Different metrics are used to evaluate classifiers. Many recent papers and shared tasks pick ‘macro’ metrics to rank systems (e.g., ‘macro F1’). However, sometimes the motivation for selecting a specific metric tends to be not fully clear, and the expectations associated with phrases like ‘macro’ seem blurry. Starting from the basic concepts of bias and prevalence, we analyze properties of metrics, with the aim of better metric understanding. In particular, we study Accuracy, macro Precision, macro Recall, macro F1, Matthews Correlation Coefficient, and Kappa.

1 Introduction

Consider a typical scenario in machine learning: We train a classifier to predict some categories of interest and want to assess its capability to predict unseen data.

To this aim, we usually evaluate the classifier’s predictions against reference labels in two steps: First, we summarize the classifier’s behavior in a confusion matrix that has a designated dimension for every possible prediction-label combination. Second, an aggregate statistic, which we here denote as metric, maps the confusion matrix onto a single number.

Obviously, a ‘perfect’ metric often doesn’t exist, since we lose important information about a classifier’s behavior when reducing the confusion matrix to a single number. Nonetheless, for classifier selection or ranking, a suitable metric has to be chosen.

Over the recent years, there has been a surge of papers that use ‘macro’ metrics for evaluation. In particular, ‘macro F1’ has become popular for comparing classifiers and determining shared task winners – its increasing popularity is also reflected in the enormous Google-books corpus (Figure 1).

When searching for reasons why a particular metric has been selected, we tend to find rather unclear statements, e.g.: ‘labels are imbalanced’ or it is wished for that ‘all class labels have equal weight’ (c.f. Table 1).

But such statements seem to point at a certain user need: the evaluation score should not marginalize classification performance of data from infrequent classes. Instead, it should tell us a bigger picture of classifier capability (Greek: makrós, ‘long’) that is robust against fluctuating class prevalences. By contrast, a micro picture (Greek: mikrós, ‘small’) strictly binds the metric score to the class occurrences in a specific sample.

Table 1: Example comments on evaluation metric selection.

<table>
<thead>
<tr>
<th>metric</th>
<th>motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] macro Prec, Recall, F1</td>
<td>‘macro because (…) skewed distribution of the label set’</td>
</tr>
<tr>
<td>[13] macro F1</td>
<td>‘macro-averaging (…) implies that all class labels have equal weight in the final score’</td>
</tr>
<tr>
<td>[17] macro F1</td>
<td>‘Given the strong imbalance between the number of instances in the different classes’</td>
</tr>
<tr>
<td>[4] accuracy, macro F1</td>
<td>‘the labels are imbalanced’</td>
</tr>
<tr>
<td>[5, 15] Mat. Corr. Coef. (MCC)</td>
<td>‘balanced measurement when the classes are of very different sizes’</td>
</tr>
<tr>
<td>[14] MCC, F1</td>
<td>‘(…) imbalanced data (…)’</td>
</tr>
</tbody>
</table>

Figure 1: Google books corpus n-gram search for macro F1.

Outline After introducing Preliminaries (§2), we consider five Metric Properties (§3) to distinguish classification metrics. Then, we use these properties in an Overview (§4) of popular metrics, to improve our understanding about the metrics. Furthermore, we provide a Discussion (§5) that contains some considerations, and a short Summary (§6) of our notes.
2 Preliminaries

Basis For any classifier \( f : D \to C = \{1, \ldots, n\} \) and finite set \( S \subseteq D \times C \), let \( m^{i,j} \in \mathbb{R}^{n \times n} \) be a confusion matrix where \( m^{i,j}_s = |\{s \in S \mid f(s) = i \land s_2 = j\}| \). We omit superscripts whenever possible. A metric \( m : \mathbb{R}^{n \times n} \to \mathbb{R} \) allows us to order confusion matrices, respectively, rank classifiers. For convenience, we say that a classifier \( f \) is better than (or preferable to) a classifier \( g \) if \( \text{metric}(m^{i,j}) > \text{metric}(m^{g,i,j}) \).

Let us define some basic quantities.

**Class bias, prevalence and correct** are given as

\[
\text{bias}(i) = \sum_x m_{i,x} \quad \text{prevalence}(i) = \sum_x m_{x,i} \quad \text{correct}(i) = m_{i,i}
\]

**Class precision** \( P_i \) denotes the precision for class \( i \):

\[
P_i = \frac{\text{correct}(i)}{\text{bias}(i)} \approx \mathcal{P}(\text{class} = i | f \to i)
\]

It approximates the probability of a correct prediction given that the classifier has predicted a specific class (which, for brevity, we denote as \( f \to i \)).

**Class recall** \( R_i \) denotes the recall for class \( i \):

\[
R_i = \frac{\text{correct}(i)}{\text{prevalence}(i)} \approx \mathcal{P}(f \to i | \text{class} = i)
\]

It approximates the probability of a correct prediction given that an example is from a certain class.

3 Defining metric properties

We introduce some useful tools that help us to distinguish among metrics. We define five metric properties: Monotonicity, class sensitivity, class decomposability, prevalence invariance and change correction.

I Monotonicity

This property checks the safety of a metric by imposing that correct (false) predictions do not decrease (increase) the score. We formalize this as

**Property I (Monotonicity).** A metric has PI if \( \forall m: \)

\[
\frac{\partial \text{metric}(m)}{\partial m_{i,j}} \begin{cases} \geq 0 & \iff i = j \\ \leq 0 & \iff i \neq j \end{cases}
\]

1If our classifier predicts a \( n \)-dimensional probability distribution: \( \sigma : D \to \{0, 1\}^n \quad | \sum_{i=1}^n \sigma_i = 1 \), we may also set \( m_{i,j}^{\sigma} = \sum_{s \in \{s_2 = j\}} \sigma(s) \cdot f(s)_i \), with \( |x| = 1 \) if \( x \) is true, else \( 0 \).

II Telling ‘macro’ from ‘micro’: different errors have different weights

Where there is ‘macro’, there is also ‘micro’. We separate micro metrics with

**Property II (Class sensitivity).** If \( \exists m \in \mathbb{R}^{n \times n}: \frac{\partial \text{metric}(m^{i,j})}{\partial m_{i,j}} \neq \frac{\partial \text{metric}(m^{g,i,j})}{\partial m_{i,j}} \) with \( (i, j) \in (C \times C) \) or \( \frac{\partial \text{metric}(m^{i,j})}{\partial m_{i,j}} \neq \frac{\partial \text{metric}(m^{i,j})}{\partial m_{k,l}} \) with \( (i, j, k, l) \in (C \times C \times C \times C) \) and \( i \neq j, k \neq l \), then metric is not a micro metric.

A ‘macro’ metric, in contrast to a ‘micro’ metric, is sensitive to class labels.

III Macro average: It’s a mean over classes

‘Macro’ metrics are sometimes named ‘macro-average’ metrics, which indicates that they may be perceived as an average over classes. We express this as

**Property III (Class decomposability).** A ‘macro-average’ metric can be stated as

\[
\text{metric}(m, g, p) = \left( \frac{1}{n} \sum_{i=1}^n g(m_i, (m^T)_i)^p \right)^\frac{1}{p}
\]

I.e., if our metric can be defined as an unweighted generalized mean over classes using a ‘local’ metric \( g \) that makes use of all instances related to a specific class \( \text{class} = i \) or \( f \to i \), then we say that it has PIII.

IV More strict: “Treat all classes equally”

A common argument for using metrics other than the rate of correct predictions is that a metric should show ‘classifier performance equally w.r.t. all classes’, or ’does not neglect rare classes’ (e.g., see [16, 7, 2, 13]).

Such a metric promises better extrapolation of the measured performance to data sets with different class frequencies. At first glance, PIII seems to imply such a feature already, since we compute an unweighted mean over classes. However, the score w.r.t. one class is (potentially) coupled to the prevalence of other classes.

So it makes sense to define such an expectation (‘treat all classes equally’) most strictly. We simulate different class prevalences with a

**Prevalence scaling.** We can use a diagonal prevalence scaling matrix \( \lambda \) to set

\[
m' = m \cdot \lambda.
\]

By scaling a column \( i \) with \( \lambda_{ii} \), we inflate (or deflate) the mass of data that belong to class \( i \) (e.g., see Tables 2, 3, 4), but retain the relative proportions of intra-class error types. Now, we can define
A simple but thorough primer on metrics for multi-class evaluation such as Micro F1, Macro F1, Kappa and MCC

There are two well-known chance ‘baseline’ classifiers: One predicts labels uniformly at random, the other always desired, since it does not correct for accuracy.

**Property IV** (Prevalence invariance). If \((\lambda, \lambda') \in \mathbb{R}^n_{>0} \times \mathbb{R}^n_{>0}\) is a pair of diagonal matrices then \(\text{metric}(m\lambda) = \text{metric}(m\lambda')\).

PIV makes a metric invariant to class prevalence.

**Prevalence calibration** There is an interesting special case of \(\lambda\). We can select \(\lambda\) s.t. all classes have the same prevalence. We call this prevalence calibration:

\[
\lambda_{ii} = \frac{1}{n \cdot \text{prevalence}(i)},
\]

and achieve equal class prevalence (without changing the proportions of intra-class error types).

### V Chance correction

There are two well-known chance ‘baseline’ classifiers: One predicts labels uniformly at random, the other based on prevalence. More arbitrary chance classifiers exist. The interpretability of a macro metric score increases if the score of the best chance classifier is not more than a function of the number of classes \(n\). This indicates robustness against chance predictions and, in addition, makes the measurement of a single classifier meaningful by providing a chance baseline score.

**Property V** (Chance Correction). A metric has this property, if, for a dataset \(S\) with \(n\) classes and a set \(A\) that contains arbitrary random classifiers:

\[
\max\{\text{metric}(m^{S'}) \mid r \in A\} = c(n),
\]

where function \(c\) provides a score that is based on the number of classes \(n\) alone. In the special case of \(\max(\text{metric}(m^{S'})) : r \in A\) = \(\min(\text{metric}(m^{S'})) : r \in A\), we say that \(\text{metric}\) is strictly chance corrected, and in the special case that the latter holds for all data set pairs \(S, S'\), we say that \(\text{metric}\) is completely chance corrected. Note that strictness or completeness isn’t always desired, since it does not correct for accuracy.

## 4 Metrics: overview and analysis

### Accuracy (aka Micro Recall aka Micro Precision aka Micro F1)

Before studying other metrics, we briefly view accuracy that tells us the ratio of correct predictions:

\[
\text{accuracy} = \frac{\sum_i m_{ii}}{\sum_{(i,j)} m_{i,j}} = \frac{\sum_i \text{correct}(i)}{\sum_i \text{prevalence}(i)}. \tag{7}
\]

It is equivalent to ‘micro Prec., micro Recall and micro F1’ that are often shown in papers (c.f. Appendix A).

**Property analysis and discussion** We easily see that it has only PI (monotonicity). This is expected, since PII-V tend to target macro metrics.

In general, accuracy is a key evaluation statistic, approximating the probability to observe a correct prediction. However, clearly the metric is strictly tied to the class prevalences that occur in a specific data set, and researchers seem interested in other metrics.

### Macro recall: ticks all boxes

Macro recall is calculated as the unweighted arithmetic mean over all class-wise recall scores:

\[
\text{macR} = \frac{1}{n} \sum_i R_i. \tag{8}
\]

**Property analysis** Macro Recall has all five properties (Proofs in Appendix B). We also note that it is strictly chance corrected with \(c(n) = 1/n\).

**Discussion** Since macro Recall has all five properties, it is a useful macro metric for transparent and meaningful classifier evaluation. Moreover, it offers three intuitive interpretations: Drawing items from class bags, Bookmaker metric and prevalence-calibrated accuracy.

In the first interpretation, we are given \(n\) random items, one from every class. We select a random one and ask: what’s the probability that it is correctly predicted? \(\text{MacR}\) knows the answer.

Macro Recall also has another interesting interpretation, since it can be viewed as a (fair) Bookmaker’s
metric. For every prediction, we pay 1 unit and gain units according to fair (European) odds. The odds for making a correct prediction, when the true class is \(i\), are 

\[ \text{odds}(i) = \frac{\text{odds}(s_i)}{\text{prevalence}(s_i)} \]

For each data example, our classifier \(f\) makes a bet, incurring a total

\[
gain = \sum_{s \in S} \left( [f(s_i) = s_j] \cdot \text{odds}(s_j) - 1 \right)
\]

\[
= -|S| + \sum_{s \in S} \left( [f(s_i) = s_j] \cdot \text{odds}(s_j) \right)
\]

\[
= -|S| + \sum_{i=1}^n \left( \text{odds}(i) \cdot \sum_{s \in S} [f(s) = i] \right)
\]

\[
= -|S| + |S| \sum_{i=1}^n \frac{\text{correct}(i)}{\text{prevalence}(i)}
\]

\[
= -|S| + n|S| \cdot \text{macR}
\]

Ignoring the cost and normalizing by the number of classes and data size \(|n|\), we obtain macro Recall. Finally, we can view macro Recall as an accuracy score after class prevalence calibration. To see this, first consider the standard accuracy measure (Eq. 7) and calibrate class prevalence (Eq. 6), observing that

\[
\text{macAcc} = \frac{\sum_i \lambda_{ii} \cdot \text{correct}(i)}{\sum_i \lambda_{ii} \cdot \text{prevalence}(i)} = \frac{\sum_i \lambda_{ii} \cdot \text{correct}(i)) = \text{macR}.}{\sum_i \lambda_{ii} \cdot \text{correct}(i)) = \text{macR}.}
\]

Macro precision: In the bias lies the issue

Macro precision is the unweighted arithmetic mean over the class-precision scores:

\[
\text{macP} = \frac{1}{n} \sum_i p_i
\]

Property analysis While properties I, II, III, V are fulfilled, macro Precision does not have prevalence invariance (Proofs in Appendix C). We can find theoretical \(m, \lambda\), where the maximum score difference \((\text{macP}(m) \text{ vs. macP}(\lambda))\) approaches \(1 - \frac{1}{n}\). Same as macro Recall, it is strictly chance corrected with \(c(n) = 1/n\)

Discussion Precision of class \(i\) approximates \(P(\text{class} = i |f \rightarrow i)\). Hence, at a glance, macP provides us with an overall measure of ‘prediction trustworthiness’, which may be valuable.

However, the issue is that \(\text{bias}(i)\) is a function of the nature of a classifier and the data \(\text{prevalence}(j), j \in C\). Therefore, even though it is decomposed over classes (PPII), it is not invariant to prevalence changes (PIV). If we have \(f, f'\) with different biases, score differences are difficult to interpret.

To mitigate this issue, we can make classifier bias more meaningful, by calibrating class prevalence (Eq. 6), controlling prevalence and (consequently) bias.

Macro F1: Metric of choice in many tasks

Macro F1 is often used for classifier evaluation. It is defined as an unweighted arithmetic mean over class-wise harmonic means of precision and recall:

\[
\text{macF1} = \frac{1}{n} \sum_i F_1(i) = \frac{1}{n} \sum_i \frac{2pR_i}{p_i + R_i}.
\]

with \(p_i, R_i, F_i = 0\) if the denominator is zero.

Property analysis Four properties are fulfilled (PI, PI, PIII, PV) while PIV is not fulfilled (Proofs in Appendix D). Interestingly, while macro F1 has PV (chance correction), the chance correction isn’t strict, differentiating it from most other macro metrics. In particular, its chance baseline upper-bound \(1/n\) is achieved when \(P(f \rightarrow i) = P(\text{class} = i)\), which means that macro F1 not only corrects for chance, but also for accuracy.

Additionally, we see that macro F1 is invariant to the false-positive and false-negative error spread for a specific class. This can be seen by writing:

\[
\text{macF1} = \frac{2}{n} \sum_i \frac{\text{correct}(i)}{\text{bias}(i) + \text{prevalence}(i)),}
\]

and viewing the denominator.

Discussion While the invariance to error types and the targeted balance between chance correction and accuracy seem useful, macro F1 inherits an issue of macro Precision: classifier bias to a class \(i\) is tied to class prevalence of all \(j, j \neq i\). Thus, in a strict sense, macro F1 does not ‘treat all classes equally’. The score may become more meaningful after class prevalence calibration (Eq. 6).

Macro F1: a doppelganger

Interestingly, there is another frequently used metric that has been coined ‘macro F1’ [12]. It is the harmonic mean of macro Precision and macro Recall:

\[
\text{macF1'} = \frac{2 \cdot \text{macP} \cdot \text{macR}}{\text{macR} + \text{macP}}.
\]

3 It is also set as default in evaluation reports of popular machine learning packages such as scikit-learn.
Property analysis In contrast to its name twin, one less property is fulfilled (PIII), since it cannot be decomposed over classes (Proofs in Appendix E), and it is strictly chance baseline corrected with \( c(n) = 1/n \). In [8], we provide more analyses of the two macro F1s.

Discussion On one hand, \( macF1' \) isn’t easy to interpret, since the numerator contains the cross-product of all class-wise recall and precision values. However, it can be viewed through the lens of an inter-annotator agreement (IAA) metric, where we do not compare a classifier against a reference, but instead we compare two reference candidates \( A \) and \( B \), which results in confusion matrices \( m_A = m \) and \( m_B = m^T \). Therefore:

\[
macF1' = \frac{2 \cdot macR(m_A) \cdot macR(m_B)}{macR(m_A) + macR(m_B)} \tag{21}
\]

falling back on \( macR \)’s clear interpretation(s).

Birds of a feather: Kappa and Matthews Correlation Coefficient

If we assume normalized confusion matrices\(^4\), we can state both metrics as concise as possible. First denote

\[
c = \sum_i correct(i) \quad b = m 1 \quad p = m^T 1, \tag{22}
\]

where \( c \) is the amount of correct predictions, \( p \) is a vector where at each index \( i \) we find\( prevalence(i) \), and \( b \) is a vector where at each index \( i \) we find\( bias(i) \).

Generalized Matthews correlation coefficient (MCC) The multi-class generalization of MCC [6] can now be written concisely as

\[
MCC = \frac{c - p^T b}{\sqrt{1 - b^T b} \sqrt{1 - p^T p}}. \tag{23}
\]

Cohen’s kappa Let us state Cohen’s kappa [3] as follows, to illuminate its similarity to MCC:

\[
KAPPA = \frac{c - p^T b}{1 - p^T b}. \tag{24}
\]

Property analysis MCC and KAPPA have PII and PV (complete chance baseline correction: \( c(n) = 0 \)). Interestingly, they are non-monotonic measures (PI). They also do not have PIII, PIV. Proofs are in Appendix F. Note also that \( MCC \geq KAPPA \) (since \( p^T b \leq b^T b, p^T p \)).

\(^4m_{ij} = \frac{1}{n} | \{ s \in \{ f(s_1) = i \land s_2 = j \} \} | \in [0,1], \sum m_{ij} = 1. \) This shows probabilities for error types and does not change the MCC or KAPPA score.

Discussion Kappa and MCC are similar measures. Since \( p^T b \approx \sum_i P(c = i) \cdot P(f \rightarrow i) \) is the probability of a random baseline classifier to correctly predict an item from a random class, and \( c \approx P(\text{correct}) \), KAPPA and MCC can be viewed as accuracy calibrated against a random chance baseline classifier.

However, overall they are calibrated in slightly different ways due to different denominators. The denominator of KAPPA simply shows the upper-bound w.r.t. the score of the perfect classifier, which is intuitive.

On the other hand, the denominator in MCC is harder to interpret due to its stronger dependence on classifier bias. This becomes evident when viewing the measures after class-prevalence calibration (Eq. 6):

\[
KAPPA(m\lambda) = \frac{c - n^{-1}}{1 - n^{-1}} \tag{25}
\]

\[
MCC(m\lambda) = \frac{c - n^{-1}}{(\sqrt{1 - b^T b})(\sqrt{1 - n^{-1}})}
\]

The influence of classifier biases can make it more difficult to meaningfully compare classifiers with MCC. By contrast, in KAPPA, the bias term is gone.

5 Discussion

Prevalence calibration of macro metrics

Macro Recall shows all five properties PI-V, and therefore offers good interpretability. However, researchers are also interested in using other macro metrics, such as macro Precision, or macro F1, to e.g., target a bias vs. prevalence balance over all classes (Eq. 19). To make the results of these metrics more meaningful, we can consider calibrating class prevalence such that all classes have the same prevalence (Eq. 5, Eq. 6). Such a calibration makes a classifier’s bias (i.e., Precision, F1) more meaningful and equips every metric with PIV.

Note that some popular software packages, e.g., scikit-learn, offer a calibration option, when creating confusion matrices from references and predictions.\(^5\)

More considerations

Other means in macro Recall We can use other \( p \) in the generalized mean (Eq. 4), besides \( p = 1 \) (arithmetic mean). E.g., we can use the geometric mean (\( p \rightarrow 0 \)):

\[
GmacR = GM(R_1, \ldots, R_n) = \sqrt[n]{R_1 \cdot \ldots \cdot R_n} \tag{25}
\]

\(^5\)from sklearn.metrics import confusion_matrix;

m = confusion_matrix(refs, preds, normalize='True')
A simple but thorough primer on metrics for multi-class evaluation such as Micro F1, Macro F1, Kappa and MCC

<table>
<thead>
<tr>
<th>metric</th>
<th>PI (mono.)</th>
<th>PII (class sens.)</th>
<th>PIII (decompose)</th>
<th>PIV (prev. invar.)</th>
<th>PV (chance correct)</th>
</tr>
</thead>
<tbody>
<tr>
<td>macro Recall</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓: 1/n, strict</td>
</tr>
<tr>
<td>macro Prec.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓: 1/n, strict</td>
</tr>
<tr>
<td>macro F1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓: 1/n, strict</td>
</tr>
<tr>
<td>macro F1'</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓: 1/n, strict</td>
</tr>
<tr>
<td>Kappa</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓: 0, complete</td>
</tr>
<tr>
<td>MCC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓: 0, complete</td>
</tr>
<tr>
<td>Accuracy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Macro Acc.</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Micro F1</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>same as accuracy (see Appendix A)</td>
</tr>
</tbody>
</table>

Table 5: Summary of evaluation metrics. (✓): a property is fulfilled after prevalence calibration.

Presenting class-wise recall scores helps to extrapolate metrics for new data We can estimate precision in another data set (under unknown class-prevalence and without reference), based on class-wise recall. First, we state an estimate of the class distribution \( P(\text{class}) \approx P(\text{class}) \) that can be expected. Then we estimate \( P(f) \approx P(f) \), by running the classifier on some data. Finally:

\[
P^*(\text{class} = i | f \rightarrow i) = \frac{P(f \rightarrow i | \text{class} = i) \cdot P(\text{class} = i)}{P(f \rightarrow i)} \tag{26}
\]

\[
\approx \frac{R_i \cdot P(\text{class} = i)}{P(f \rightarrow i)} \tag{27}
\]

From here, estimated scores of macro metrics, including MCC and KAPPA, easily follow. Note that it is not possible to approximate expected recall values on new data by providing precision values on old data (since these do not transfer). Hence, this corroborates the value of recall statistics.

6 Conclusion

Table 5 shows an overview of the visited metrics.

We make some observations: i) only macro Recall has all five properties (PI-PV). Given \( n \) random items, one from every class, \( GmacR \) approximates a (class-count normalized) probability that all are correctly predicted.

\( GmacR \) can be useful when it is important to have good performance for all classes.

How do metrics behave in a practical evaluation setting? So far, we mainly focused on theory. In an empirical setting, differences between most metrics are unlikely to be as extreme as in hypothetically constructed cases. However, when studying results of a shared task on document classification, we can observe important differences in the rankings of classifiers (c.f. Appendix G). This underlines the importance of understanding metrics, to inform metric selection and achieve meaningful classifier rankings.

References


A simple but thorough primer on metrics for multi-class evaluation such as Micro F1, Macro F1, Kappa and MCC

A Accuracy aka micro Precision/Recall/F1

Micro F1 is defined as the harmonic mean (HM) of ‘micro Precision’ and ‘micro Recall’, where micro Precision is

$$\frac{\sum_x \text{correct}(x)}{\sum_x \text{bias}(x)}$$

and micro Recall is

$$\frac{\sum_x \text{correct}(x)}{\sum_x \text{prevalence}(x)}.$$  

Now it suffices to see that \( \sum_x \text{prevalence}(x) = \sum_x \text{bias}(x) = \sum_{i,j} m_{i,j} \), and \( \sum_x \text{correct}(x) = \sum_i m_{i,j} \), and \( HM(a, a) = a \).

6E.g., see [12]

Appendices

A Accuracy aka micro Precision/Recall/F1

Micro F1 is defined as the harmonic mean (HM) of ‘micro Precision’ and ‘micro Recall’, where micro Precision is

$$\frac{\sum_x \text{correct}(x)}{\sum_x \text{bias}(x)}$$

and micro Recall is

$$\frac{\sum_x \text{correct}(x)}{\sum_x \text{prevalence}(x)}.$$  

Now it suffices to see that \( \sum_x \text{prevalence}(x) = \sum_x \text{bias}(x) = \sum_{i,j} m_{i,j} \), and \( \sum_x \text{correct}(x) = \sum_i m_{i,j} \), and \( HM(a, a) = a \).

6E.g., see [12]
A simple but thorough primer on metrics for multi-class evaluation such as Micro F1, Macro F1, Kappa and MCC

B  Macro Recall

B.1 Monotonicity ✓

If \( i \neq j \): \( \frac{\partial \text{macR}(m)}{\partial m_{i,j}} = -\frac{\text{correct}(i)}{n \cdot \text{prevalence}(i)^2} \leq 0 \); else \( \frac{\partial \text{macR}(m)}{\partial m_{i,j}} = \frac{\text{correct}(i)}{n \cdot \text{bias}(i)^2} \geq 0 \)

B.2 Class sensitivity ✓

Follows from above.

B.3 Class decomposability ✓

In Eq. 4 set \( g(\text{row}, \text{col}) = \frac{\text{row}_i}{\sum \text{col}_i} \) and \( p = 1 \).

B.4 Prevalence invariance ✓

\( R'_i = \frac{\lambda_i \cdot m_{i,j}}{\lambda_i \cdot m_{j,i}} = \frac{\lambda_i \cdot m_{i,j}}{\lambda_i \cdot m_{j,i}} = R_i \).

B.5 Chance correction ✓

Assume normalized class prevalences \( p \in [0,1]^n \) s.t. \( \sum_{i=1}^n z_i = 1 \) and arbitrary random baseline \( z \in [0,1]^n \) s.t. \( \sum_{i=1}^n z_i = 1 \):

\[
\text{MacR} = \frac{1}{n} \sum_i R_i = \frac{1}{n} \sum_i \frac{p_i \cdot z_i}{n \cdot \text{bias}_i} = \frac{1}{n} \sum_i z_i = 1
\]  

(C30)

C  Macro precision

C.1 Monotonicity ✓

If \( i \neq j \): \( \frac{\partial \text{macP}(m)}{\partial m_{i,j}} = -\frac{\text{correct}(i)}{n \cdot \text{bias}(i)^2} \leq 0 \); else \( \frac{\partial \text{macP}(m)}{\partial m_{i,j}} = \frac{\text{correct}(i)}{n \cdot \text{bias}(i)^2} \geq 0 \)

C.2 Class sensitivity ✓

Follows from above.

C.3 Class decomposability ✓

In Eq. 4 set \( g(\text{row}, \text{col}) = \frac{\text{row}_i}{\sum \text{row}_i} \) and \( p = 1 \).

C.4 Prevalence invariance ✓

A counter-example \( P'_i = \frac{\lambda_i \cdot m_{i,j}}{\lambda_i \cdot m_{j,i}} \neq P_i \) is easily found.

E.g., in Table 2, 3, 4: \( \text{macP} = 0.5 \frac{3}{4} + 0.5 \frac{1}{2} = \frac{5}{8} \neq \text{macP}' = 0.5 \frac{3}{5} + 0.5 \frac{1}{2} = \frac{19}{20} \).

C.5 chance correction ✓

Assume normalized class prevalences \( p \in [0,1]^n \) s.t. \( \sum_{i=1}^n z_i = 1 \) and arbitrary random baseline \( z \in [0,1]^n \) s.t. \( \sum_{i=1}^n z_i = 1 \):

\[
\text{MacP} = \frac{1}{n} \sum_i R_i = \frac{1}{n} \sum_i \frac{p_i \cdot z_i}{n \cdot \text{bias}_i} = \frac{1}{n} \sum_i p_i = \frac{1}{n}
\]  

(31)

D  Macro F1

D.1 Monotonicity ✓

Let \( Z_i = \text{bias}(i) + \text{prevalence}(i) \). If \( i \neq j \):

\[
\frac{\partial \text{macF1}(m)}{\partial m_{i,j}} = -\frac{2 \cdot \text{correct}(i)}{n Z_i^2} - \frac{2 \cdot \text{correct}(j)}{n Z_j^2} \leq 0
\]  

else:

\[
\frac{\partial \text{macF1}(m)}{\partial m_{i,j}} = \frac{2}{n Z_i} - \frac{2 \cdot \text{correct}(i)}{n Z_i^2} + \frac{2}{n Z_j} - \frac{2 \cdot \text{correct}(j)}{n Z_j^2} \geq 0
\]  

(32)

D.2 Class sensitivity ✓

Follows from above.

D.3 Class decomposability ✓

In Eq. 4 set \( p = 1 \), \( g(\text{row}, \text{col}) = \frac{\text{row}_i}{\sum \text{row}_i + \text{col}_i} \).

D.4 Prevalence invariance ✓

By counter-example, similar to (C.4).

D.5 chance correction ✓

Assume normalized class prevalences \( p \in [0,1]^n \) s.t. \( \sum_{i=1}^n z_i = 1 \) and arbitrary random baseline \( z \in [0,1]^n \) s.t. \( \sum_{i=1}^n z_i = 1 \). We have

\[
\text{MacF1} = \frac{1}{n} \sum_i \frac{2 \cdot p_i \cdot z_i}{p_i + z_i} = \frac{1}{n} \sum_i \frac{2 \cdot p_i \cdot z_i}{p_i + z_i}
\]  

(35)

We see that a maximum is attained when \( p = z \), and that this maximum is \( \frac{1}{n} \).
E  Macro F1 (name twin)

E.1  Monotonicity ✓

We have \( \frac{\partial \text{macF}_1(m)}{\partial m_{i,j}} = \frac{2x}{\text{macR} \cdot \text{macP}} \) where \( x = \)

\[
\left( \frac{\partial \text{macR}(m)}{\partial m_{i,j}} + \frac{\partial \text{macP}(m)}{\partial m_{i,j}} \right) \cdot \left( \text{macR} + \text{macP} - \text{macP} \cdot \text{macR} \right)
\]

(36)

Since \( \text{macR} \) and \( \text{macP} \) have monotonicity and (37) \( \geq 0 \), macF1’ also has monotonicity.

E.2  Label sensitivity ✓

Follows from above.

E.3  Class decomposability

Not possible.

E.4  Prevalence invariance

See \( \text{macP} \).

E.5  chance correction ✓

Since \( \text{macF1}' \) is the harmonic mean from (strictly chance corrected) macro precision and macro recall, we also have strictly chance correction with \( \frac{1}{n} \).

F  KAPPA and MCC

F.1  Monotonicity

Kappa  Let us state

\[
\text{KAPPA} = \frac{cs - p^T b}{s^2 - b^T b} = \frac{N_K}{D_M}
\]

(38)

with \( s = \sum_{i,j} m_{i,j} \). Other variables were introduced before (Eq. 22). Now, let \( z_{ij} = \text{bias}(j) + \text{prevalence}(i) \).

Then, iff \( i \neq j \):

\[
\frac{\partial \text{KAPPA}}{\partial m_{i,j}} = \frac{(c - z_{ij})D_K^2 - (2s - z_{ij})N_K}{D_K^2}
\]

(39)

\[
\sum_i p_i \cdot z_i - \sum_i \left( \sum_j z_j \cdot p_j \right) \left( \sum_j z_j \cdot p_i \right) = 0
\]

(44)
A simple but thorough primer on metrics for multi-class evaluation such as Micro F1, Macro F1, Kappa and MCC

<table>
<thead>
<tr>
<th>sys</th>
<th>macR</th>
<th>GmacR</th>
<th>HmacR</th>
<th>macF1</th>
<th>kappa</th>
<th>mcc</th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>68.1</td>
<td>66.8</td>
<td>65.4</td>
<td>65.4</td>
<td>67.7</td>
<td>46.5</td>
<td>48.0</td>
<td>45.8</td>
<td>51.2</td>
</tr>
<tr>
<td>B</td>
<td>68.1</td>
<td>66.5</td>
<td>65.0</td>
<td>66.0</td>
<td>67.7</td>
<td>47.3</td>
<td>49.2</td>
<td>49.2</td>
<td>87.8</td>
</tr>
<tr>
<td>C</td>
<td>67.6</td>
<td>66.8</td>
<td>66.0</td>
<td>66.0</td>
<td>67.5</td>
<td>47.2</td>
<td>48.1</td>
<td>56.0</td>
<td>65.2</td>
</tr>
<tr>
<td>D</td>
<td>67.4</td>
<td>66.5</td>
<td>65.6</td>
<td>65.1</td>
<td>67.1</td>
<td>46.3</td>
<td>47.3</td>
<td>54.2</td>
<td>67.6</td>
</tr>
<tr>
<td>E</td>
<td>66.9</td>
<td>66.8</td>
<td>66.6</td>
<td>66.0</td>
<td>67.3</td>
<td>47.0</td>
<td>47.0</td>
<td>69.8</td>
<td>64.0</td>
</tr>
<tr>
<td>F</td>
<td>65.9</td>
<td>65.6</td>
<td>65.4</td>
<td>64.5</td>
<td>66.1</td>
<td>45.0</td>
<td>45.4</td>
<td>58.7</td>
<td>65.6</td>
</tr>
<tr>
<td>G</td>
<td>64.9</td>
<td>64.2</td>
<td>63.5</td>
<td>63.4</td>
<td>64.9</td>
<td>43.0</td>
<td>43.8</td>
<td>53.9</td>
<td>63.5</td>
</tr>
<tr>
<td>H</td>
<td>64.5</td>
<td>64.5</td>
<td>63.7</td>
<td>64.9</td>
<td>43.6</td>
<td>43.6</td>
<td>65.3</td>
<td>63.6</td>
<td>64.5</td>
</tr>
</tbody>
</table>

Table 8: Shared task ranking with different metrics. r_i: recall for class i.

Figure 2: Team ranking correlation matrix with paired metrics. metric_c means that the confusion matrix has been calibrated before metric computation (Eq. 6).

G Small empirical study

We use metrics to rank teams that participated in the Semeval-2017 task [11] to predict the sentiment of tweet documents: negative, neutral, or positive.

The two winning systems were determined with macR, which is fair (A, B, Table 8). Yet, system E also does quite well, because it achieves a better balance over the three classes ($R_1 = 69.8, R_2 = 64.0, R_3 = 66.8, \text{max.} \Delta = 5.8$) as opposed to, e.g., system B ($R_1 = 87.8, R_2 = 51.4, R_3 = 65.2 \text{ max.} \Delta = 36.4$), and indicated also by GmacR and HmacR metrics. Hence, if a user wants to ensure that all classes are well predicted under high uncertainty of prevalence (perhaps to be expected in Twitter data?), they may prefer system E.

Figure 2 shows a pair-wise Spearman correlation of team rankings. When considering the same metric before and after prevalence calibration, we see that only recall metrics agree with each other in their rankings. In particular, as indicated before, it seems that the second class is the one that may tip the scale: $R_2$ observably disagrees in its team ranking with all other metrics (Spearman's $\rho \leq 14$). Furthermore, KAPPA assigns the exact ranks as macR, but only after prevalence calibration (Spearman's $\rho$ Kappa, vs. macR: 100).