ASSESSING BINARY CLASSIFIERS

PRECISION, RECALL, ROC CURVE

By

0xLeo (github.com/0xleo)

September 5, 2019

DRAFT X.Y
Missing:

Contents

1 Measuring the quality of a binary classifier

The problem we attempt to answer in this article is; when is a binary classifier reliable given a dataset? What's a good quality measure? From now on, we assume that the classifier can predict only two classes - *positive* and *negative*.

1.1 Binary classification terms – actual vs predicted

We assume that we have a number of positive and negative samples as input as attempt to predict (label) them as positive or negative. Therefore we have 4 prediction cases:

1. Positive input and positive prediction – True Positive (TP). Good.
2. Positive input and negative prediction – False Negative (FN). Bad.
3. Negative input and positive prediction – False Positive (FP). Bad.
4. Negative input and negative prediction – True Negative (TN). Good.

To make it easier to understand these terms, we can visualise them in a matrix called `confusion matrix`. It is split into four quarters.

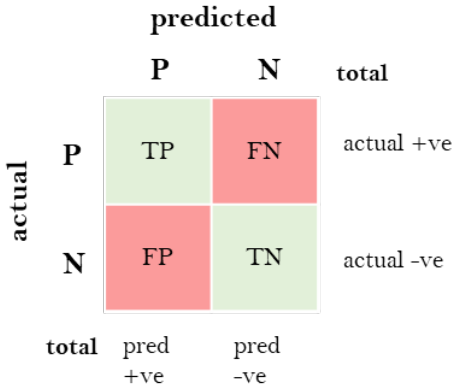

Fig. 1. Confusion matrix.

Ideally, a binary classifier labels all inputs 100% correctly, therefore outputs only TP's or TN's, but of course that's not always the case. What does the number of FP and FN compared to TP and TN say about the quality of its predictions?

1.2 Performance metrics

DEFINITION 1.1 (accuracy). *Accuracy is the ratio of correctly predicted samples over the total number of samples.*

$$Acc := \frac{TP + TN}{TP + FP + TN + FN} \tag{1.1}$$

Accuracy does not perform well with unbalanced datasets. For example, imagine we have a large image of N pixels and we want to detect a face in the distance by classifying whether each pixel corresponds to the face or not. The vast majority of pixels in the image are *not* face pixels (actual negatives) and a few are (actual positives). A badly-designed classifier could predict 0 face pixels. Therefore $TN \approx N$, $TP = 0$, $FP = 0$, $FN \approx 0$. Then the accuracy would falsely indicate that this is an almost perfect classifier as

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \approx \frac{N}{N} \approx 1$$

Precision aims to answer the question; "What proportion of positive predictions was actually correct?".

DEFINITION 1.2 (Precision). *Therefore precision is defined as the ratio of TP over the total predicted positives.*

$$Prec = \frac{TP}{TP + FP} \tag{1.2}$$

For the face detection example, precision would be the ratio of correctly predicted face pixels compared to all pixels predicted as face. Assume a badly-designed classifier correctly detected only a few face pixels but did not falsely detect background objects as face. Therefore it would have some TP, lots of TN, some FN and 0 FP. The precision would be perfect, although the classifier may not necessarily perform well:

$$Prec = \frac{TP}{TP + FP} = \frac{TP}{TP + 0} = 1$$

Precision itself is not a good metric and we don't know whether it is applied on all skin pixels or only a small sample of then – i.e. identify all relevant instances. The measure that identifies all relevant instances is the recall (a.k.a. sensitivity). Recall attempts to answer; "What proportion of actual positives was identified correctly?"

DEFINITION 1.3 (recall). *Therefore recall is defined as the ratio of TP over all the actuall positives (TP + FN):*

$$Rec = \frac{TP}{TP + FN} \tag{1.3}$$

Referring to the face detection example, if another bad classifier labelled all face pixels as face *and* all pixels around them as such, overestimating the face pixels, then $FN = 0$ and let $TP = s$. Then the recall would be perfect, although the classifier does not necessarily perform well:

$$Rec = \frac{TP}{TP + FN} = \frac{s}{s + 0} = 1$$

The latter classifier would correctly extract all relevant face pixel instances and have high recall. However it wouldn't correctly label all predicted positives, having lots of FP's, therefore low precision.

The figures below illustrate how recall and precision alone are not always good metrics.

Fig. 2. Face pixel classifier has detected nothing at all. However, $Acc \approx 100\%$.

Fig. 3. Detected face pixels are green. Classifier has failed to mark a lot of face pixels. $FP = 0$ therefore $Prec = 100\%$ but recall is low.

Fig. 4. Classifier has laballed a lot of background pixels as skin (FP). However, $FN = 0$ therefore $Rec = 100\%$ and precision is low.

A measure that combines both precision and recall is the $F1$ score.

DEFINITION 1.4 (f1 score). $F1$ *score is defined as the harmonic mean of precision and recall:*

$$F1 = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec} \tag{1.4}$$

$F1$ Score is best if there is some sort of balance between precision (p) and recall (r) in the system. Oppositely, $F1$ score isn't so high if one measure is improved at the expense of the other. For example, if $p = 1$, $r = 0$, $F1 = 0$.

1.3 When is a binary classifier accurate? The ROC curve.

A binary classifier is considered good when it performs better than the random classifier. We will see how we can compare a classifier to the random one in the newxt few paragraphs. One thing to keep in

mind about the random classifier is the following; let's say we have an input set with n_p positive and n_n negative instances. Then the actual positive probability is $p(X=1) = \frac{n_p}{n_p+n_n}$ and the actual negative is $p(X=0) = 1 - p(X=1)$. A random classifier (is random because) will randomly assign each input to the positive class with probability ρ and to the negative class with probablity $1-\rho$. Therefore for the random classifiers the numbers of TP, FN, TN, FP are the combined probabilities of the class and the assignment:

$$TP = \rho p(X=1) \tag{1.5}$$

$$FN = (1-\rho)p(X=1) \tag{1.6}$$

$$TN = (1-\rho)p(X=0) \tag{1.7}$$

$$FP = \rho p(X=0) \tag{1.8}$$

As shown in the following diagram, due to its nature a random classifier labels *equal fraction* ρ of positive instances as positive (TP/P) and negative instances as positive (FP/N).

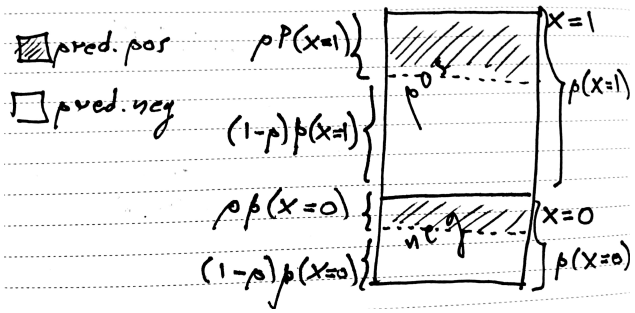

Fig. 5. How the random classifier's output depends on the class balance and the current decision (ρ).

The tool to measure a classifier's performance is the Receiver Operating Characteristic (ROC) curve. ROC curve uses some terms which we define below.

The name comes from WWII when it was used to evaluate the performance of radars!

DEFINITION 1.5 (sensitivity). *In ROC curve terms, recall is also called sensitivity or True Positive Rate (TPR), defined as we saw before as:*

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} \tag{1.9}$$

DEFINITION 1.6 (fall-out). *False positive rate (FPR) (or fall-out) is defined as:*

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} \tag{1.10}$$

In a ROC curve the true TPR (sensitivity) is plotted in function of the FPR for different cut-off points. Therefore to construct the whole curve we need to vary either the positive or the negative class probability in the input dataset from $P(X=1)=0$ to $P(X=1)=1$. Each point on the ROC curve represents a TPR/ FPR pair therefore a different confusion matrix instance. A test with perfect discrimination has a ROC curve that passes through the upper left corner (100% sensitivity – no FN, 0% fall-out – no FP). Therefore the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test.

As mentioned before, a classifier is good if it performs better than the random one. The figure below visualises some good and bad points in the ROC space. The "good" points are above the line $y = x$, which is the ROC curve of the random classifier.

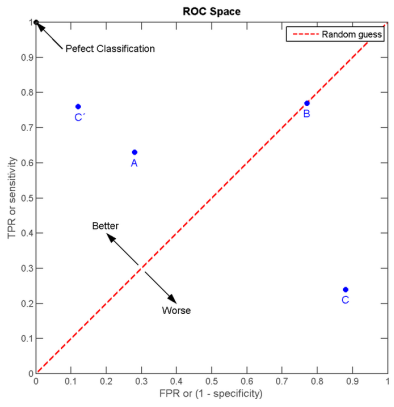

Fig. 6. The ROC space and plots of the four prediction examples. Source: wikepedia.

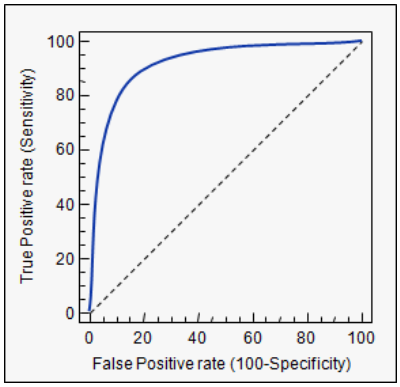

Fig. 7. Ideal ROC curve. It passes near point $(FPR, TPR) = (0,1)$

COROLLARY 1.1. *The ROC curve (TPR over FPR) of the random classifier is the line $y = x$.*

Proof. From Eq. (1.5) and Eq. (1.6), for the TPR of a random classifier we have:

$$TPR = \frac{TP}{TP+FN} = \frac{\rho p(X=1)}{\rho p(X=1) + (1-\rho)p(X=1)} = \rho \tag{1}$$

From Eq. (1.8) and Eq. (1.7) for the FPR:

$$FPR = \frac{FP}{FP+TN} = \frac{\rho p(X=0)}{\rho p(X=0) + (1-\rho)p(X=0)} = \rho \tag{2}$$

Therefore $FPR = TPN = \rho \;\; \forall \; 0 \leq \rho \leq 1$. This fact that these rates are both equal to ρ is also intuitive from Fig. 5.

□

1.4 Area Under ROC Curve (AUC)

The last question is how exactly do we measure the "quality" of the ROC curve, especially when two curves look similar? A robust measure we use to assess the overall performance of a binary classifier is the Area Under the ROC Curve (AUC). AUC is a robust measure as it relies on the complete ROC curve and thus involves all possible classification thresholds. It measures the area under the curve within points the box $(0,0),(1,1)$.

One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. AUC ranges in $[0,1]$. The higher the value, the better the classifier (Fig. 8). In general, AUC value in range $[0.5,1]$ are considered good as the AUC of a random classifier is 0.5. A AUC of that range means that our binary classifier in general performs better than a random one.

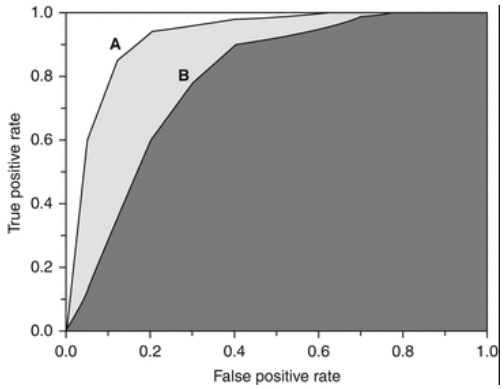

Fig. 8. Binary classifier A has higher AUC so it performs better [1].

COROLLARY 1.2. *A binary classifier performs well if it's AUC is higher than 0.5, where 0.5 is the AUC of a random classifier.*

References

[1] F. Melo, "Area under the roc curve," in *Encyclopedia of Systems Biology*, W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, Eds. New York, NY: Springer New York, 2013, pp. 38–39, ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_209. [Online]. Available: https://doi.org/10.1007/978-1-4419-9863-7_209.