An Information Theoretic Metric for Multi-Class Categorization

Sam Steingold
Magnetic Media Online
sds@magnetic.com

Michal Laclavík
Magnetic Media Online
laclavik@magnetic.com

ABSTRACT

The most common metrics used to evaluate a classifier are *accuracy*, *recall* and *precision*, and F_1*-score*. These metrics are widely used in machine learning, information retrieval, and text analysis (e.g., text categorization). Each of these metrics is imperfect in some way (captures only one aspect of predictor performance and can be fooled by a weird data set). None of them can be used to compare predictors across different datasets. In this paper we present an information-theoretic performance metric which does not suffer from the aforementioned flaws and can be used in both classification (binary and multi-class) and categorization (each example can be placed in several categories) settings. The code[1] to compute the metric is available under the Apache open-source license.

Keywords

Probability Theory, Information Theory, Predictive Analytics, Multi-Class Prediction, Entropy, Information Retrieval, Text Categorization

1. MOTIVATION

In this paper we use the terms *predictor* and *classifier* interchangeably to mean an algorithm which maps *observations* (or *examples*) into a discrete set of classes (e.g., $\{F, T\}$ or $\{0, 1\}$ for a binary predictor).

Evaluating a predictor is a fundamental data science task which often requires understanding of how the predictor will be used and the relative cost of correct vs. erroneous prediction. These costs are usually unavailable, and "application-independent" metrics are used.

In the case of binary (true/false) prediction, the most common metrics[2] are ($\mathbb{P}$ denotes probability):

$$\begin{aligned}
\text{Accuracy} &= \mathbb{P}(\text{Actual} = \text{Predicted}) \\
&= \frac{\text{tp} + \text{tn}}{N} \\
\text{Recall} &= \mathbb{P}(\text{Predicted} = 1 | \,\text{Actual} = 1) \\
&= \frac{\text{tp}}{\text{tp} + \text{fn}} \\
\text{Precision} &= \mathbb{P}(\text{Actual} = 1 | \,\text{Predicted} = 1) \\
&= \frac{\text{tp}}{\text{tp} + \text{fp}}
\end{aligned}$$

Here tp is the *true positive* count, fn is the *false negative* count, etc, and $N = \text{tp} + \text{fp} + \text{fn} + \text{tn}$ is the total population. These metrics have an important advantage of having a *clear meaning*:

Accuracy - the percentage of the correct predictions

Recall - the percentage of the positive examples captured by the predictor (how many bad guys do we catch)

Precision - the percentage of the positive examples among those identified by predictor as positive (how many guys we caught are actually bad)

but each has a blind spot:

- Accuracy misses the *San Diego weather forecast* problem: always predicting *false* for rare events gives a very high accuracy while revealing no information about the actual events.
- Recall and Precision miss complementary problems: predicting *true* generously increases Recall and decreases Precision while preferring *false* decreases Recall and improves Precision.

The important feature (or bug, depending on how one looks at it) of Recall and Precision is that they treat positive and negative outcomes differently, i.e., they are *not invariant* (asymmetric) with respect to swapping the 0 and 1 labels. Specifically, Recall turns into *Specificity* while Precision becomes *Negative predictive value*. Wikipedia[3] lists these and many more similar metrics, all of which suffer from the same flaw of capturing a single aspect of the predictor's performance.

[1] `https://github.com/Magnetic/proficiency-metric`

[2] *Recall* is sometimes known as *Sensitivity*.

[3] `http://en.wikipedia.org/wiki/Confusion_matrix`

These flaws lead to a widespread use of F_1-*score* (or F_1-*measure*) which is the harmonic mean of Recall and Precision:

$$\begin{aligned} F_1 &= \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \\ &= \frac{2 \times \text{tp}}{2 \times \text{tp} + \text{fp} + \text{fn}} \end{aligned}$$

or, more generally,

$$\begin{aligned} F_\beta &= \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \\ &= \frac{(1+\beta^2) \times \text{tp}}{(1+\beta^2) \times \text{tp} + \beta^2 \times \text{fn} + \text{fp}} \end{aligned}$$

which "attaches β times as much importance to recall as precision" [3] but, in practice, $\beta \neq 1$ is never used.

The advantage of F_1-score is that it is low whenever *either* Recall or Precision is low, thus its high value seems to indicate a good predictor. The first problem with F_1-score is that it has no meaning other than being the mean of two meaningful numbers. Thus, knowing a specific value of F_1 limits the range of Recall and Precision but does not give a clear insight into the predictor's performance.

F_1 also inherits the aforementioned asymmetry of Recall and Precision: it treats positive and negative outcomes differently. This leads to F_1 being fooled by an unusually high base rate (also known as *prevalence*). E.g., if the base rate $(\text{tp} + \text{fn})/N$ is, say, 90%, then the worthless (random with $\mathbb{P}(\text{positive}) = 0.9$) predictor will have a very respectable 90% Recall, Precision, and F_1-score (not to mention the 82% Accuracy). This example looks contrived, but it shows that the F_1-scores (and Accuracy, Recall, Precision etc) are *not* comparable across data sets. E.g., we know that that a Deep Learning algorithm achieves 1.2% error rate $(1 - \text{Accuracy})$ on the MNIST Dataset [8] and 8.4% on the OCR Dataset [9]. Is this difference in performance a result of the datasets' compositions or is there a deeper difference?

The mere multitude of metrics used indicates an unsatisfactory state of affairs: we need a single simple metric which satisfies the following conditions:

Meaning : the meaning of the metric should be transparent without resorting to averages of meaningful values

Discrimination :

- **Weak** : its value is 1 for the perfect predictor (and only for it)
- **Strong** : *additionally*, its value is 0 for a worthless (random with any base rate) predictor (and only for such a predictor)

Universality : the metric should be usable in any setting, whether binary or multi-class, classification (a unique class is assigned to each example) or categorization/community detection (an example can be placed into multiple categories or communities)

In case of binary (two-class) prediction, we do have a nice metric which satisfies the first two conditions: the ϕ coefficient (introduced by Karl Pearson himself), also known as Matthews correlation coefficient (rediscovered 70 years later):

$$\phi = \frac{\chi}{\sqrt{N}} = \frac{\text{tp} \times \text{tn} - \text{fp} \times \text{fn}}{\sqrt{(\text{tp}+\text{fp}) \times (\text{tp}+\text{fn}) \times (\text{fp}+\text{tn}) \times (\text{fn}+\text{tn})}}$$

However, it does not generalize too well to multi-class prediction (see section 3). Its meaning (when it is neither 0 nor 1) is also far from transparent.

2. INFORMATION THEORY TO THE RESCUE!

The goal of data science is extracting *information* from data. Thus it is only natural to measure performance of a predictor as the percentage of information it extracts from the data:

$$\alpha = \frac{I(\text{Predicted}; \text{Actual})}{H(\text{Actual})} \tag{1}$$

where (A = Actual and P = Predicted)

$$H(A) = -\sum_{i=1}^{2} \mathbb{P}(A = i) \log_2 \mathbb{P}(A = i)$$

is the *entropy* of the actual distribution and

$$I(P; A) = \sum_{i=1}^{2} \sum_{j=1}^{2} \mathbb{P}(A = i \& P = j) \log_2 \frac{\mathbb{P}(A = i \& P = j)}{\mathbb{P}(A = i)\, \mathbb{P}(P = j)}$$

is the *mutual information* of the two random variables.

Despite the apparent complexity of the formulas, the *meaning* is clear: α measures how many bits of information contained in the data (H) are captured by the model (I).

This metric was called *uncertainty coefficient* in [1] and *proficiency metric* in [2]. We will stick with the latter because it emphasizes the use of α as a *performance metric* and not just a *measure of association.*

2.1 Properties of the Proficiency Metric

Both entropy and mutual information are non-negative, and the range of proficiency metric is the interval $[0; 1]$.

For a perfect predictor Actual = Predicted and $I = H$, thus $\alpha = 1$ as desired.

If the predictor is worthless, i.e., *independent* from the actual value, $I = 0$ and, thus, $\alpha = 0$ as prescribed.

If the predictor is *mislabeled*, i.e., if Predicted = 1 whenever Actual = 0, and Predicted = 0 whenever Actual = 1, in other words, tp = tn = 0, then one can easily calculate $\alpha = 1$. This observation can give one a start, but it can be argued that this is actually a desirable feature, especially in a situation when the class *labels* are less interesting than class *membership.* A good example of such situation is group detection, when class (i.e., group) label is often not even defined properly.

Specifically, a very important problem in on-line marketing is cross-device user identification, i.e., establishing which different cookies on different devices (e.g., laptops and phones) represent the same individual. This is, essentially, a group detection problem, which is best evaluated using using the Proficiency Metric because the class labels are irrelevant.

2.2 Example

Figure 1 compares the values of the four performance metrics for a range of predictors based on a population of size 200 with a 10% base rate, with the following confusion matrices:

$$\begin{bmatrix} \text{tp} = (200 - i)\frac{1}{10} & \text{fn} = i\frac{1}{10} \\ \text{fp} = i\frac{9}{10} & \text{tn} = (200 - i)\frac{9}{10} \end{bmatrix}$$

One can see that all the metrics properly return 1 for the perfect predictor corresponding to $i = 0$, but then their

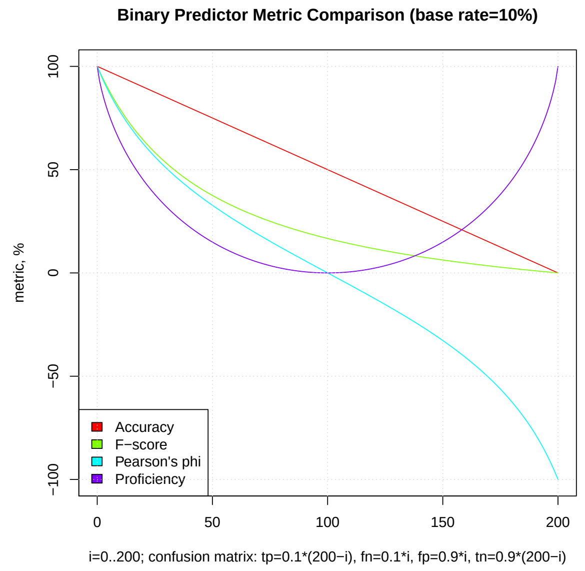

Figure 1: Comparison of Accuracy, F_1-score, Pearson's ϕ and the Proficiency metric for different binary predictors.

values diverge. Proficiency metric quickly declines to 15% for $i = 50$, when the confusion matrix is

$$\begin{bmatrix} \text{tp} = 15 & \text{fn} = 5 \\ \text{fp} = 45 & \text{tn} = 135 \end{bmatrix}$$

Accuracy is 75% and $F_1 = 38\%$, while Pearson's $\phi = 33\%$. When $i = 100$, the Predicted and Actual are independent, i.e., the predictor tells us absolutely nothing about the actual values, and the confusion matrix is

$$\begin{bmatrix} \text{tp} = 10 & \text{fn} = 10 \\ \text{fp} = 90 & \text{tn} = 90 \end{bmatrix}$$

Here $\alpha = \phi = 0$ and Accuracy is 50% while $F_1 = 17\%$. Finally, when $i = 200$, the predictor is perfect (up to swapping the predicted labels), $\phi = -1$ and $\alpha = 1$, both indicating that the predictor is perfect (ϕ also capturing the need to re-label the predictor), but both Accuracy and F_1 are 0.

2.3 Monotonicity

The above example may give a mistaken impression that, except for some exotic, poorly performing predictors, all metrics move in step, e.g., better Accuracy implies better Proficiency, better F_1-score, and better Pearson's ϕ. This is often, but not always, the case.

In the following examples we consider different predictors on the same set of 50 examples, 5 positive and 45 negative (10% base rate).

2.3.1 2 Against 2

Here Proficiency agrees with F_1-score against Pearson's ϕ and Accuracy:

$$A = \begin{bmatrix} \text{tp} = 2 & \text{fn} = 3 \\ \text{fp} = 0 & \text{tn} = 45 \end{bmatrix}; B = \begin{bmatrix} \text{tp} = 5 & \text{fn} = 0 \\ \text{fp} = 7 & \text{tn} = 38 \end{bmatrix}$$

	A	B
Proficiency	30.96%	**49.86%**
Pearson's ϕ	**61.24%**	59.32%
Accuracy	**94.00%**	86.00%
F_1-score	57.14%	**58.82%**

In this example Proficiency agrees with Pearson's ϕ against F_1-score and Accuracy:

$$A = \begin{bmatrix} \text{tp} = 3 & \text{fn} = 2 \\ \text{fp} = 2 & \text{tn} = 43 \end{bmatrix}; B = \begin{bmatrix} \text{tp} = 5 & \text{fn} = 0 \\ \text{fp} = 7 & \text{tn} = 38 \end{bmatrix}$$

	A	B
Proficiency	28.96%	**49.86%**
Pearson's ϕ	55.56%	**59.32%**
Accuracy	**92.00%**	86.00%
F_1-score	**60.00%**	58.82%

Interestigly enough, in Pearson's ϕ never disagrees with *both* Proficiency and Accuracy *at the same time.*

2.3.2 Proficiency – The Odd One Out

Here Proficiency disagrees with the other 3 metrics:

$$A = \begin{bmatrix} \text{tp} = 3 & \text{fn} = 2 \\ \text{fp} = 1 & \text{tn} = 44 \end{bmatrix}; B = \begin{bmatrix} \text{tp} = 5 & \text{fn} = 0 \\ \text{fp} = 6 & \text{tn} = 39 \end{bmatrix}$$

	A	B
Proficiency	35.55%	**53.37%**
Pearson's ϕ	**63.89%**	62.76%
Accuracy	**94.00%**	88.00%
F_1-score	**66.67%**	62.50%

2.3.3 Accuracy – The Odd One Out

In this example Accuracy stands alone:

$$A = \begin{bmatrix} \text{tp} = 1 & \text{fn} = 4 \\ \text{fp} = 0 & \text{tn} = 45 \end{bmatrix}; B = \begin{bmatrix} \text{tp} = 5 & \text{fn} = 0 \\ \text{fp} = 13 & \text{tn} = 32 \end{bmatrix}$$

	A	B
Proficiency	14.77%	**34.57%**
Pearson's ϕ	42.86%	**44.44%**
Accuracy	**92.00%**	74.00%
F_1-score	33.33%	**43.48%**

2.3.4 F_1-score – The Odd One Out

Here is an example when F_1 argues against the rest:

$$A = \begin{bmatrix} \text{tp} = 1 & \text{fn} = 4 \\ \text{fp} = 0 & \text{tn} = 45 \end{bmatrix}; B = \begin{bmatrix} \text{tp} = 2 & \text{fn} = 3 \\ \text{fp} = 2 & \text{tn} = 43 \end{bmatrix}$$

	A	B
Proficiency	**14.77%**	14.71%
Pearson's ϕ	**42.86%**	39.32%
Accuracy	**92.00%**	90.00%
F_1-score	33.33%	**44.44%**

2.4 A Note On The Re-Labeled Predictor

For a predictor P, let $1 - P$ be the re-labeled predictor, i.e., when P predicts 1, $1 - P$ predicts 0 and vice versa. Then

$$\begin{aligned}
\text{Accuracy}(1-P) &= 1 - \text{Accuracy}(P) \\
\phi(1-P) &= -\phi(P) \\
\alpha(1-P) &= \alpha(P)
\end{aligned}$$

and no similar simple relationship exists for F_1.

3. MULTI-CLASS PREDICTION

When the classification problem has multiple classes (e.g., character recognition), neither Recall nor Precision (and thus F_1) make sense, because there is no preferred "positive" outcome.

One can, of course, compute Recall$_c$, Precision$_c$, and F_c for each class c and then define an average of all the class-specific F_c-scores as the model F-score. These metrics are criticized for giving the same weight to all classes regardless of their frequency, and, of course, variations trying to fix those issues abound.

One can still define Pearson's ϕ using

$$\phi^2 = \frac{\chi^2}{N} = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where

$$\begin{aligned}
O_{ij} &= \mathbb{P}(\text{Predicted} = i \& \text{ Actual} = j) \\
E_{ij} &= \mathbb{P}(\text{Predicted} = i) \times \mathbb{P}(\text{Actual} = j)
\end{aligned}$$

Alas, the meaning (away from $\phi = 0$, when the predictor is worthless, being independent from the actual distribution) is now opaque and the range is $[0; \infty]$ instead of $[-1; 1]$ which makes it worse than our $\alpha \in [0; 1]$. Note that Pearson's ϕ's relationship to the χ^2 test is similar to α's relationship to the Likelihood-ratio test [6]. This is another argument in favor of the Proficiency metric, because the Likelihood-ratio test has the highest power among all competitors. It has been traditionally neglected in favor of χ^2 because of the prohibitive cost of logarithm calculation in the pre-computer days.

All we are left with are Accuracy, still defined as $\mathbb{P}(\text{Actual} = \text{Predicted})$, and Proficiency metric α, defined above in formula 1.

3.1 Examples

Consider a classification problem where the probability of class $i = 1, \ldots$ is $\mathbb{P}(i) = 2^{-i}$. The Accuracy of the worthless predictor (i.e., the predictor independent from the actual variable but distributed identically to it) is

$$\sum_{i=1}^{\infty} 2^{-2i} = \frac{1}{3}$$

while the Proficiency metric is, of course, $\alpha = 0$.

We applied the popular machine learning library Vowpal Wabbit[4] to two common datasets MNIST [8] and OCR [9]. The performance[5] was

	Proficiency	Accuracy	Random Accuracy
MNIST	94.24%	97.97%	10.04%
OCR	81.41%	88.05%	5.91%

Note that it is not clear whether the better Accuracy on MNIST is due to the smaller variety of data (i.e., larger random accuracy) or to the better predictive power of the algorithm in that setting.

However, we see that the Proficiency metrics are different, i.e., VW captures a larger share of information from the digits than from the letters, i.e., it actually does performs better on the MNIST data set.

4. MULTI-LABEL CATEGORIZATION

We now consider the problem of labeling objects where each object can have *several* labels. Examples include text categorization (the same article can be both `News` and `Sports`) and community detection (the same researcher might publish papers on both `Physics` and `Biology`).

4.1 Precision and Recall

Denote the categories as c_i, objects as o_j. Our old friends Recall and Precision are defined as ([3], [4]):

$$\begin{aligned}
\text{Recall} &= \frac{\sum_i \#\{\text{objects correctly classified as } c_i\}}{\sum_i \#\{\text{objects classified as } c_i\}} \\
&= \frac{\sum_i \#\{o_j | c_i \in \text{Actual}(o_j) \cap \text{Predicted}(o_j)\}}{\sum_i \#\{o_j | c_i \in \text{Predicted}(o_j)\}} \\
&= \frac{\sum_j \# \,\text{Actual}(o_j) \cap \text{Predicted}(o_j)}{\sum_j \# \,\text{Predicted}(o_j)} \\
\text{Precision} &= \frac{\sum_i \#\{\text{objects correctly classified as } c_i\}}{\sum_i \#\{\text{objects actually in } c_i\}} \\
&= \frac{\sum_i \#\{o_j | c_i \in \text{Actual}(o_j) \cap \text{Predicted}(o_j)\}}{\sum_i \#\{o_j | c_i \in \text{Actual}(o_j)\}} \\
&= \frac{\sum_j \# \,\text{Actual}(o_j) \cap \text{Predicted}(o_j)}{\sum_j \# \,\text{Actual}(o_j)}
\end{aligned}$$

And F_1 is, as before, their harmonic average.

Note that in the setup of section 3, i.e., if every object is always categorized into a single category, the above definitions collapse into the definition of Accuracy, i.e., Accuracy = Recall = Precision = F_1.

Note also that this definition is *symmetric* with respect to the categories (i.e., it is invariant under *simultaneous* permutations of both actual and predicted categories), in contrast to the Recall and Precision in binary classification, see section 1.

In short, both Precision and Recall look much more reasonable in this setting, although neither captures *all* aspects of the classifier, necessitating the use of F_1 (see, however, the commitee example at the end of section 4.4).

4.2 Accuracy

Defining Accuracy = $\mathbb{P}(\text{Actual} = \text{Predicted})$ makes little sense in this setting because it ignores the nuances of partial correctness. E.g., the string `athlon xp` can be categorized as

1. `Computers\Hardware`

[4] `http://hunch.net/~vw/`

[5] *Random Accuracy* is the accuracy which can be expected from a random predictor which predicts every class (digit for MNIST and character for OCR) *randomly* with the same probability as the frequency of the class in the dataset.

2. `Shopping\Buying Guides & Researching`
3. `Information\Companies & Industries`

or

1. `Computers\Hardware`
2. `Computers\Other`
3. `Computers\Mobile Computing`

and the above definition of Accuracy ignores the presence of a common category while the Precision and Recall take it into account.

4.3 Proficiency

To define the Proficiency metric, we define binary random variables

$$\begin{aligned}\mathrm{Ac}_i &:= c_i \in \mathrm{Actual}\\ \mathrm{Pc}_i &:= c_i \in \mathrm{Predicted}\end{aligned}$$

i.e., $\mathrm{Ac}_i(o_j)$ is 1 iff o_j is actually in the category c_i, etc. Then we can use the usual definition of the Proficiency metric:

$$\alpha = \frac{I(\prod_i \mathrm{Pc}_i; \prod_i \mathrm{Ac}_i)}{H(\prod_i \mathrm{Ac}_i)}$$

where $\prod$ denotes the Cartesian product of random variables.

The apparent problem with this definition is that one cannot really compute anything based on it because the Cartesian product is just too large. The taxonomy used in the KDD Cup 2005 [5] had 67 categories; the size of the Cartesian product is $2^{67} > 10^{20}$ and dwarfs the $800,000$ examples provided. This means that the resulting experimental confusion matrix is extremely sparse, which leads to the Proficiency metric computed from it being always close to 1.

However, all is not lost as we can try to *estimate* the Proficiency metric making the same *independence* assumption which made the Naive Bayes modeling feasible.

We know that $H(A\times B) \geq H(A)+H(B)$ with equality iff A and B are independent. This takes care of the denominator. For the numerator, we assume that Pc_i is independent on everything but Ac_i. This leads to the definition

$$\begin{aligned}\alpha &= \frac{\sum_i I(\mathrm{Pc}_i; \mathrm{Ac}_i)}{\sum_i H(\mathrm{Ac}_i)}\\ &= \frac{\sum_i H(\mathrm{Ac}_i)\alpha(\mathrm{Pc}_i, \mathrm{Ac}_i)}{\sum_i H(\mathrm{Ac}_i)}\end{aligned}$$

where

$$\alpha(\mathrm{Pc}_i, \mathrm{Ac}_i) = \frac{I(\mathrm{Pc}_i; \mathrm{Ac}_i)}{H(\mathrm{Ac}_i)}$$

is the Proficiency metric restricted to the i-th category. Note that the total Proficiency is now defined as a weighted mean of proficiencies for individual categories, the weights being the actual category entropies.

The problem with this definition is that it is now sensitive to predicted category re-labeling, unlike the original Proficiency metric defined by formula 1 (cf. section 2.1). This sensitivity is detrimental in the community detection problem, but may seem to be desired in the problem of categorization (see, however, the next section, where we observe that human labelers, apparently, confuse categories).

Actual	labeler 1	labeler 2	labeler 3
Predicted	labeler 2	labeler 3	labeler 1
Precision	63.48%	36.50%	58.66%
Recall	41.41%	58.62%	55.99%
α	24.73%	28.06%	33.26%
α'	25.02%	28.62%	33.51%
reassigned	9	12	11

Table 1: Cross-comparison of hand-labeled sets

To recover the re-labeling detection property, we use the Munkres algorithm [7] to assign predicted and actual categories, using the matrix of pairwise mutual informations[6], defining the *Permuted Proficiency metric*:

$$\begin{aligned}\alpha' &= \frac{\sum_i I(M(\mathrm{Pc}_i); \mathrm{Ac}_i)}{\sum_i H(\mathrm{Ac}_i)}\\ &= \frac{\sum_i H(\mathrm{Ac}_i)\alpha(M(\mathrm{Pc}_i), \mathrm{Ac}_i)}{\sum_i H(\mathrm{Ac}_i)}\end{aligned}$$

where $M(c)$ is the optimal category assignment. The optimality of M implies $\alpha \leq \alpha'$ with equality iff the optimal assignment is the identity.

The unfortunate side effect of the independence assumption we made is that it weakens our claim that the Proficiency metric can be used to compare classifiers across domains and data sets. However, the **Strong Discrimination** property (see sections 1 and 6) is preserved.

4.4 Examples

3 persons categorized the same 800 queries from the KDD Cup 2005 [5] dataset into 67 categories. Comparing them against each other one by one, we get the results presented in table 4.4. One can see that the second labeler is the odd one out, indicated by the Precision and Recall values.

The labelers also missed some categories altogether:

1. The first labeler never assigned the `Information\Local & Regional Information\Education` category.
2. The second labeler never assigned the `Sports\Olympic Games` category.
3. The third labeler never assigned the `Computers\Mobile Computing` and `Information\Local & Regional Information\Education` categories.

An additional insight can be gained from the optimal category assignment between the labelers, see tables 2, 3, 4:

1. It appears that the second labeler merely forgot about the special `Sports\Olympic Games` category and used `Sports\News & Scores` instead.
2. Also, the first two labelers used `Computers\Mobile Computing` when the third one preferred `Computers\Networks & Telecommunication`.

It appears that 67 categories is too much for a human to keep in mind.

[6] We could also use the pairwise Proficiency metrics instead, but then we will lose the $\alpha \leq \alpha'$ property.

α_c	Labeler 1	N	Labeler 2	N
31.41%	Sports\ Olympic Games	2	Sports\ News & Scores	3
29.43%	Online Community\ Personal Services	3	Online Community\ Forums & Groups	18
10.42%	Online Community\ Other	8	Information\ Local & Regional Information\ Education	1
5.46%	Shopping\ Buying Guides & Researching	234	Information\ References & Libraries	93
0.42%	Information\ References & Libraries	77	Online Community\ Personal Services	2
0.41%	Shopping\ Other	51	Shopping\ Buying Guides & Researching	33
0.36%	Online Community\ Forums & Groups	21	Shopping\ Other	13
0.02%	Sports\ News & Scores	8	Online Community\ Other	1
0.00%	Information\ Local & Regional Information\ Education	0	Sports\ Olympic Games	0

Table 2: Labeler 1 vs. Labeler 2

α_c	Labeler 2	N	Labeler 3	N
58.55%	Computers\ Mobile Computing	7	Computers\ Networks & Telecommunication	13
48.46%	Online Community\ Personal Services	2	Information\ Other	28
15.61%	Online Community\ Chat & Instant Messaging	3	Online Community\ Personal Services	6
11.90%	Computers\ Networks & Telecommunication	8	Living\ Other	7
9.66%	Online Community\ Forums & Groups	18	Online Community\ Chat & Instant Messaging	5
7.31%	Computers\ Other	11	Computers\ Multimedia	26
3.85%	Computers\ Multimedia	14	Online Community\ Other	2
3.13%	Living\ Other	14	Shopping\ Other	3
3.09%	Shopping\ Other	13	Online Community\ Forums & Groups	116
0.09%	Information\ Other	9	Computers\ Other	4
0.00%	Information\ Local & Regional Information\ Education	1	Computers\ Mobile Computing	0
0.00%	Online Community\ Other	1	Information\ Local & Regional Information\ Education	0

Table 3: Labeler 2 vs. Labeler 3

α_c	Labeler 3	N	Labeler 1	N
39.67%	Computers\ Networks & Telecommunication	13	Computers\ Mobile Computing	6
28.69%	Online Community\ Chat & Instant Messaging	5	Online Community\ Personal Services	3
19.94%	Online Community\ Other	2	Living\ Other	13
14.18%	Computers\ Other	4	Online Community\ Homepages	36
8.73%	Online Community\ Personal Services	6	Online Community\ Chat & Instant Messaging	3
5.63%	Shopping\ Other	3	Information\ Other	44
3.62%	Living\ Other	7	Computers\ Networks & Telecommunication	12
1.24%	Online Community\ Homepages	351	Shopping\ Other	51
0.33%	Information\ Local & Regional	59	Computers\ Other	9
0.24%	Information\ Other	28	Online Community\ Other	8
0.00%	Computers\ Mobile Computing	0	Information\ Local & Regional	51

Table 4: Labeler 3 vs. Labeler 1

We can also pit each of the three labelers against the random labeler with the same category probability distribution. The resulting Proficiency metrics are, of course, 0, and Precision = Recall = F_1:

	Labeler 1	Labeler 2	Labeler 3
F_1	14.3%	7.7%	19.2%
examples/category	3.7 ± 1.1	2.4 ± 0.9	3.8 ± 1.1
categories/example	44 ± 56	28 ± 31	48 ± 71

The differences in the F_1 values are due to the different distributions of the number of examples per category and the number of categories per example.

Taking this example one step further, consider a typical University department, where every professor serves on 9 administrative commitees out of 10 available. Then the worthless predictor which assigns each professor to random 9 commitees will have Precision = Recall = F_1 = 90% (of course, the Proficiency metrics is still $\alpha = 0$).

5. NUMERIC STABILITY

In this age of Big Data, it is reasonable to think of the data as an infinite stream of observations, and view the actually available data as a sample. This perspective leads to the question of *numeric stability* of the various metrics, i.e., how they would change if the sample were different. To answer this question, we split the test data into two halves at random 500 times and computed the metrics on these 1,000 datasets[7]. Interestingly enough, all metrics have approximately the same variability (standard deviation around 1% for 800 observations of the three human labelers in section 4.4 and around 0.5% for 10,000 observations in the Magnetic data set).

6. CONCLUSION

The advantages of the Proficiency metric we advocate are the combination of two important properties:

Clear Meaning : the proportion of the information contained in the actual distribution which is captured by the classifier.

Strong Discrimination : $\alpha = 0$ iff the classifier is worthless, $\alpha = 1$ iff the classifier is perfect.

Many other common metrics have the **Clear Meaning** property (e.g., Accuracy, Recall, Precision - but *not* F_1), and some have the **Weak Discrimination** property (e.g., Accuracy and F_1) but *none* have the **Strong Discrimination** property: both Accuracy and F_1 can be close to 1 for a worthless predictor.

We consider three machine learning problems:

Binary Prediction (section 1): All metrics can be computed, but all of them - with the notable exception of the Proficiency metric - have serious flaws.

Multi-Class Prediction (section 3): Precision and Recall no longer make sense; Accuracy can still be defined, but it keeps its flaws and is not as informative as the Proficiency metric.

[7] This methodology is different from cross-validation, where one splits the data and trains the models on different splits. Here we have a *single* model tested on many test sets.

Multi-Label Categorization (section 4): Accuracy no longer makes sense, but Precision and Recall can be meaningfully defined. Proficiency can only be computed approximately, but preserves all its nice properties.

Therefore we can add a third property to **Clear Meaning** and **Strong Discrimination**:

Universality : the Proficiency metric can be used in any setting and has the same meaning in all of them.

No other metric discussed here has this property.

7. REFERENCES

[1] L.A. Goodman, W.H. Kruskal "Measures of Association for Cross Classifications", New York: Springer-Verlag, 1979.

[2] J.V. White, S. Steingold, C.G. Fournelle "Performance Metrics for Group-Detection Algorithms", *Computing Science and Statistics*, 36, 1032-1046, 2004.

[3] C.J. van Rijsbergen "Information Retrieval" London, U.K., 1979.

[4] C.D. Manning and H. Schtze "Foundations of Statistical Natural Language Processing" London, U.K., 1999.

[5] Y. Li, Z. Zheng, H. Dai "KDD CUP-2005 Report: Facing a Great Challenge" ACM SIGKDD Explorations, 7:2, December 2005.

[6] J. Neyman, E.S. Pearson "On the Problem of the Most Efficient Tests of Statistical Hypotheses" Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 231 (694706): 289337, 1933.

[7] J. Munkres "Algorithms for the Assignment and Transportation Problems" Journal of the Society of Industrial and Applied Mathematics, 5(1):32-38, March, 1957.

[8] MNIST Hand-written Digit Dataset (60,000 examples of 10 digits) `http://yann.lecun.com/exdb/mnist/`

[9] OCR Hand-written Character Dataset (42,152 examples of 26 English letters) `http://ai.stanford.edu/~btaskar/ocr/`