Assessing model performance

Fraida Fund

February 3, 2020

Contents

In this lecture

- Performance metrics for classification
- Case study: COMPAS
- Fairness metrics for classification

Evaluating model performance

- Suppose we have a series of data points $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\}$ and there is some (unknown) relationship between $\mathbf{x_i}$ and y_i.
- We also have a black box *model* that, given some input $\mathbf{x_i}$, will each produce as its output an estimate of y_i, denoted $\widehat{y}_i$.
- The question we will consider in this lecture - without knowing any details of the model - is *how can we judge the performance of the estimator*?

Classifier performance metrics

Binary classifier performance metrics

Suppose in our example, the output variable y is constrained to be either a 0 or 1. The estimator is a *binary classifier*.

- a 1 label is considered a *positive* label.
- a 0 label is considered a *negative* label.

y is the actual outcome and $\widehat{y}$ is the predicted outcome.

Error types

A binary classifier may make two types of errors:

- Type 1 error (also called *false positive* or *false alarm*): Outputs $\widehat{y} = 1$ when $y = 0$.
- Type 2 error (also called *false negative* or *missed detection*): Output $\widehat{y} = 0$ when $y = 1$.

Confusion matrix

The number of *true positive* (TP) outputs, *true negative* (TN) outputs, false positive (FP) outputs, and false negative (FN) outputs, are often presented together in a *confusion matrix*:

Real $\downarrow$ Pred. $\rightarrow$	1	0
1	TP	FN
0	FP	TN

$P = TP + FN$, $N = FP + TN$

Accuracy

A simple performance metric, *accuracy*, is defined as

$$\frac{TP + TN}{TP + FP + TN + FN}$$

i.e., the portion of samples classified correctly.

Balanced accuracy

With imbalanced classes ($P >> N$ or $P << N$), we get good accuracy by "predicting" all 1 or all 0!
Balanced accuracy is more appropriate for highly imbalanced classes -

$$\frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

gives the proportion of correct predictions in each class, averaged across classes.

More binary classifier metrics (1)

- *True Positive Rate (TPR)* also called *recall* or *sensitivity*:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = P(\hat{y} = 1 | y = 1)$$

- *True Negative Rate (TNR)* also called *specificity*:

$$TNR = \frac{TN}{N} = \frac{TN}{FP + TN} = P(\hat{y} = 0 | y = 0)$$

More binary classifier metrics (2)

- *Positive Predictive Value (PPV)* also called *precision*:

$$PPV = \frac{TP}{TP + FP} = P(y = 1 | \hat{y} = 1)$$

- *Negative Predictive Value (NPV)*:

$$NPV = \frac{TN}{TN + FN} = P(y = 0 | \hat{y} = 0)$$

More binary classifier metrics (3)

- *False Positive Rate (FPR)*:

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR = P(\hat{y} = 1 | y = 0)$$

- *False Discovery Rate (FDR)*:

$$FDR = \frac{FP}{FP + TP} = 1 - PPV = P(y = 0|\hat{y} = 1)$$

More binary classifier metrics (4)

- *False Negative Rate (FNR)*:

$$FNR = \frac{FN}{FN + TP} = 1 - TPR = P(\hat{y} = 0|y = 1)$$

- *False Omission Rate (FOR)*:

$$FOR = \frac{FN}{FN + TN} = 1 - TPR = P(y = 1|\hat{y} = 0)$$

Summary of binary classifier metrics

Selected classifier metrics

F1 score

Combines precision ($\frac{TP}{TP+FP}$) and recall ($\frac{TP}{TP+FN}$) in one metric:

$$F_1 = 2\left(\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}\right)$$

Which metric?

Consider

- class balance
- relative cost of each kind of error

Example: identifying key metrics

Imagine a classifier for non-invasive prenatal testing that analyzes blood samples of pregnant women, to:

- Identify whether the fetus is a boy or a girl.
- Identify women that should undergo more invasive diagnostic tests for possible fetal health problems.

Soft decisions and thresholds

Some classifiers give *soft* decisions:

- **Hard decision**: output is either a 0 or 1
- **Soft decision**: output is a probability, $P(y = 1|\mathbf{x})$

We get a "hard" label from a "soft" classifier by setting a threshold: $\hat{y} = 1$ if we estimate $P(y = 1|\mathbf{x}) > t$ for some threshold t.

Soft decisions and performance metrics

With a threshold, we can get a confusion matrix and compute the other performance metrics - but these all depend on choice of t.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.DataFrame({'x': [1,2,3,4,5,6,7,8,9,10],
    'True y': [0,0,0,0,0,1,1,1,1,1],
    'Probability Estimate': [0.1, 0.24, 0.16, 0.52, 0.44, 0.45, 0.61, 0.81, 0.73, 0.9]})

sns.scatterplot(data=df, x='x', y='Probability Estimate', hue='True y')
plt.axhline(y=0.3, xmin=0, xmax=1, color='gray')
plt.axhline(y=0.5, xmin=0, xmax=1, color='gray')
plt.axhline(y=0.7, xmin=0, xmax=1, color='gray')
plt.savefig('images/threshold.svg')
```

Metrics depend on threshold

We could set t to maximize overall accuracy, set it higher to decrease FPR (but also decrease TPR), or set it lower to increase TPR (but also include FPR).

ROC curve

The *ROC curve* shows tradeoff between FPR and TPR for a specific *classifier* with varying t

- Each point shows the FPR and TPR of the classifier for a different value of t
- Plot FPR on x-axis, TPR on y-axis

(*ROC* stands for receiver operating characteristic" - the term is from radar applications.)

ROC curve example

ROC curve - via bu.edu

AUC

Area under the [ROC] curve (AUC) is a performance metric for the overall classifier, independent of t

- Higher AUC is better
- Higher AUC means for a given FPR, it has higher TPR

Multi-class classifier performance metrics

Output variable $y \in 1, 2, \cdots, K$

- Accuracy: number of correct labels, divided by number of samples
- Balanced accuracy: direct extension of two-class version
- Other metrics: pairwise comparisons between one class and all others

Soft classifier: probability for each class.

Multi-class confusion matrix

Example via Cross Validated

Using `scikit-learn` to compute metrics

The `scikit-learn` library in Python includes functions to compute many performance metrics.

For reference, you can find these at: scikit-learn metrics.

Function definitions

```
sklearn.metrics.accuracy_score(y_true, y_pred,
    normalize=True, sample_weight=None, ...)
```

Function calls

```
from sklearn import metrics

# assuming you have the vectors y_true and y_pred...
acc = metrics.accuracy(y_true, y_pred)
```

What causes poor performance?

- Data (garbage in, garbage out)
- Variability in observations, not explained by features
- Incomplete coverage of the domain
- Model error: too simple, too complicated

Evaluating models - not just performance

- Cost/time for training and prediction
- Interpretability
- Fairness/bias

Bias in model output

Many potential *fairness* issues when ML models are used to make important decisions:

- ML used for graduate admissions
- ML used for hiring
- ML used to decide which patients should be admitted to hospital
- Even ML used to decide which ads to show people...

Bias in the ML lifecycle

- **Pre-existing**: exists independently of algorithm, has origins in society
- **Technical**: introduced or exacerbated by the technical properties of the ML system
- **Emergent**: arises due to context of use

(Source: [Professor Julia Stoyanovich @NYU](https://dataresponsibly.github.io/rds/assets/1_Intro.pdf))

Causes of bias

- Models trained with less data for minority group, are less accurate for that group
- Sampling issues: Street Bump example
- Inherent bias in society reflected in training data, carries through to ML predictions

- Target variable based on human judgment
- Lack of transparency exacerbates problem!

Fairness metrics

Suppose samples come from two groups: a and b

How can we tell whether the classifier treats both groups *fairly*?

Group fairness

(also called *statistical parity*). For groups a and b,

$$P(\hat{y} = 1|G = a) = P(\hat{y} = 1|G = b)$$

i.e. equal probability of positive classification.

Related: *Conditional statistical parity* (controlling for factor F)

$$P(\hat{y} = 1|G = a, F = f) = P(\hat{y} = 1|G = b, F = f)$$

Balance for positive/negative class

This is similar to *group fairness*, but it is for classifiers that produce soft output - applies to every probability S produced by the classifier.

The expected value of probability assigned by the classifier should be the same for both groups -

For positive class balance,

$$E(S|y = 1, G = a) = E(S|y = 1, G = b)$$

For negative class balance,

$$E(S|y = 0, G = a) = E(S|y = 0, G = b)$$

Predictive parity

(also called *outcome test*)

$$P(y = 1|\hat{y} = 1, G = a) = P(y = 1|\hat{y} = 1, G = b)$$

Groups have equal PPV. Also implies equal FDR:

$$P(y = 0|\hat{y} = 1, G = a) = P(y = 0|\hat{y} = 1, G = b)$$

The prediction should carry similar meaning (w.r.t. probability of positive outcome) for both groups.

Calibration

(also called *test fairness*, *matching conditional frequencies*).

This is similar to *predictive parity*, but it is for classifiers that produce soft output - applies to every probability S produced by the classifier.

$$P(y = 1|S = s, G = a) = P(y = 1|S = s, G = b)$$

Well-calibration extends this definition to add that the probability of positive outcome should actually be s:

$$P(y = 1|S = s, G = a) = P(y = 1|S = s, G = b) = s$$

False positive error rate balance

(also called *predictive equality*)

$$P(\hat{y} = 1|y = 0, G = a) = P(\hat{y} = 1|y = 0, G = b)$$

Groups have equal FPR. Also implies equal TNR:

$$P(\hat{y} = 0|y = 0, G = a) = P(\hat{y} = 0|y = 0, G = b)$$

False negative error rate balance

(also called *equal opportunity*)

$$P(\hat{y} = 0|y = 1, G = a) = P(\hat{y} = 0|y = 1, G = b)$$

Groups have equal FNR. Also implies equal TPR:

$$P(\hat{y} = 1|y = 1, G = a) = P(\hat{y} = 1|y = 1, G = b)$$

This is equivalent to group fairness **only** if the prevalence of positive result is the same among both groups.

Equalized odds

(also called *disparate mistreatment*)

$$P(\hat{y} = 0|y = i, G = a) = P(\hat{y} = 0|y = i, G = b), i \in 0, 1$$

Both groups should have equal TPR *and* FPR

Satisfying multiple fairness metrics

If the prevalence of (actual) positive result p is **different** between groups, then it is not possible to satisfy FP and FN *error rate balance* and *predictive parity* at the same time.

Conditional use accuracy equality

Groups have equal PPV *and* NPV

$$P(y = 1|\hat{y} = 1, G = a) = P(y = 1|\hat{y} = 1, G = b)$$

AND

$$P(y = 0|\hat{y} = 0, G = a) = P(y = 0|\hat{y} = 0, G = b)$$

Overall accuracy equality

Groups have equal overall accuracy

$$P(\hat{y} = y|G = a) = P((\hat{y} = y|G = b)$$

Treatment equality

Groups have equal ratio of FN to FP, $\frac{FN}{FP}$

Causal discrimination

Two samples that are identical w.r.t all features except group membership, should have same classification.

Fairness through unawareness

- Features related to group membership are not used in classification.
- Samples that are identical w.r.t all features except group membership, should have same classification.

Summary - model fairness

- A model can be biased with respect to age, race, gender, if those features are not used as input to the model.
- There are many measures of fairness, sometimes it is impossible to satisfy some combination of these simultaneously.
- People are not necessarily more fair.