serve to quantify the amount of agreement (or disagreement) between the bias of two different models. In addition, we plan to investigate the reasons why an algorithm outperforms others. In Table 4 we ignore why Gain Ratio, followed by Gini and χ^2 yield on average better predictive accuracy than Information Gain, the G statistic, and the Laplace function.

ACKNOWLEDGMENTS

This paper got benefited from many valuable suggestions provided by Se June Hong, Sholom Weiss, and Chid Apte. This work was supported in part by IBM T.J. Watson Research Center (USA).

APPENDIX A. PLOTS FOR EVALUATION METRICS

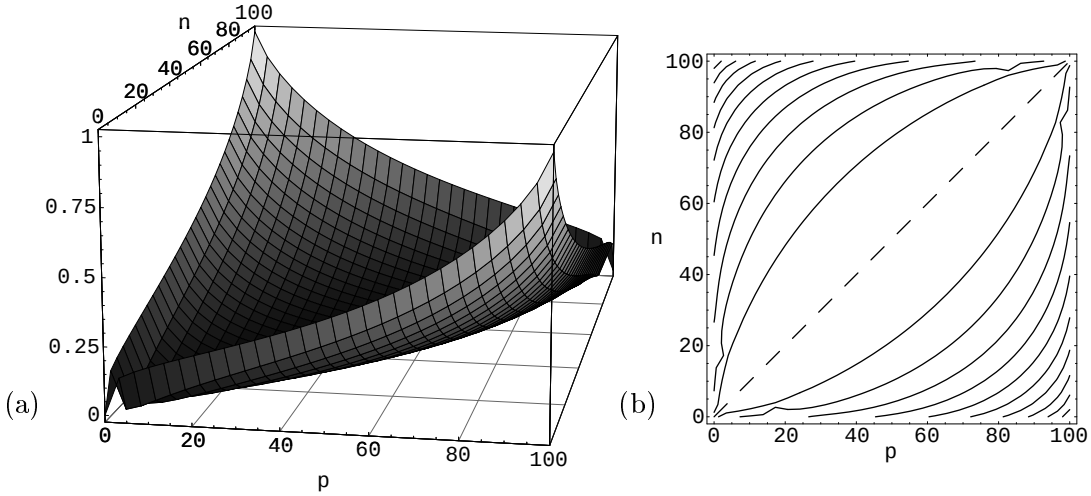

FIGURE 5: (a) Gain Ratio function. (b) A contour plot of (a).

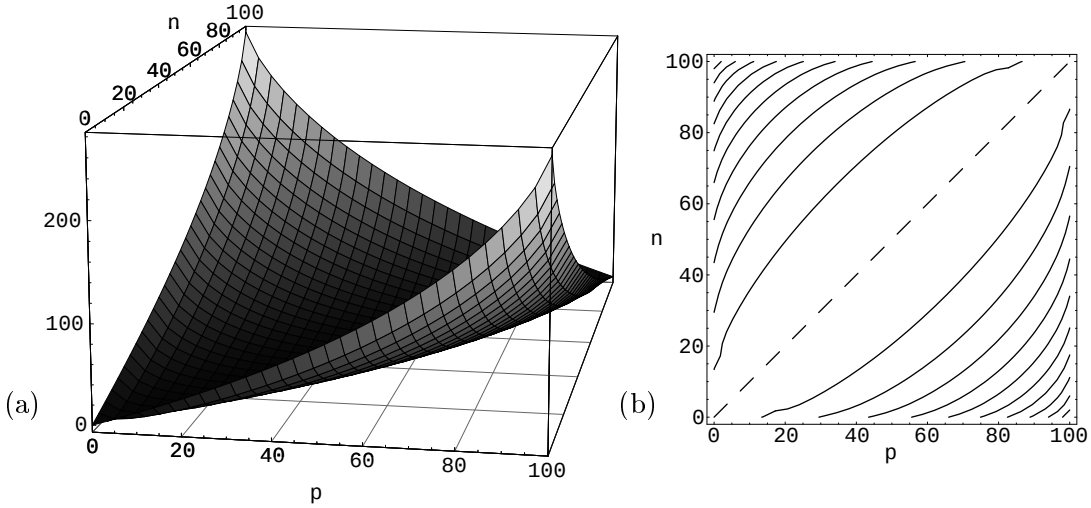

FIGURE 6: (a) G function. (b) A contour plot of (a).

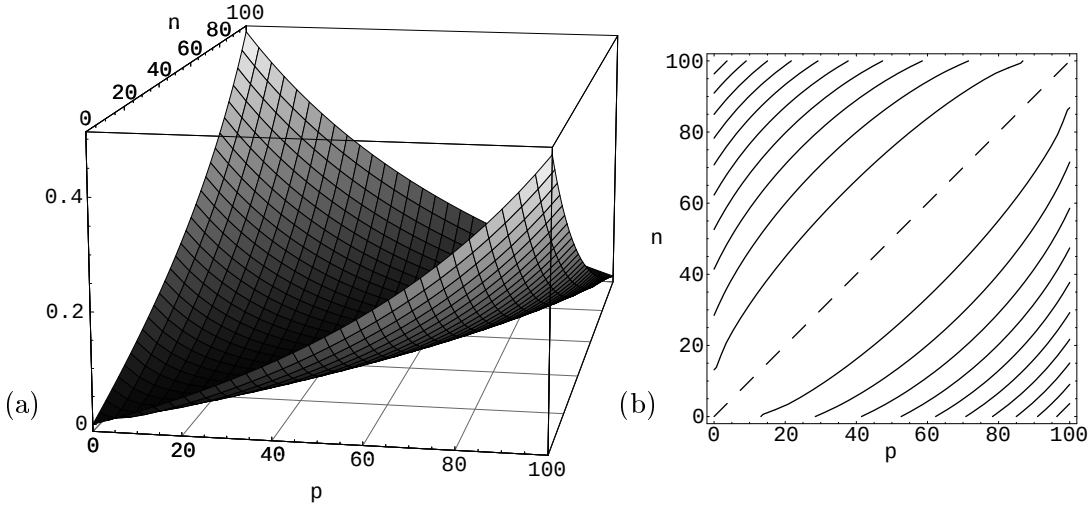

FIGURE 7: (a) Gini function. (b) A contour plot of (a).

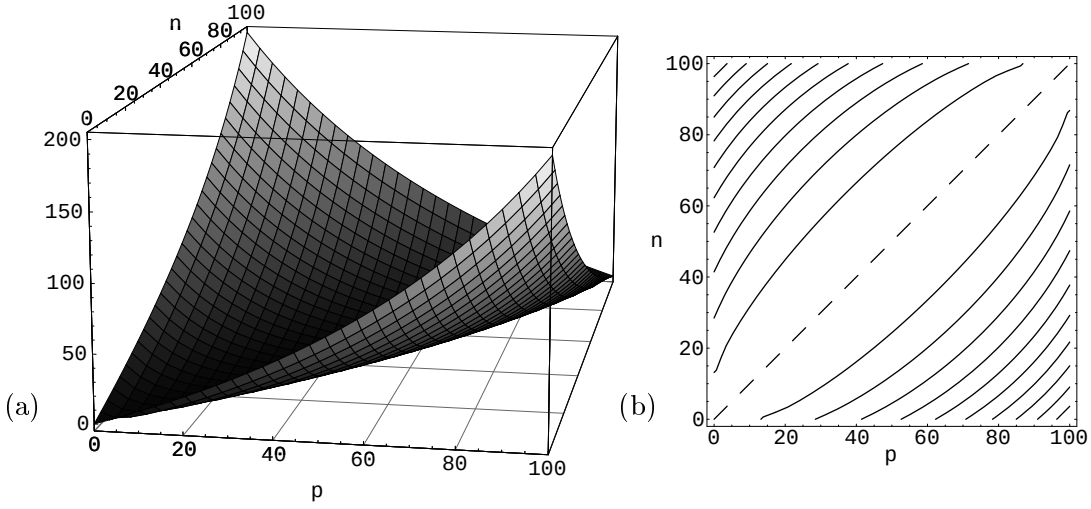

FIGURE 8: (a) χ^2 function. (b) A contour plot of (a).

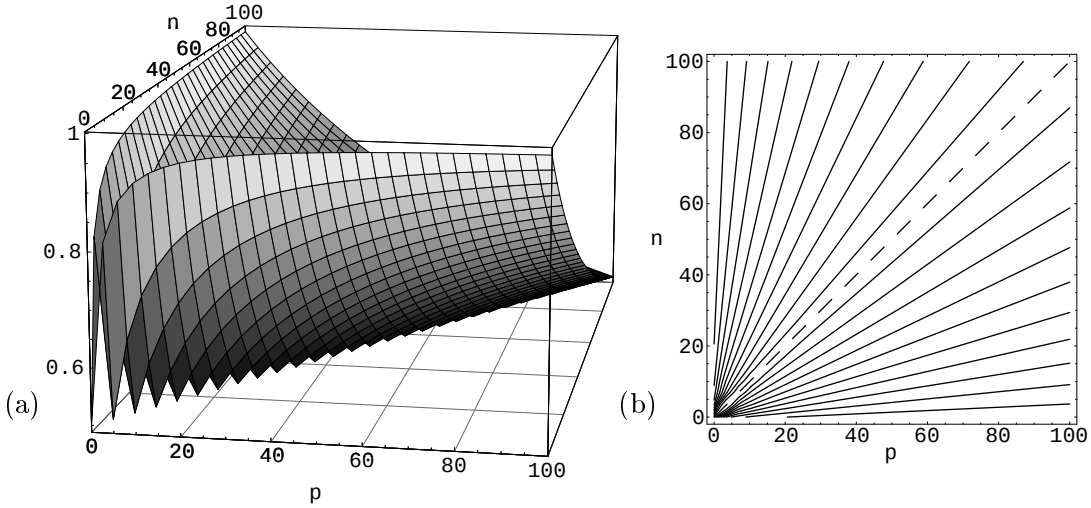

FIGURE 9: (a) Laplace function. (b) A contour plot of (a).

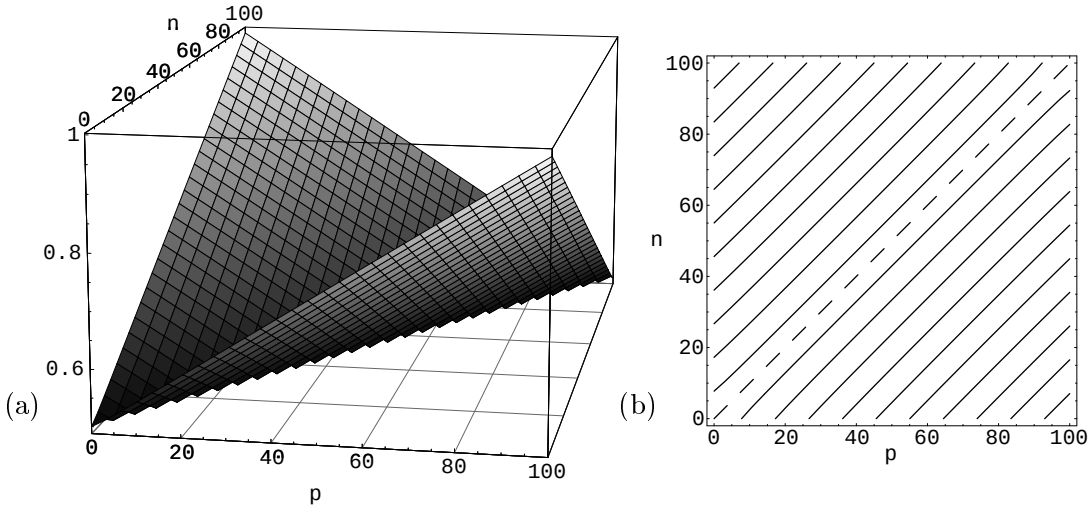

FIGURE 10: (a) The alternate Laplace function. (b) A contour plot of (a).

APPENDIX B. THE RULE LEARNING ALGORITHM

The learning algorithm in Section 5 conducts a beam search over the space of expressions, each expression represented as the conjunction of literals (boolean features or their negations). The algorithm adds one literal at a time to the α best previously retained expressions in the beam (the beam starts with the best single literals). In our experiments α is set to 10. The process continues while keeping track of the best global expression F_{best} (see Figure 11). To avoid exploring too many states (i.e., expressions), the size of the search space can be constrained with two operations:

1. A systematic search to avoid redundant expressions (Rymon, 1993; Webb, 1995), Lines 5-6, Fig. 11. Each expression F_i conjoins several boolean features (or their complements), e.g., $F_i = x_1 \bar{x_3} x_5$. Because conjunction is commutative, the search space is defined by avoiding any F_j that is identical to F_i except for the order in which features appear, e.g., $F_j = \bar{x_3} x_5 x_1$.

2. A pruning technique (Webb, 1995; Quinlan, 1995) (Line 7, Fig. 11). Define F_{best} as the best explored expression according to some evaluation metric $M : g(E)$ (e.g., Information Gain), such that, for all explored F_i, $g(F_{\text{best}}) > g(F_i)$. As long as H is monotonic, F_i can be eliminated if the best value it can ever attain along its search path – according to M – is worse than F_{best}.

The search above terminates when no more expressions are left on the beam, at which point the best explored expression, F_{best}, is used to construct the final hypothesis rule.

Algorithm 1: Single-Rule Learning Algorithm
Input: Original features, beam width α
Output: A single hypothesis rule
BEST_RULE(L_{bool})
(1) Let L_{bool} be the list of all literals
(2) $L_{\text{beam}} \leftarrow$ best α literals in L_{bool}
(3) **while** (**true**)
(4) $L_{\text{new}} \leftarrow$ Systematically form the conjunction
(5) of every $F_i \in L_{\text{beam}}$ with every $F_j \in L_{\text{bool}}$
(6) Eliminate unpromising expressions from L_{new}
(7) **if** $L_{\text{new}} = \emptyset$
(8) **break**
(9) $L_{\text{beam}} \leftarrow$ best α combinations in L_{new}
(10) **end while**
(11) Let c be the majority class in F_{best} (best combination)
(12) **return if** F_{best} **then** c

FIGURE 11: A learning algorithm that returns a single rule as the final hypohesis.

REFERENCES

Andrews, D., & Herzberg, A. (1985). *Data: A Collection of Problems from Many Fields for the student and Research Worker*. Springer-Verlag.

Breiman, L. (1996). Technical note: Some properties of splitting criteria. *Machine Learning*, *24*, 41–47.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of the Fifth European Working Session on Learning*, pp. 164–178. Springer-Verlag.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*(4), 261–284.

Codrington, C., & Brodley, C. E. (1999). On the qualitative behavior of impurity-based splitting rules 1: The minima-free property. Tech. rep. TR 97-5, Purdue University.

Cohen, P. (1995). *Empirical Methods for Artificial Intelligence* (First edition). MIT Press.

Dietterich, T. G., Kearns, M., & Mansour, Y. (1996). Applying the weak learning framework to understand and improve c4.5. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 96–104.

Elomaa, T., & Rousu, J. (1999). General and efficient multisplitting of numerical attributes. *Machine Learning*, *36*(3), 201–244.

Hong, S. J. (1997). Use of contextual information for feature ranking and discretization. In *IEEE Transactions of Knowledge and Data Engineering*.

Jensen, D., & Cohen, P. (1997). Overfitting explained. In *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics*, pp. 115–122.

Jensen, D., & Cohen, P. (1999). Multiple comparisons in induction algorithms. *Machine Learning*, *To Appear*.

Kearns, M., & Mansour, Y. (1996). On the boosting ability of tpo-down decision tree learning algorithms. In *Proceedings of the 28th Annual ACM Symposium on the theory of Computing*, pp. 459–468.

Kira, K., & Rendell, L. (1992). A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, pp. 249–256. Morgan Kaufmann Publishers, Inc.

Kohavi, R. (1995). A study of cross validation and boostrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1137–1143. Morgan Kaufmann.

Kononenko, I. (1994). Estimating attributes: Analysi and extensions of relief. In *Proc. European Conf. on Machine Learning*, pp. 171–182. Springer Verlag.

Kononenko, I. (1995). On biases in estimating multi-valued attributes. In *IJCAI-95*, pp. 1034–1040.

Kononenko, I., & Hong, S. J. (1997). Attribute selection for modelling. *Future Generation Computer Systems.*

Liu, W., & White, A. (1994). The importance of attribute selection measures in decision tree induction. *Machine Learning*, *15*, 25–41.

Merz, C., & Murphy, P. (1998). *UCI repository of machine learning databases.* Available: www.ics.uci.edu/ mlearn/MLRepository.html.

Mingers, J. (1989). An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, *3*, 319–342.

Morishita, S. (1998). On clasiification and regression. *Lecture Notes in Artificial Intelligence*, *1532*, 40–57.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1994). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, Inc.

Quinlan, R. (1995). Oversearching and layered search in empirical learning. In *IJCAI-95*, pp. 1019–1024. Morgan Kaufmann.

Rymon, R. (1993). An SE-tree based characterization of the induction problem. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 268–275. San Francisco: Morgan Kaufmann.

Upton, G. (1982). A comparison of alternative tests for the 2×2 comparative trial. *Journal of the Royal Statistical Society, A Series, 145*, 86–105.

Vilalta, R., & Oblinger, D. (2000). A quantification of distance-bias between evaluation metrics in classification. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 1087–1094. Morgan Kaufman.

Webb, G. I. (1995). Opus: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, *3*, 431–435.

White, A., & Liu, W. (1994). Bias in information-based measures in decision tree induction. *Machine Learning*, *15*, 321–329.

Wolfram, S. (1999). *The Mathematica book 4th Ed.* Wolfram Media/Cambridge University Press.

List of Figures

1 (a) cross-classification of expression values and classes; (b) probabilities estimated from the data. 3
2 (a) Information Gain as a function of the possible coverage (number of positive and negative examples) of an expression. (b) A contour plot of (a) showing isometric lines over the coverage plane. 4
3 (a) The difference in bias between metrics M_1 and M_2 represented as the area between $n : g_1(p)$ and $n : g_2(p)$. (b) Three points positioned at different regions of the coverage plane. 7
4 (a) A contour plot for the Laplace function (b) A contour plot for the alternate definition of the Laplace function in which both an expression E and its complement E' are taken into account. 9
5 (a) Gain Ratio function. (b) A contour plot of (a). 16
6 (a) G function. (b) A contour plot of (a). 16
7 (a) Gini function. (b) A contour plot of (a). 16
8 (a) χ^2 function. (b) A contour plot of (a). 17
9 (a) Laplace function. (b) A contour plot of (a). 17
10 (a) The alternate Laplace function. (b) A contour plot of (a). 17
11 A learning algorithm that returns a single rule as the final hypohesis. 18

List of Tables

1 A pair-wise comparison of evaluation metrics. 8
2 A comparison of the two versions of the Laplace function against Information Gain. 10
3 Results on real-world domains. Numbers enclosed in parentheses represent standard deviations. 13
4 Correlation coefficients on data comparing the distance-bias measure vs. differences in accuracy for a single-rule hypothesis. 15