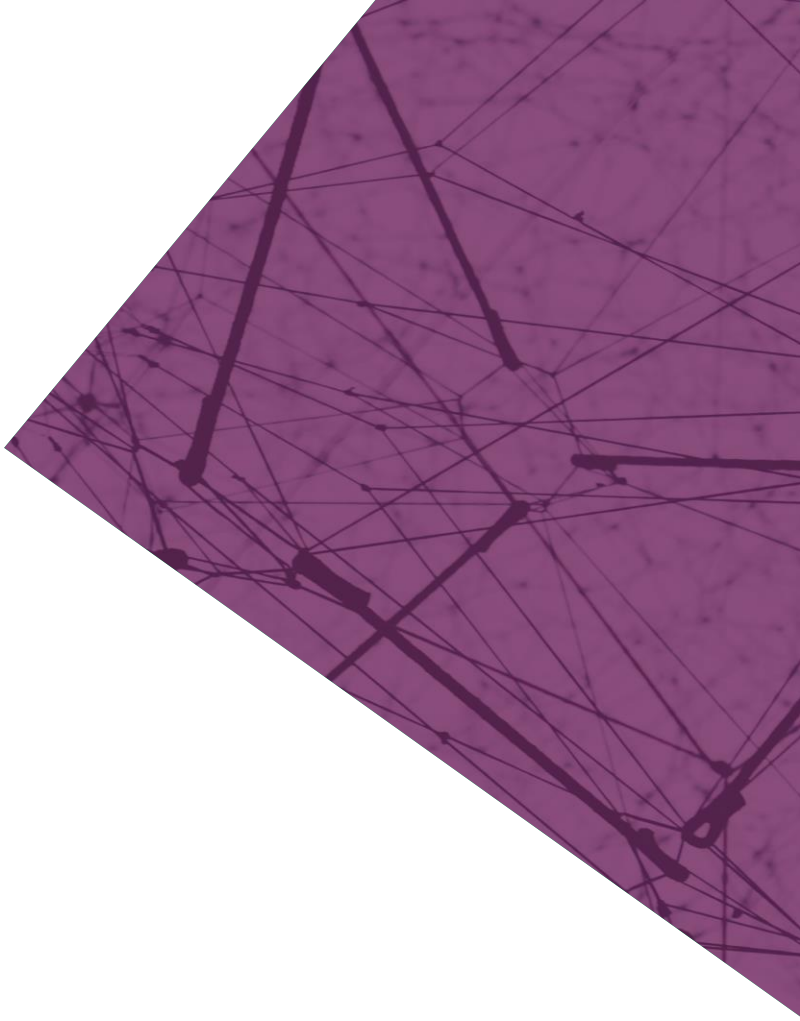
The Alan Turing Institute



Milestone 4:

Mitigating Algorithm Bias and Discrimination in Supervised Learning

Catherine Inness

Learning Objectives

- Milestone 3 introduced measures of model fairness. This Milestone introduces mitigation strategies to improve fairness of supervised learning models.
- By the end of this milestone, you will be able to select, implement and evaluate example methods from the three families of intervention:
  - Pre-processing
  - In-processing
  - Post-processing

Contents

1. Introduction to mitigating unwanted bias in supervised learning
    - What are we mitigating for?
    - What intervention techniques exist to mitigate
    - Selecting a target fairness measure to mitigate
2. Example exercises:
    - Pre-processing example: Reweighing (Kamiran and Calders 2012)
    - In-processing example: Reductions (Argarwal et al 2018)
    - Post-processing example: Equalised Odds Post-processing (Hardt et al 2016)
3. Closing thoughts

Before We Start

Watch this video from two prominent researchers in algorithm fairness:

“A snapshot of the frontiers of fairness in machine learning”, Alexandra Chouldechova and Aaron Roth, May 2020

https://dl.acm.org/doi/10.1145/3376898

Mitigating Algorithm Bias: Choices to Make

To commence our work on mitigating algorithm bias, we'll need to decide on 2 things:

1. Fairness definition and metric
2. Intervention method

Selecting a target fairness measure

- Prior to implementing a mitigation technique, we must agree a definition of fairness.
- There is no universal measure of algorithm fairness.
- Typically a decision made by accountable individuals, not just a data science problem.
- A common approach – use domain knowledge to identify protected groups, e.g.:
  - Groups subject to historic prejudice
  - Groups for which the impact of an erroneous outcome would be highest
- Be sure to validate that your mitigating action has not adversely impacted other groups
- “What gets measured, gets managed” – calculate and present a wide range of fairness measures to understand the impact of a model on individuals and groups

Selecting an Intervention Method

For the purposes of mitigating unwanted bias, causes can be grouped:

1. Inadequate training data:

- A model approximates a function f *(X) that maps the input data X to the target label or value Y. Training data consists of "noisy, approximate examples of *f *(X)* evaluated at different training points" (Goodfellow et al., 2016) and as such may not always accurately represent the function we are aiming to approximate.
- Issues with the data collection process or the distribution of historic outcomes can mean training data contains unwanted bias.

2. Unwanted bias caused by model generalisations:

- Many common machine learning models work by minimising the sum of prediction errors; it is therefore less costly to mis-classify the minority than it is the majority.

3. Model design choices prioritised without considering fairness:

- Common misconception: algorithms are neutral; ‘the data doesn’t lie‘. In reality, model design involves choices of inputs, feature engineering, architecture and hyper-parameters that impact outcomes (Mittelstadt et al., 2016).

What techniques exist to mitigate?

1. Pre-processing:
   - The training data is modified to reduce or remove unwanted bias before training the model, for example by re-sampling.
   - This type of technique is typically model agnostic - you can then use the pre-processed data in any kind of machine learning model.

Example further reading:

- Calmon et al. “Optimized Pre-Processing for Discrimination Prevention”, Advances in Neural Information Processing Systems 30, 2017
- F. Kamiran and T. Calders, “Data Preprocessing Techniques for Classification without Discrimination”, Knowledge and Information Systems, 2012

What techniques exist to mitigate?

2. In-processing:
   - The model architecture and/or optimisation process is adjusted to target fairness, as well as prediction error minimisation.
   - This type of technique will not usually be model-agnostic.

Example further reading:

- Agarwal et al, “A Reductions Approach to Fair Classification”, 35th International Conference on Machine Learning, 2018
- Zemel et al. “Learning Fair Representations”, Proceedings of Machine Learning Research, Vol. 28, 2013
- Zhang et al., “Mitigating Unwanted Biases with Adversarial Learning”, AAAI/ACM Conference on AI, Ethics, and Society, 2018

What techniques exist to mitigate?

3. Post-processing:
   - Fairness is targeted in separate processing using outputs from a standard predictive model.
   - Usually model-agnostic, but it does require that the protected attribute is available when inference is performed.

Example further reading:
- Hardt et al., Equality of Opportunity in Supervised Learning, 30th International Conference on Neural Information Processing Systems, 2016
- Kim et al., Multiaccuracy: Black-Box Post-Processing for Fairness in Classification, 2019

Exercises

- There are 3 Jupyter notebooks that will guide you through 3 different example implementations of algorithm fairness intervention methods.
- The next few slides provide further guidance.

Our target fairness measure

- For our example exercise:
  - We focus on group fairness, defined as the absence of discrimination against those with a protected attribute or characteristic.
  - Our protected variable is ethnicity (White/ Non-white)
  - We will cover two group fairness definitions. See Milestone 3 for detailed definitions.

1. Statistical parity:

"Our hiring algorithm must assign a 'hire' outcome to the same proportion of white applicants compared with other ethnicities."

2. Equalised odds:

"Our hiring algorithm must assign a 'hire' outcome to the same proportion of appropriately qualified white applicants compared with equally qualified applicants of other ethnicities"

Our example fairness interventions

Example notebooks contain exercises to implement:

1. Pre-processing example:
   - Reweighing (Kamiran and Calders 2012) to target Statistical Parity

2. In-processing example:
   - Reductions approach to fairness (Argarwal et al 2018), to target Statistical Parity

3. Post-processing example:
   - Equalised Odds Postprocessing (Hardt et al 2016)

A note on libraries

This course makes use of two commonly used Python libraries for fairness: IBM’s AIF360 and Microsoft’s Fairlearn. Both libraries are open-source and implement methods based on peer-reviewed scientific papers.

Our Example Hiring Algorithm

- Binary classification: hire (1) or do not hire (0)

- Training data:
  - $X_i \in \mathbb{R}^d$ (large number of feature variables describing each individual candidate)
  - $Y_i \in \{0,1\}$ (hired or not hired)

- For pre-processing and in-processing examples we use a standard ridge classifier as our base model as the training dataset has a high number of independent variables, likely to be correlated.

- For our post-processing example we use logistic regression as our base model for simplicity, as this method requires predicted probabilities as an input.

Pre-processing Example: Reweighing

- Reweighing applies a weight to each tuple in the training data. Each weight is calculated such that the training data, with weights applied, achieves statistical parity between a protected group compared with the privileged group.

- *'Assigns lower weights to objects that have been deprived or favoured'* in the training set
- *'The weight of each tuple is the expected probability to see an instance with its sensitive attribute value and class given independence, divided by its observed probability.'*

D = labelled dataset

S = binary sensitive attribute with domain *{b,w}*

Class = binary target attribute with domain *{-,+}*

X = random unlabelled data object

W = Weight assigned to each object *X*

Algorithm : *Reweighing*

Input: $(D, S, Class)$
Output: Classifier learned on reweighed D
1: **for** $s \in \{b, w\}$ **do**
2: **for** $c \in \{-, +\}$ **do**
3: Let $W(s, c) := \frac{|\{X \in D \mid X(S) = s\}| \times |\{X \in D \mid X(Class) = c\}|}{|D| \times |\{X \in D \mid X(Class) = c \text{ and } X(S) = s\}|}$
4: **end for**
5: **end for**
6: $D_W := \{\}$
7: **for** X in D **do**
8: Add $(X, W(X(S), X(Class)))$ to D_W
9: **end for**
10: Train a classifier C on training set D_W, taking onto account the weights
11: **return** Classifier C

Paper: F. Kamiran and T. Calders, "Data Preprocessing Techniques for Classification without Discrimination," Knowledge and Information Systems, 2012

Reweighing: Implementation Notes

- Implemented using the AIF360 library.

- Within the AIF360 library:
  - *BinaryLabelDataset* is the base class for binary labelled datasets
  - *BinaryLabelDatasetMetric* is the class for computing metrics based on a *BinaryLabelDataset*

- The privileged group is typically the group that we have reason to believe has been treated favourably in the past compared with the rest of the population : in our case we use ‘Ethnicity_White’. Our unprivileged group is all other ethnicities.

- Be sure to only reweigh the training data, do not apply reweighing at inference time.

Paper: F. Kamiran and T. Calders, “Data Preprocessing Techniques for Classification without Discrimination,” Knowledge and Information Systems, 2012

Reweighing: Exercise

Use the Jupyter notebook provided to:

1. Install libraries and load the data
2. Set up the fairness metrics to be computed
3. Calculate Disparate Impact and Statistical Parity Difference metrics for our training dataset
4. Use the Reweighing technique on the training dataset (without a predictive model) to show that Disparate Impact approaches 1 and Statistical Parity Difference approaches 0.
5. Train a baseline ridge regression classifier on the dataset before Reweighing and compute fairness and model performance metrics.
6. Train the same classifier on the dataset after Reweighing and compare the fairness and performance metrics.

In-processing Example: Reductions

- The Reductions approach (Agarwal et al 2018) makes use of constrained optimisation to reduce binary classification to a series of cost-sensitive, weighted classification problems. The optimal solution is then an equilibrium between two min max expressions.

- Inputs: training data, standard machine learning algorithm (black box)
- Approach: iteratively call the black box algorithm and reweight/ relabel the data
- Output: a classifier that minimises prediction error subject to the selected fairness constraints

Paper: Agarwal et al, “A Reductions Approach to Fair Classification”, 35th International Conference on Machine Learning, 2018

Reductions: Implementation Notes

- Implemented using the Fairlearn library.

- The sensitive feature is typically the feature belonging to the group that we have reason to believe has been treated favourably in the past compared with the rest of the population: in our case we use ‘Ethnicity_White’.

- Fairlearn mitigation algorithms use the *fit* method to train a model and *predict* to make predictions.

Reductions: Exercise

Use the Jupyter notebook provided to:

1. Install libraries and load the data
2. Set up the fairness metrics to be computed
3. Train a baseline ridge regression classifier on the dataset before Reweighing and compute fairness and model performance metrics.
4. Train the same classifier using the Reductions method, targeting Statistical Parity. Compare the metrics with the baseline model.

Post-Processing Example: Equalised Odds

- The Equalised Odds Post-Processing approach (Hardt et al 2016) flips predictions outputted from our base model until the desired error rate distribution between the protected group and the privileged group is achieved.

- Inputs: the predicted probability from our base model, the target label, and the protected group

- Equalized Odds Postprocessing first uses convex optimisation to find the optimum rates at which to flip predictions, using the True Positive Rates and False Positive Rates for each group. The method then flips predictions using these rates to achieve equalised odds.

Paper: Hardt et al., “Equality of Opportunity in Supervised Learning”, 30th International Conference on Neural Information Processing Systems, 2016

Post-Processing: Implementation Notes

- The privileged group is typically the group that we have reason to believe has been treated favourably in the past compared with the rest of the population : in our case we use ‘Ethnicity_White’. Our unprivileged group is all other ethnicities.

- Within the AIF360 library:
  - *BinaryLabelDataset* is the base class for binary labelled datasets
  - *BinaryLabelDatasetMetric* is the class for computing metrics based on a *BinaryLabelDataset*

- We use logistic regression, as this post-processing method takes predicted probabilities as an input rather than predicted labels.

Post-Processing: Exercise

Use the Jupyter notebook provided to:

1. Install libraries and load the data
2. Set up the fairness metrics to be computed
3. Train a baseline ridge regression classifier on the dataset before Reweighing and compute fairness and model performance metrics.
4. Train the same classifier using the Reductions method, targeting Statistical Parity. Compare the metrics with the baseline model.

Closing Thoughts

- For our example, Equalised Odds post-processing led to the best fairness metrics and is simplest to implement. But requires access to the protected feature at inference time.

- Fairness is not just a data scientist problem. To mitigate algorithm bias, a definition and understanding of impacts on individuals and groups must have been agreed with accountable experts, e.g. in ethics and legal. Considerations:
  - Seeking demographic parity in predictions where the true, ‘real world’ values do not represent demographic parity means corrective action or positive discrimination.
  - Equality Of Odds requires that the model performs equally well for different subgroups – harder to argue with. But can be met simply by reducing the accuracy of the best performing group down to the same level as the group with the lowest performance. Consider whether this is acceptable.

Closing Thoughts

- Consider fairness throughout the full Machine Learning lifecycle, rather than rely on mitigating algorithms.

- Always calculate and present fairness metrics.

- Consider non-technical solutions too e.g. algorithm impact assessments and model cards (Mitchell et al 2019).