## **About Measure Metrics**



## **1 WHY MCC?**

Although the *F-measure* and *AUC* are widely used, we see them as problematic due to bias particularly in the presence of unbalanced data sets, which is of course precisely the scenario we are interested in studying. Consequently, we use *MCC* (Matthews correlation coefficient [4] (MCC) otherwise known as  $\phi$  - see [5]) as our measure of predictive performance.

	Actually Positive	Actually Negative
Predict Positive	TP	FP
Predict Negative	FN	TN

## TABLE 1: Confusion Matrix

The starting point for most classification performance measures is the confusion matrix. This represents counts of the four possible outcomes when using a dichotomous classifier to make a prediction (see Table 1)<sup>1</sup>. For example,  $F_1$  is the most commonly used derivative of the *F*-measure family and is defined by Eqn. 1.

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{1}$$

However, it excludes True Negatives (*TN*) in its calculation which is potentially problematic. The reason is that it originated from the information retrieval domain where typically the number of true negatives, e.g., irrelevant web pages that are correctly not returned is neither knowable nor interesting. However, unlike recommendation tasks<sup>2</sup>, this is not so for defect prediction because test managers are definitely interested to know if components are truly non-defective.

Let us compare  $F_1$  with MCC. MCC is the geometric mean of the regression coefficients of the problem and its dual [1] and is defined as:

$$MCC = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$
(2)

As a correlation coefficient it measures the relationship between the predicted class and actual class, *MCC* is on a scale [-1,1] where 1 is a perfect positive correlation (also perfect prediction), zero no association and -1 a perfect negative correlation. In contrast, we illustrate the problematic nature of  $F_1$  with a simple example and compare it with *MCC*.

Suppose our hypothetical defect classifier predicts as Case 1 in Table 2.

	Case 1	Case 2	Case 3	Case 4
TP	5	5	4	9
FP	45	45	9	54
FN	5	5	6	1
TN	45	0	81	36
$F_1$	0.17	0.17	0.35	0.25
MCC	0.00	-0.67	0.27	0.19
G-mean	0.50	0.00	0.60	0.60

TABLE 2: Example Classification Cases

We can see the proportion of cases correctly classified is 0.5 i.e., TP+TN/n = 5+45/100. This yields an  $F_1$  of 0.17 on a scale [0,1] which is somewhat difficult to interpret. Let us compare  $F_1$  with *MCC*. In this case, *MCC*=0 which is intuitively reasonable since there is no association between predicted and actual<sup>3</sup>. Now suppose the True Negatives are removed so n=55 as in Case 2 in Table 2.  $F_1$  remains unchanged at 0.17 whilst *MCC*=-0.67 signifying substantially worse than random performance. The proportion of correctly classified cases is now 5/55 = 0.09, clearly a great deal worse than guessing and so we have a perverse classifier. However,  $F_1$  cannot differentiate between the two situations. This means experimental analysis based upon  $F_1$  would be indifferent to the two outcomes.

This example illustrates not only a drawback with  $F_1$ , but also the weakness of all derivative measures from *Recall* and *Precision* as they ignore TNs. Measures such as *Accuracy* are also well-known to be biased as they are sensitive to data distributions and the prevalence of the positive class [8].

One alternative measure that covers the whole confusion matrix is the *G-mean*, defined as the geometric mean of the accuracies of the two classes (see Eqn. 3) and was developed specifically for assessing the performance under imbalanced domains [3]. It assumes equal weight of the precision for both classes.

$$G\text{-mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}$$
(3)

However, there are disadvantages with *G-mean*. As observed by López et al.[2], "due to this symmetric nature of the geometric mean ... it is hard to contrast different models according to their precision on each class". For example, in Table 2 we observe that Case 3 ( $TP_{rate} = 0.4$ ,  $TN_{rate} = 0.9$ )

<sup>1.</sup> Note: in the context of software defect prediction, the positive class and negative class denote defective and non-defective respectively.

<sup>2.</sup> Examples of recommendation tasks include bug triage [6] or recommending code snippets [7].

<sup>3.</sup> This is a typical random guess where the accuracy for both classes is 50%.

and Case 4 ( $TP_{rate} = 0.9$ ,  $TN_{rate} = 0.4$ ) the G-mean is the same 0.60. However, Case 3 is clearly preferred by MCC and F1. An alternative version called the G-measure replaces  $TN_{rate}$  with *precision*, however, it ignores TN and suffers the same drawback as the F-measure.

Thus, we seek a single measure that:

- 1) Covers the entire confusion matrix;
- 2) Evaluates a specific classifier<sup>4</sup>;
- Properly takes into account the underlying frequencies of true and negative cases;
- 4) can be easily interpreted

The third requirement needs further discussion in that AUC — another commonly used measure for evaluating classifiers — is also problematic. AUC calculates the area under an ROC curve which depicts relative trade-offs between TPR (true positive rate which is TP/(TP+FN)) and FPR (false positive rate which is FP(FP+TN)) of classification for every possible threshold. One classifier can only be preferred to another if it strictly dominates i.e., every point on the ROC curve of this classifier is above the other curve. Otherwise, we cannot definitively determine which classifier is to be preferred since it will depend upon the relative costs of FPs and FNs.

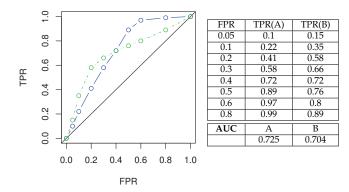


Fig. 1: ROC curves of Classifier A (the solid curve) and Classifier B (the dotdash curve)

TABLE 3: Points on the A and B ROC curves

Consider the example in Fig. 1 that shows ROC curves for two classifiers (Classifier Family A and Classifier Family B) derived from the values of some points on these curves (Table 3). We can observe that B is better than A when FPR is less than 0.4, but this reverses when FPR is greater than 0.4. Without knowing the relative costs of FP and FN we cannot determine which classifier is to be preferred. As a compromise, the area under the curve can be calculated to quantify the overall performance of classifier families, i.e. the AUC of A is 0.725 which is greater than the AUC of B (0.704). The AUC values indicate A is better than B, but this still doesn't help us determine which *specific* classifier we should actually choose.

Moreover, AUC is incoherent in that it is calculated on different misclassification cost distributions for different classifiers [10], since various thresholds relate to varying misclassification costs. Hence we conclude AUC is unsuitable for our purposes. Consequently, we select *MCC* as our performance measure. For a fuller discussion of the merits and demerits of various classification performance metrics see [1], [11], [3].

## REFERENCES

- D. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
- [2] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.
- [3] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," ACM Computing Surveys (CSUR), vol. 49, no. 2:31, 2016.
- [4] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.
- [5] M. Warrens, "On association coefficients for 2 × 2 tables and properties that do not depend on the marginal distributions," *Psychometrika*, vol. 73, no. 4, pp. 777–789, 2008.
  [6] J. Xuan, H. Jiang, Y. Hu, Z. Ren, W. Zou, Z. Luo, and X. Wu,
- [6] J. Xuan, H. Jiang, Y. Hu, Z. Ren, W. Zou, Z. Luo, and X. Wu, "Towards effective bug triage with software data reduction techniques," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 264–280, 2015.
- [7] H. Jiang, L. Nie, Z. Sun, Z. Ren, W. Kong, T. Zhang, and X. Luo, "Rosf: Leveraging information retrieval and supervised learning for recommending code snippets," *IEEE Transactions on Services Computing*, 2016.
- [8] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms." in *ICML*, vol. 98, 1998, pp. 445–453.
- [9] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [10] D. Hand, "Measuring classifier performance: a coherent alternative to the area under the ROC curve," *Machine Learning*, vol. 77, no. 1, pp. 103–123, 2009.
- [11] P. Flach and M. Kull, "Precision-recall-gain curves: PR analysis done right," in Advances in Neural Information Processing Systems, 2015, pp. 838–846.

<sup>4.</sup> As opposed to a family of classifiers such as is the case for the Area Under the Curve (AUC) measure [9]