Holistic AI × The Alan Turing Institute

Bias in Multiclass Classification Part III

Content by: Sachin Beepath, Giulio Filippi, Cristian Munoz, Roseline Polle, Nigel Kingsman, Sara Zannone

Speaker: Sara Zannone

Contents

III – Measuring Bias in Multiclass Classification

- 1) Introduce one Equality of Outcome Metric
- 2) Introduce one Equality of Opportunity Metric
- 3) Explain how the holisticai library can help in computing bias metrics in multiclass setting

Equality of Outcome Metric

Reminder - Frequency Matrix

- The frequency Matrix is a matrix indexed on groups and classes (shape $M \times N$) with the g, i entry being the proportion of group g that is allocated to class i.
- In equations, $FM_{gi} = P\left(Y_{pred} = i \middle| \mathcal{P} = g\right)$
- *All equality of outcome metrics start from the Frequency Matrix!*

Multiclass Statistical Parity (Step 1)

- We start by computing the frequency matrix.
- Recall that each row is a distribution (the allocation of a group to classes).
- We will usually need a way to compare these matrices row by row.
- In the most general setting, we can use any distance on distributions.
- In practice we use the total variation distance.
- In equations $d(g,h) = \frac{1}{2}\sum_i |FM_{gi} - FM_{hi}|$

Multiclass Statistical Parity (Step 1)

- We start by computing the frequency matrix.
- Recall that each row is a distribution (the allocation of a group to classes).
- We will usually need a way to compare these matrices row by row.
- In the most general setting, we can use any distance on distributions.
- In practice we use the total variation distance.
- In equations $d(g,h) = \frac{1}{2}\sum_i |FM_{gi} - FM_{hi}|$

Multiclass Statistical Parity (Step 2)

- Once we have the $\frac{M(M-1)}{2}$ distances $d(g,h)$ between group allocation distributions.
- We need a way of aggregating these scores
- Usually, we use a maximum to get an idea of the worst-case scenario.
- We can also use an average or a weighed average with weights being the importance of each pair of groups being similarly treated.

Multiclass Statistical Parity (Example)

- Consider the following example with $M = 3$ groups (A, B, C) and $\mathrm{N} = 3$ classes $(1,2,3)$.
- $Data = \begin{bmatrix} A & A & A & A & B & B & B & B & C & C \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \end{bmatrix}$
- We compute the Frequency Matrix
- $FM = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.5 & 0.25 \\ 0.5 & 0 & 0.5 \end{bmatrix}$

Multiclass Statistical Parity (Example)

- We then compute the distances between the rows of the confusion matrix
- $d(row1, row2) = 0.5(|0.5 - 0.25| + |0.25 - 0.5| + |0.25 - 0.25|) = 0.25$
- $d(row2, row3) = 0.5(|0.25 - 0.5| + |0.5 - 0| + |0.25 - 0.5|) = 0.5$
- $d(row1, row3) = 0.5(|0.5 - 0.5| + |0.25 - 0| + |0.25 - 0.5|) = 0.25$

Multiclass Statistical Parity (Example)

- With a maximum approach we get the following metric
- $\max(0.5, 0.25, 0.25) = 0.5$
- With a mean approach we get the following metric
- $\mathrm{m}ean(0.5, 0.25, 0.25) = \frac{1}{3}$

Equality of Opportunity Metric

Reminder - Conditional Confusion Matrices

- Recall the conditional confusion matrices for each group is defined as
- $CM_{ij}^{g} = P(Y_{pred} = i | Y_{true} = j, \mathcal{P} = g)$
- *All equality of opportunity metrics start from the Conditional Confusion Matrices (we will later call this the confusion tensor)!*

Reminder - Conditional Confusion Matrices

- Recall the conditional confusion matrices for each group is defined as
- $CM_{ij}^{g} = P(Y_{pred} = i | Y_{true} = j, \mathcal{P} = g)$
- *All equality of opportunity metrics start from the Conditional Confusion Matrices (we will later call this the confusion tensor)!*

Multiclass Equality of Odds (Step 1)

- The first step is computing the conditional confusion matrix of each group CM^g.
- Next we compute a pairwise distance between these matrices.
- We could define any distance we like. But in this case we will take a mean average deviation approach.
- In equations $\frac{1}{2N}\sum_{ij}\left|CM_{ij}^{g} - CM_{ij}^{h}\right| = d(g,h)$
- If different misclassifications $i \to j$ have different levels of importance, we can weigh this sum $\frac{1}{2N\sum_{ij} w(i,j)}\sum_{ij}\left|CM_{ij}^{g} - CM_{ij}^{h}\right| w(i,j) = d(g,h)$

Multiclass Equality of Odds (Step 1)

- The first step is computing the conditional confusion matrix of each group CM^g.
- Next we compute a pairwise distance between these matrices.
- We could define any distance we like. But in this case we will take a mean average deviation approach.
- In equations $\frac{1}{2N}\sum_{ij}\left|CM_{ij}^{g} - CM_{ij}^{h}\right| = d(g,h)$
- If different misclassifications $i \to j$ have different levels of importance, we can weigh this sum $\frac{1}{2N\sum_{ij} w(i,j)}\sum_{ij}\left|CM_{ij}^{g} - CM_{ij}^{h}\right| w(i,j) = d(g,h)$

Multiclass Equality of Odds (Step 1)

- The first step is computing the conditional confusion matrix of each group CM^g.
- Next we compute a pairwise distance between these matrices.
- We could define any distance we like. But in this case we will take a mean average deviation approach.
- In equations $\frac{1}{2N}\sum_{ij}\left|CM_{ij}^g - CM_{ij}^h\right| = d(g,h)$
- If different misclassifications $i \to j$ have different levels of importance, we can weigh this sum $\frac{1}{2N\sum_{ij} w(i,j)}\sum_{ij}\left|CM_{ij}^g - CM_{ij}^h\right| w(i,j) = d(g,h)$

Multiclass Equality of Odds (Step 2)

- Once we have the $\frac{M(M-1)}{2}$ distances $d(g, h)$ between between group conditional confusion matrices.
- We can aggregate them with a max approach to get an idea of the worst-case scenario.
- Or we can aggregate them with a mean.

Multiclass Equality of Odds (Example)

- Consider the following example with $M = 3$ groups (A, B, C) and $\mathrm{N} = 3$ classes $(1,2,3)$. In the following matrix, the first row is the groups, the second row is the predictions, and the third row is the true values.
- $Data = \begin{bmatrix} A & A & A & A & A & B & B & B & B & C & C & C \\ 0 & 1 & 2 & 0 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 & 2 & 1 & 0 & 2 & 0 & 2 & 1 & 0 \end{bmatrix}$
- The first step is computing the Conditional Confusion Matrices.
- $CM^A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2/3 & 0 \\ 0 & 1/3 & 0 \end{bmatrix}, CM^B = \begin{bmatrix} 0.5 & 0 & 1 \\ 0 & 1 & 0 \\ 0.5 & 0 & 0 \end{bmatrix}, CM^C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Multiclass Equality of Odds (Example)

- $CM^A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2/3 & 0 \\ 0 & 1/3 & 0 \end{bmatrix}, CM^B = \begin{bmatrix} 0.5 & 0 & 1 \\ 0 & 1 & 0 \\ 0.5 & 0 & 0 \end{bmatrix}, CM^C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- We take a traditional mean absolute deviation distance
- $d(A, B) = \frac{1}{6}\left(0.5 + 0.5 + \frac{1}{3} + \frac{1}{3}\right) = \frac{5}{24}$
- $d(B, C) = \frac{1}{6}(0.5 + 0.5 + 1 + 1 + 1 + 1) = \frac{5}{6}$
- $d(A, C) = \frac{1}{6}\left(1 + \frac{2}{3} + \frac{1}{3} + 1 + 1\right) = \frac{4}{6}$

Multiclass Equality of Odds (Example)

- $distances = [\frac{5}{24}, \frac{5}{6}, \frac{4}{6}]$
- If we take a max aggregation approach, we get $5/6$.
- If we take a mean aggregation approach, we get 0.57.

Metrics in Python

Installing the holisticai library

- Documentation for multiclass metrics can be found here https://holisticai.readthedocs.io/en/latest/metrics.html#multiclass-classification.
- First step is installing the library, this can be done via pip. The following is run in a jupyter cell.

```
!pip install holisticai
```

✓ 0.2s

Computing Bias Metrics with holisticai library

- Suppose we have some multiclass data and we would like to compute the frequency matrix / conditional confusion matrices.

```
import numpy as np
import pandas as pd
from holisticai.bias.metrics import frequency_matrix, confusion_tensor
```

✓ 0.2s

```
p_attr = np.array(['A', 'A', 'A', 'A', 'B', 'B', 'B', 'B', 'C', 'C'])
y_pred = np.array([0, 1, 2, 0, 1, 2, 0, 1, 2, 0])
y_true = np.array([0, 1, 1, 0, 1, 0, 2, 1, 2, 1])
```

✓ 0.2s

Computing Bias Metrics with holisticai library

- Suppose we have some multiclass data and we would like to compute the frequency matrix / conditional confusion matrices.

```
import numpy as np
import pandas as pd
from holisticai.bias.metrics import frequency_matrix, confusion_tensor
```

✓ 0.2s

```
p_attr = np.array(['A', 'A', 'A', 'A', 'B', 'B', 'B', 'B', 'C', 'C'])
y_pred = np.array([0, 1, 2, 0, 1, 2, 0, 1, 2, 0])
y_true = np.array([0, 1, 1, 0, 1, 0, 2, 1, 2, 1])
```

✓ 0.2s

Frequency Matrix with holisticai library

- We can compute the Frequency Matrix as follows.

```
frequency_matrix(p_attr, y_pred, normalize='class')
```

✓ 0.3s

	0	1	2
A	0.50	0.333333	0.333333
B	0.25	0.666667	0.333333
C	0.25	0.000000	0.333333

- Note that we add a normalize over class parameter. This is because there is also the option of normalising over group or None (no normalisation).

Confusion Tensor with holisticai library

- Note that what we previously called the conditional confusion matrices, is what we named the confusion tensor in the code.
- We give the option of outputting it as a tensor or as a multi-level pandas DataFrame.

```
confusion_tensor(p_attr, y_pred, y_true, as_tensor=True)
```

✓ 0.2s

```
array([[[2., 0., 0.],
        [0., 1., 0.],
        [0., 1., 0.]],

       [[0., 0., 1.],
        [0., 2., 0.],
        [1., 0., 0.]],

       [[0., 1., 0.],
        [0., 0., 0.],
        [0., 0., 1.]]])
```

```
confusion_tensor(p_attr, y_pred, y_true, as_tensor=False)
```

✓ 0.5s

	A			B			C		
	0	1	2	0	1	2	0	1	2
0	2.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
1	0.0	1.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0
2	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0

Multiclass Statistical Parity with holisticai library

- Suppose we wish to compute the multiclass statistical parity with both the max and mean aggregation methods.

```
from holisticai.bias.metrics import multiclass_statistical_parity
multiclass_statistical_parity(p_attr, y_pred, aggregation_fun='max')
```

✓ 0.2s

```
0.5
```

```
multiclass_statistical_parity(p_attr, y_pred, aggregation_fun='mean')
```

✓ 0.2s

```
0.3333333333333333
```

Contents

References

- [1] Putzel et al, Blackbox Postprocessing for Multiclass Fairness (https://arxiv.org/abs/2201.04461)