Bias in Regression Cheat Sheet

Part 1 - Introduction

In binary classification, the output of our AI/ML model was allowed to take two values, often denoted as 0 and 1. In the regression setting, we allow a continuous range of outputs. The algorithm therefore learns a function $f(x)$ that can approximate the trend of the data.

Examples:

- Student grade prediction
- Crime recidivism prediction
- Credit score prediction

Since regression is a very common task in AI, looking at the fairness of regression data is essential.

Part 2 - Notions of Fairness

Reference: Berk et al, 2017, https://arxiv.org/pdf/1706.02409.pdf (https://arxiv.org/pdf/1706.02409.pdf)

Taxonomy

- Individual Fairness
- Group Fairness

Individual Fairness: in this fairness paradigm, we require that similar individuals be treated similarly.

Group Fairness: in this fairness paradigm, we require that different groups be treated similarly on average.

Individual Fairness Cost, with A and B two groups, d a distance function on labels:

$$l(\mathbf{w}, A, B) = \frac{1}{n_1 n_2} \sum_{i \in A, j \in B} d(y_i, y_j)(f(x_i) - f(x_j))^2$$

Group Fairness Cost, with A and B two groups, d a distance function on labels:

$$l(\mathbf{w}, A, B) = \left[\frac{1}{n_1 n_2} \sum_{i \in A, j \in B} d(y_i, y_j)(f(x_i) - f(x_j)) \right]^2$$

The main difference between the two is that the Group fairness notion allows for compensation. **Compensation** refers to the phenomenon of high and low scores within a group "cancelling out".

As with other tasks there is also the distincton between the Equality of Opportunity and Equality of Outcome notions.

Equal Outcome: we want the algorithm to have a similar output distribution for all groups.

Equal Opportunity: we want the algorithm have a similar performance for all groups.

Reference: Agarwal et al, 2019, https://arxiv.org/abs/1905.12843 (https://arxiv.org/abs/1905.12843)

The above paper defines the notion of **bounded group loss** for Equal Opportunity, let A be the set of all protected groups, bounded group loss can be stated as:

$$\forall a \in A, \mathbb{E}[l(y, f(x)|A = a)] \leq \eta$$

The above paper defines the notion of **statistical parity** for Equal Outcome, let A be the set of all protected groups, statistical parity can be stated as:

$$\forall z, a, P(f(x) = z|A = a) = P(f(x) = z)$$

Part 3 - Measuring Bias

To measure and systematize the analysis of bias in regression tasks, we must define and use metrics.

Part 3.1 - Equality of Opportunity

Suppose we have computed some form of loss L of our classifier for groups A and B, we can look at the ratio as a measure equality of opportunity fairness

$$m(A, B) = \frac{L_A(y, \hat{y})}{L_B(y, \hat{y})}$$

We can use the following approach using MSE and RMSE losses. This defines two metrics

$$MAEratio(A, B) = \frac{MAE_A(y, \hat{y})}{MAE_B(y, \hat{y})}$$
$$RMSEratio(A, B) = \frac{RMSE_A(y, \hat{y})}{RMSE_B(y, \hat{y})}$$

The ideal value for such a metric is 1, indicating both groups have the same loss. High values indicate bias towards group A and low values towards group B.

Part 3.2 - Equality of Outcome

In the equality of outcome paradigm, we wish for different groups to have similar outcome distributions.

Comparing distributions with continuous domain involves more challenges than binary classification data. One common approach is to first reduce the regression data to binary data, by using a binarization threshold. With this approach we can define the binary disparate impact of regression data at a given quantile. We will use $Q(q)$ to denote the quantile function (https://en.wikipedia.org/wiki/Quantile_function). We first define the success rate of group a at quantile q:

$$SR_a(q) = P(f(x) \geq Q(q)|A = a)$$

and the disparate impact is

$$DI_a(q) = \frac{P(f(x) \geq Q(q)|A = a)}{\max_a P(f(x) \geq Q(q)|A = a)}$$

Fixed binarization is simple, yet it was the method proposed for the new **Local Law 144** which mandates bias audits in the New York city coucil. They propose to compute a binary disparate impact with a binarization at the median $q = 0.5$. This metric was studied and new metrics are proposed in: Filippi et al, 2023, https://arxiv.org/abs/2302.04119 (https://arxiv.org/abs/2302.04119).

For a two sample statistical testing approach for the equality of distributions, on can look at the Kolmogorov-Smirnov test (https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test).

Part 3.3 - Holisticai library

The **holisticai library** can help in measuring bias. To install the library in a notebook, run the following line of code.

In [ ]:
```
%pip install holisticai
```

Once the library is installed, the documentation can be found here (https://holisticai.readthedocs.io/en/latest/). To import all metrics, one can run the following line.

In [ ]:
```
from holisticai.bias.metrics import *
```

Part 4 - Mitigating Bias

Taxonomy

- Pre-processing
- In-processing
- Post-processing

Pre-processing refers to techniques where mitigation is applied to the data.

In-processing refers to techniques where the mitigation is included in the model and it's training.

Post-processing refers to techniques where the mitigation happens after training, directly on the outcomes.

Part 4.1 - Pre-processing

One possible Pre-processing method, that would work for any AI task, is the **Correlation Remover**. This method works by removing (or reducing) correlations between the sensitive attributes and all attributes used in training. Note: removing correlations does not mean removing all dependence! For instance sensitive attributes could still be recovered using pairs of training attributes.

In equations, this can be formulated as follows. Suppose $\mathbf{s}$ is a sensitive attribute, and $\mathbf{x_1}, \ldots, \mathbf{x_n}$ are the training feature vectors. We find the new vectors $\mathbf{z_1}, \ldots, \mathbf{z_n}$ by solving the following optimization problem

$$\min_{\mathbf{z_1},\ldots,\mathbf{z_n}} ||\mathbf{z_i} - \mathbf{x_i}||^2$$

subject to

$$\sum_{i=1}^{n} \mathbf{z_i}(\mathbf{s_i} - \bar{s})^T = 0$$

Part 4.2 - In-processing

References:

Agarwal et al, 2018, https://arxiv.org/pdf/1803.02453.pdf (https://arxiv.org/pdf/1803.02453.pdf)

Agarwal, 2019, https://arxiv.org/abs/1905.12843 (https://arxiv.org/abs/1905.12843)

Part 4.3 - Post-processing

References:

Chzhen et al, 2020, https://proceedings.neurips.cc/paper/2020/file/51cdbd2611e844ece5d80878eb770436-Paper.pdf (https://proceedings.neurips.cc/paper/2020/file/51cdbd2611e844ece5d80878eb770436-Paper.pdf)

Part 4.4 - Holisticai library

The **holisticai library** can help in mitigating bias. To install the library in a notebook, run the following line of code.

In [ ]:
```
%pip install holisticai
```

Once the library is installed, the documentation can be found here (https://holisticai.readthedocs.io/en/latest/). To import all bias mitigation strategies, one can run the following line.

In [ ]:
```
from holisticai.bias.mitigation import *
```