Holistic AI × The Alan Turing Institute

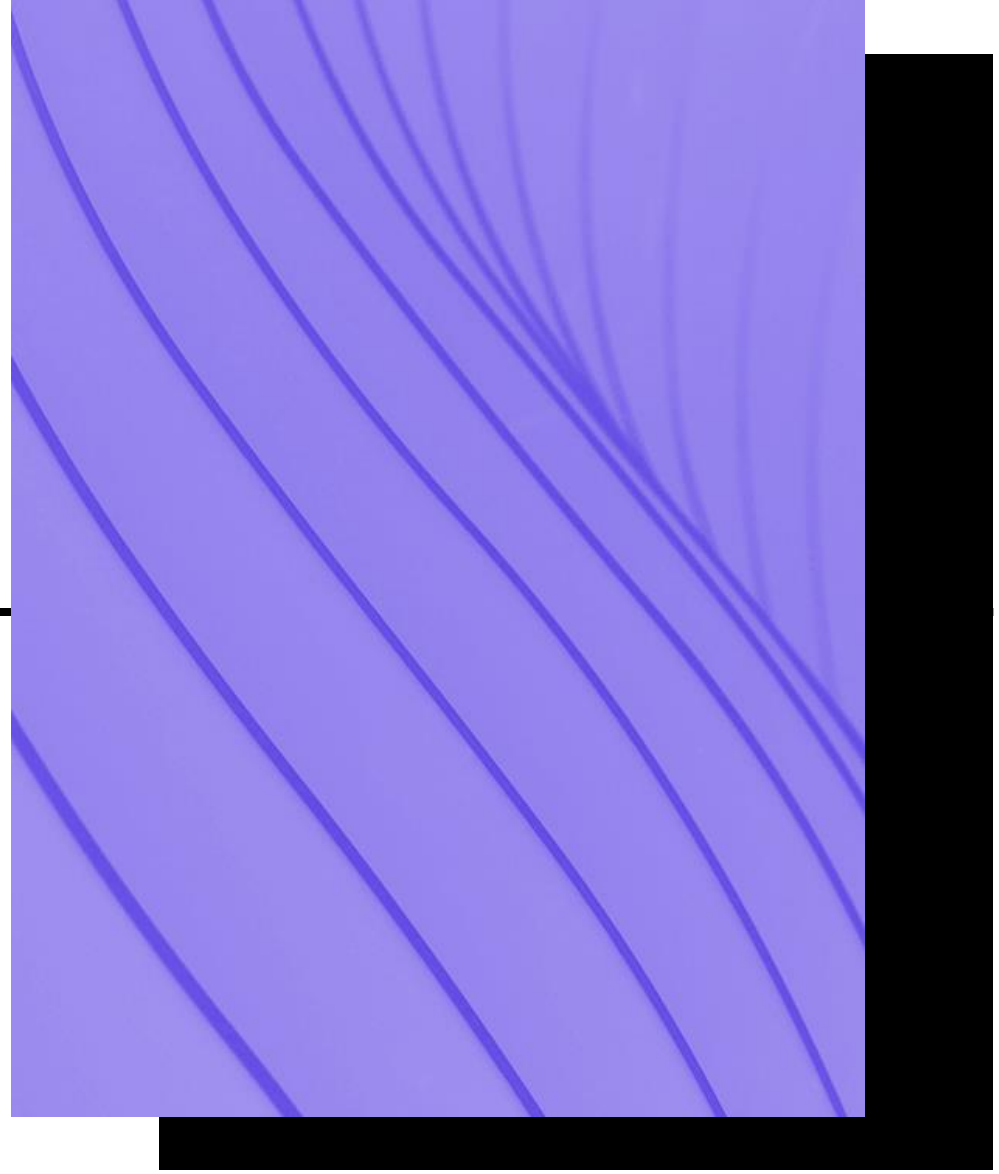
Bias in Multiclass Classification Part IV

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

Speaker: Sara Zannone

Contents

IV – Mitigating Bias in Multiclass Classification



- 1) Introduce one Pre-processing technique
- 2) Introduce one In-processing technique
- 3) Introduce one Post-processing technique

Taxonomy

- As with other tasks, mitigation of bias can be split into three broad categories: Pre-processing, In-processing, Post-processing.
- Pre-processing: the changes are made to the dataset, before the model is trained
- In-processing: the bias mitigation is included in the way we devise and train our model
- Post-processing: the data and model are left unchanged, but we alter the outputs of the model

Pre-processing: Correlation Remover

- The Correlation Remover is a pre-processing technique that works for any AI task. It is also advantageous because it also works with continuous protected attributes.
- As the name implies, this method works by removing the correlations between the protected attributes and any feature vector used in training (while preserving maximal amount of information).
- Note: removing correlations does not mean making independent! There could still be information about protected attribute in the features, for instance in combinations of them.

Pre-processing: Correlation Remover

- Suppose that p is a protected attribute vector and $x_1, \ldots, x_k$ are feature vectors. We can remove the correlations between p and the features as follows.
- The new feature vectors $z_1, \ldots, z_k$ are given by solving
- $\min_{z_1,\ldots,z_k} \left|\left|z_i - x_i\right|\right|^2$ subject to $\sum_i z_i (p_i - \bar{p})^T = 0$.
- Where $\bar{p}$ denotes the mean value of vector p.

In-processing: Fair Score Classifier

- This method was proposed by Rouzot et al, 2022.
- The method is based upon Mixed Integer Linear Programming methods.
- A One VS All model is trained for each class, and the result is then obtained choosing the class associated with the model with highest probability.
- The method can optimize for accuracy, or balanced accuracy subject to fairness constraints.
- The method allows to add one, or a number of different fairness constraints (e.g., prefect statistical parity).

Fairness Constraints

- Satisfying statistical parity is equivalent to a set of linear constraints, which is generally not problematic in the field of optimization.
- Assuming there are M protected groups and N output classes. The statistical parity constraints can be written as
- $P(y_i = k|p_i = j) = P(y_i = k|p_i = l)$
- For all groups j, l and class k.

Post-processing: ML Debiaser

- This method was proposed by Alabdulmohsin et al, 2021.
- The method works in post, which is advantageous for debiasing models trained at scale (DNN) or black-box models.
- Supposing X is the instance, $f(X)$ is the prediction, $p(X)$ is the protected attribute. We must derive a new prediction $h(X)$. We can see this as a chain:
- $X \rightarrow \big(f(X), p(X)\big) \rightarrow h(X)$
- The new predictions are derived with fairness guarantees, such as perfect statistical parity.

Post-processing: ML Debiaser

- The paper proposes to derive the new predictions by minimizing the following objective, subject to the fairness constraints (e.g., statistical parity):
- $L = \frac{\gamma}{2} E[h(X)^2] - E[h(X) \cdot f(X)]$
- The second term ensures the dot product between old and new predictions is maximised (so we are keeping maximal information).
- The first term is a regularization, which helps in achieving the near optimality (see paper for details). In essence gamma controls the randomization of predictions near the threshold.

Mitigating Bias with the holisticai library

- The first step is installing the library with a simple pip install by running the following line of code in a terminal.
- pip install holisticai
- The documentation for the mitigation strategies can be found here.

Example

- In the following example, we show how to train a **Logistic Regression** model with a **Correlation Remover** within the **Pipeline**.

```python
# sklearn imports
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

# import bias mitigation technique and pipeline classes
from holisticai.bias.mitigation import CorrelationRemover
from holisticai.pipeline import Pipeline

# initialize the pipeline
pipeline = Pipeline(
    steps=[
        ('scalar', StandardScaler()),
        ("bm_preprocessing", CorrelationRemover()),
        ("model", LogisticRegression()),
    ]
)
```

```python
# prepare training data and parameters
X, y, group_a, group_b = train_data
fit_params = {
    "bm__group_a": group_a,
    "bm__group_b": group_b
}

# apply steps in pipeline
pipeline.fit(X, y, **fit_params)

# prepare testing data and parameters
X, y, group_a, group_b = test_data
predict_params = {
    "bm__group_a": group_a,
    "bm__group_b": group_b,
}

# make a prediction
y_pred = pipeline.predict(X, **predict_params)
```

References

- [1] Denis et al, Fairness guarantee in multi-class classification.
- [2] Rouzot et al, Learning Optimal Fair Scoring Systems for Multi-Class Classification, 2022
- [3] Alabdulmohsin et al, A Near-Optimal Algorithm for Debiasing Trained Machine Learning Models, 2021