BiMi Sheets: Infosheets for bias mitigation methods

MaryBeth Defrance, Guillaume Bied, Maarten Buyl, Jefrey Lijffijt, Tijl De Bie
Ghent University
Technologiepark-Zwijnaarde 116,
9052 Gent, Belgium
{firstname}.{lastname}@ugent.be

Abstract

Over the past 15 years, hundreds of bias mitigation methods have been proposed in the pursuit of fairness in machine learning (ML). However, algorithmic biases are domain-, task-, and model-specific, leading to a 'portability trap': bias mitigation solutions in one context may not be appropriate in another. Thus, a myriad of design choices have to be made when creating a bias mitigation method, such as the formalization of fairness it pursues, and where and how it intervenes in the ML pipeline. This creates challenges in benchmarking and comparing the relative merits of different bias mitigation methods, and limits their uptake by practitioners.

We propose BiMi Sheets as a portable, uniform guide to document the design choices of any bias mitigation method. This enables researchers and practitioners to quickly learn its main characteristics and to compare with their desiderata. Furthermore, the sheets' structure allow for the creation of a structured database of bias mitigation methods. In order to foster the sheets' adoption, we provide a platform for finding and creating BiMi Sheets at bimisheet.com.

Introduction

The risk of algorithms reproducing or amplifying social biases has drawn considerable attention to fairness in AI. While technical solutions may not always be adequate (Wachter, Mittelstadt, and Russell 2021), hundreds of algorithmic bias mitigation methods have nevertheless been proposed in the pursuit of fair machine learning. The diversity of these methods stems from a large number of necessary design choices, and from the complex nature of fairness itself. For instance, methods may differ in (i) how fairness is formally defined, (ii) the dataset types they are compatible with, (iii) the machine learning tasks they target, (iv) the stage of the machine learning pipeline where they intervene, (v) the machine learning models they are compatible with, and more. Despite being all ostensibly designed with the same goal–mitigating algorithmic bias–they thus generalize poorly across socio-technical contexts. This difficulty was dubbed the 'portability trap' by Selbst et al. (2019).

Yet, this diversity of bias mitigation methods raises challenges–both for their adoption in practice and for the progress of academic research. On the one hand, practitioners report challenges when searching for bias mitigation methods appropriate for their use case. Based on a user study, Deng et al. (2022) find that the adoption of fairness tools would improve if *more guidance and support in contextualizing toolkit functionalities or outputs, beyond the level of documentation provided by standard software packages* was provided. On the other hand, good benchmarking practices have been key to progress in various sub-fields of machine learning. Yet, in fair-ML, novel methods are often benchmarked against a small, established set of bias mitigation methods (most of which proposed between 2016 and 2019), ignoring more recent advances in the literature, and even though some of the baseline methods may not be viable alternatives in practice (Delaney et al. 2024; Han et al. 2024; Cruz and Hardt 2024).

In the end, such challenges may result in an increasing misalignment between the technical tools for mitigating algorithmic bias and the socio-technical concerns that motivated the tackling of this bias in the first place (Weerts et al. 2024). We contend that steps must be taken to elaborate on the technical details and assumptions for bias mitigation methods in a consistent manner, such that the technical scope of each method can be clarified. In turn, a consistent common language for bias mitigation tools may then facilitate the socio-technical discussion on the appropriateness of a given method for a particular use case.

Contributions We therefore introduce *BiMi Sheets*: information sheets for bias mitigation methods with two goals.

1. To serve as *documentation*, providing detailed, consistent information to practitioners on the bias mitigation method's design choices and constraints
2. To establish *structure* for the myriad of existing bias mitigation methods such that research and benchmarks can find common ground and continue to progress.

The sheets' design is informed by a variety of taxonomies and terminologies from surveys on AI fairness, as well as key needs from practioners and researchers identified in previous research. A schematic example is shown in Fig. 1. The BiMi Sheets consist of six sections that respectively detail the bias mitigation method, its location in an ML pipeline, its formalization of fairness, other implementation constraints, reports on tested use cases, and any additional metadata. To

Example: BiMi Sheet

Metadata

Name: Example
Authors: Name 1, Name 2, and Name 3
Version: Version number License: License info
Proposed in Reference to the source paper (if it exists)

Method Description

Method Type$^{\dagger *}$ | ML Task$^{\dagger *}$ | Compatible Dataset Type$^{\dagger *}$

Description detailing what intervention the method does in the pipeline.

Pipeline Architecture

Pipeline Location | Compatible Model$^{\dagger *}$

A more detailed description on the requirements for the model.

Fairness Type

Composition Sens. Attr.* | Fairness Guarantee
Fairness Type$^{\dagger *}$ | Fairness Definition$^{\dagger *}$

Detailed description of the fairness concept that the method enforces on a system, including the calculation.

Implementation Constraints

Programming Language$^{\dagger *}$ | Package$^{\dagger *}$

Description of specific implementation information, such as dataset format and expected ML model format.

Tested Use Cases

Dataset$^{\dagger *}$

Information about performance details compared to other models and additional testing of the method, which was not done using this source code.

Figure 1: Bare-bones example of a BiMi Sheet. The symbol † denotes that the label in the tag is free to choose. The symbol * denotes that multiple tags for this property can be provided in a BiMiSheet.

kickstart the adoption of BiMi Sheets, we already provide 24 BiMi Sheets for popular, recent bias mitigation methods on a searchable web application at bimisheet.com.

Outline The rest of this paper is organized as follows. After reviewing related work, we highlight the existence of documentation debt for bias mitigation methods, motivating the proposal of BiMi Sheets. We proceed to describe the proposed structure of a BiMi Sheet, motivating the adopted design choices based on the needs of users and common practices within AI fairness research. We end with a discussion on the ongoing challenges for the documentation and structuring of bias mitigation methods in the future.

Related work

Our work is informed by past efforts on improving the *documentation* and *structuring* of bias mitigation methods.

Documentation for fairness in AI

Documentation of bias mitigation methods remains understudied. However, some documentation initiatives exist for communicating fairness characteristics of resources, other than bias mitigation methods. We limit our discussion to works focussing specifically on the fairness aspects and therefore not discuss initiatives like System Cards (Gursoy and Kakadiaris 2022) or Audit Cards (Staufer et al. 2025).

Datasheets for datasets (Gebru et al. 2021) Datasheets for datasets call for the documentation of the creation, composition, intended uses, maintenance, and other properties of datasets. The goal of datasheets is two-fold.First, to ensure the proper use of datasets by making the critical knowledge of domain experts available through the datasheet, which quickly communicates the capabilities, strengths, and limitations of the dataset to the machine learning expert. Second, to urge the creators to critically reflect on the limitations and (un)intended impact of their dataset by pushing them to answer a diverse set of questions.

Model cards for models (Mitchell et al. 2019) Model cards are short documents accompanying trained machine learning models that provide benchmarked evaluation in a variety of conditions. Model cards are widely in use, as they are integrated for models provided on HuggingFace (Face 2025). The main elements discuss its intended use, the sensitive attributes the model can take into account, its performance metrics, training data, evaluation data, and limitations. These let users determine the trained model's applicability for their use case and enable a performance comparison with other trained models.

FactSheets by IBM (Arnold et al. 2019) Factsheets focus on documented AI services. This notion differs slightly from the model cards. An AI service could consist out of a single AI model, but it can also be constructed of many elements. FactSheets also aim to encompass more than just the fairness aspect of the service, including also explainability and security. The concept 'AI service' is very broad, so the suggested structure of the FactSheets is inevitably rather large in order to encompass all conceivable characteristics that such a system might exhibit.

Aequitas (Saleiro et al. 2019) The Aequitas Toolkit aims at auditing machine learning models on the possible biases they may exhibit and provide guidance for determining the desired properties of a fair machine learning model. This initiative focusses on creating documentation for a specific application rather than a method.

BiMi Sheets in comparison Datasheets, Model cards, and FactSheets document the biases present in the dataset, model, and AI service respectively. These biases in part arise from real-world biases. BiMi Sheets are distinct in this regard, as they do not document biases, but the design choices of bias mitigation methods. Indeed, a bias mitigation method's strengths and weaknesses stem from the design choices made when creating the algorithm. These design choices translate into enforcing a specific fairness notion (Green 2019).

Structuring for fairness in AI

BiMi Sheets complement surveys of bias mitigation methods in machine learning, and build on the taxonomies they propose to categorize these methods. While Mehrabi et al. (2021); Caton and Haas (2024); Pagano et al. (2023) review fair machine learning with an emphasis on binary classification, other works focus on specific domains such as machine learning on graphs (Laclau, Largeron, and Choudhary 2024), large language models (Gallegos et al. 2024; Chu, Wang, and Zhang 2024), information access and recommendation (Ekstrand et al. 2022; Li et al. 2023), or gender bias in natural language processing and computer vision (Bartl et al. 2024). Beyond their apparent diversity, these surveys leverage common elements to describe bias mitigation methods–a method's associated fairness types and definitions, compatible dataset types, associated machine learning tasks, pipeline, and model compatibility. BiMi Sheets build on these common elements and on domain-specific taxonomies to propose a flexible documentation structure for bias mitigation methods across domains.

Documentation debt in bias mitigation

Our work is a response to the *documentation debt* in the current state of bias mitigation methods that limits their comparability in research and their usability in practice. In what follows, we revisit the motivation for documentation for researchers and practioners, before mapping the current state of documentation across recent and popular toolkits.

Documentation for researchers Researchers that propose bias mitigation methods need a way to compare their method against existing methods such that its strengths and weaknesses can be analyzed on a quantitative and qualitative basis. Yet, unlike other fields in ML where progress can steadily be mentioned on benchmarks and leaderboards, no general leaderboard can exist for bias mitigation methods (Wang, Hertzmann, and Russakovsky 2024) because formalizations of algorithmic bias are so specific to the socio-technical context. Hence, the design choices of a method, and the resulting constraints for the socio-technical context, must be alignable across methods for their comparison to be meaningful (Defrance, Buyl, and Bie 2024).

	Method Approach	*Compatible Models*	*Pipeline Location*	*Compatible Datasets*	*Composition Sensitive Attributes* [1]	*Fairness Guarantee*	*Fairness Notion*	*Implementation Constraints*
Adversarial Debiasing - AIF360[2]	A	A*	A	A*	A	A*	A	N
Calibrated Equalized Odds - AIF360	?	A*	A	-	?	?	A	A
Deterministic Reranking - AIF360	N	N*	A	-	?	N*	N*	A
Disparate Impact Remover - AIF360	?	-	A	A*	?	A*	?	A
Exponent. Gradient Reduct. - AIF360	A	-	A	-	?	A	A	A
GerryFair Classifier - AIF360	?	A	A	-	?	N*	A	A, N
Grid Search Reduction - AIF360	A	-	A	-	?	?	A	A
LFR - AIF360	A	-	A	?	?	A*	A	A
MetaFair Classifier - AIF360	?	?	A	?	A	A*	A, N	A
Optimized Preprocessing - AIF360	A	-	A	?	?	?	A	A
Prejudice Remover - AIF360	A	A*	A	-	?	A*	?	A
Reject Option Classification - AIF360	A	A*	A	-	?	?	A	A
Reweighing - AIF360	?	A*	A	-	?	?	?	A
Adversarial Mitigation - Fairlearn	U	U	U*	U*	U	A*	U	U
Correlation Remover - Fairlearn	U	-	U	U*	A	U, A	U	U*
Reductions - Fairlearn	U	U	U*	-	?	A	U	A
Threshold Optimizer - Fairlearn	U	U*	U	-	?	U	U	U
Error-Parity	U	U	U	-	N*	U*	U	U
Fairret	U*	U*	U*	?	N*	N*	A*	U
OxonFair	U	U*	U	-	U	U	U	U

Table 1: Occurrence for where different characteristics of a bias mitigation methods can be found in online documentation. A: API reference, U: User documentation, N: Notebooks, ?: Not found, -: Not relevant for the method. *: The information can be deduced by someone with an advanced knowledge in AI fairness. This table only states whether or not this information is available, not how straightforward this information is to find.

Documentation for practitioners In contrast, practitioners want methods that are easily integrated into their existing ML pipelines without imposing invasive requirements (Lee and Singh 2021). Yet, practioners have complained that documentation of bias mitigation methods is often incomplete or overly technical (Deng et al. 2022), while they struggle to find methods that are compatible with their ML task and the datasets they are working with (Richardson et al. 2021).

The current state of documentation Having established the importance of documentation for both researchers and practitioners, let us now consider if and how the online documentation of bias mitigation methods sufficiently address their needs. Focusing on recent and well-known fairness toolkits, we consulted the documentation of AIF360 (Bellamy et al. 2018), FairLearn (Weerts et al. 2023), error-parity (Cruz and Hardt 2024), fairret (Buyl, Defrance, and Bie 2024), and OxonFair (Delaney et al. 2024). For each, we checked the documentation along several axes, including

(i) a description of the method's overall approach
(ii) the underlying ML models it is compatible with (e.g. logistic regression, neural nets, etc.)
(iii) the stage of the ML pipeline where it intervenes
(iv) the types of compatible datasets
(v) how it delineates protected groups
(vi) whether it offers guarantees on fairness
(vii) how it defines fairness mathematically
(viii) any implementation constraints (e.g. the need for a *scikit-learn* pipeline)

Table 1 documents if, and where, selected characteristics of bias mitigation methods can be found in their online documentation. We find no uniformity to exist with respect to where or even whether these elements are communicated in online documentation across, or even within, libraries. Additionally, many characteristics are left implicit to some extent, requiring deduction based on advanced knowledge of bias mitigation methods. Altogether, the importance of documenting various aspects of bias mitigation methods for different types of users, and the existence of a *documentation debt*, motivate the introduction of BiMi Sheets as a portable and uniform guide to document bias mitigation methods.

[1] We assume that each bias mitigation method can use binary sensitive attributes. We note a question mark if it is not clearly stated which composition it can handle and only examples using binary sensitive attributes are provided.

[2] In AIF360 the API reference and User documentation are identical, we therefore only report the API reference.

Structure of the sheets

BiMi Sheets document the specific implementation of a bias mitigation method. Any algorithmic intervention aimed at reducing bias, either somewhere in the model pipeline or in the outcome, is considered a bias mitigation method. Accordingly, BiMi Sheets need to account for a wide range of methods.

BiMi Sheets are structured in six sections, each of which discusses a specific property of the bias mitigation method. All sections, except the one for metadata, contain both labels and free text. The labels provide a structuring of the bias mitigation methods, making their comparison easier. The free text serves to communicate the intricate details which define the specific method.

For the sake of illustration, a bare-bones version of a BiMi Sheet can be found in Figure 1. The names for all labels can be found in this bare-bones BiMi Sheet. Remark the shading of these labels: lighter-shaded labels are conceptually children components of the normally shaded version.

In the following subsections we discuss each section of the BiMi Sheet. We first provide a narrative description which situates the content of the section with the literature. Each subsection is concluded with a list discussing the label information, such as possible values.

Metadata

The Metadata section communicates basic non-fairness related information for the bias mitigation method.

Label Information

- **Name and Authors.** The combination of the bias mitigation method and authors' names identify a specific method. The authors are the creators of the implementation, not the concept. The authors can refer to a set of individuals or to the fairness toolkit directly.
- **Version.** The version conveys the method's software version number. A BiMi Sheet corresponds to a specific version as other versions might have a different set of capabilities, or approach fairness slightly differently.
- **License.** License information is important for practitioners: An unknown software license complicates adoption.
- **Proposed in.** If the method was first proposed in a research paper, then refer to the paper here.

Method description

Several taxonomies have been proposed in order to provide an initial structuring of bias mitigation methods (Bartl et al. 2024; Caton and Haas 2024; Chu, Wang, and Zhang 2024). This structuring is based on the type of intervention a method undertakes on the pipeline. The type of intervention is a design choice that corresponds to a fairness notion. We rename the labels from these taxonomies to **Method Type**. A well-known example in binary classification of such a method type is *Adversarial Debiasing*. With this method type some element in the pipeline is transformed in order to prevent an adversarial model from guessing a sample's sensitive attributes.

The need for fairness interventions stretches a broad range of applications. As a consequence, bias mitigation methods cannot function for all applications. We characterize an application as the combination of two labels: **ML Task** and **Compatible Dataset Type**. The ML Task stands for the objective the ML model aims to achieve. Practitioners have critiqued the large focus on classification in proposed bias mitigation methods (Richardson et al. 2021). The ML Task and Dataset Type labels will facilitate finding methods for other purposes and perhaps entice researchers to create bias mitigation methods for a broader range of problems.

Both labels are needed to illustrate the application in order to account for pre- and post-processing methods. The nature of pre- and post-processing methods often causes one of the aforementioned labels to become trivial. Pre-processing methods mitigate biases in the dataset themselves, meaning that these methods generally can be used for any ML Task compatible with a specific dataset type. For example, pre-processing methods for graphs can be seen as *task independent*, however its possible application is communicated through its compatible dataset type, *graphs* (Laclau, Largeron, and Choudhary 2024). On the other hand, post-processing methods affect the model outputs and become independent of the dataset type. A post-processing method for binary classification can be used for both tabular, image, and text datasets (Delaney et al. 2024).

A bias mitigation method might focus on a specific subproblem within an ML task. For instance, binary classification can either be evaluated with respect to hard labels or soft scoring. Bias mitigation methods designed for hard labels suffer in performance when evaluated for soft scoring compared to methods designed for that purpose (Defrance, Buyl, and Bie 2024). The reported depth within a ML task depends on the bias mitigation method's properties and focus.

Label Information

- **Method Type.** Method type provides information on the type of intervention the method does in the pipeline. These method types and their characteristics can be found in surveys discussing AI fairness. An overview of method types can be found in Table 2.
- **ML Task.** ML Task communicates the specific machine learning tasks for which the method was designed. It is possible to have multiple ML task labels for one method. Some examples of ML tasks are given in Table 3.
- **Compatible Dataset Type.** The compatible dataset labels communicate the dataset types for which the method is known to work. Examples of dataset types include tabular datasets, image datasets, text datasets, and recommendation datasets.
- **Method Description.** This section provides a detailed explanation of the intervention that the bias mitigation method does in the pipeline. This explanation should be unique compared to other BiMi Sheets, unless a specific method has several implementations.

Source	Pipeline location	Method Types
Caton and Haas (2024) - Binary Classification	Pre-processing	Blinding, Causal Methods, Sampling and Subgroup Analysis, Transformation, Relabelling and Perturbation, Reweighing, Adversarial Learning
	In-processing	Transformation, Reweighing, Regularization and Constraint Optimisation, Adversarial Learning, Bandits
	Post-processing	Calibration, Thresholding
Bartl et al. (2024) - Computer Vision	Pre-processing	Sampling
	In-processing	Adversarial Debiasing
	Intra-processing	Adversarial Debiasing, Learning Representations, Model fine-tuning
	Post-processing	Model fine-tuning
Chu, Wang, and Zhang (2024) -	Pre-processing	Data Augmentation, Prompt Tuning
Large Language Models	In-processing	Loss Function Modification, Auxiliary Module
	Intra-processing	Model Editing, Decoding Method Modification
	Post-processing	Chain of Thought, Rewriting
Gallegos et al. (2024) - LLMs	Pre-processing	Data Balancing, Selective Replacement, Interpolation, Dataset Filtering, Instance Reweighting, Equalized Teacher Model Probabilities, Exemplary examples, Word Lists, Modified Prompting Language, Control Tokens, Continuous Prompt Tuning, Projection-based Mitigation
	In-processing	Architecture Modification, Equalizing Objectives, Fair Embeddings, Attention, Predicted token distribution, Dropout, Contrastive Learning, Adversarial Learning, Reinforcement Learning, Selective Parameter Updating, Filtering Model Parameters
	Intra-processing	Constrained Next-token Search, Modified Token Distribution, Weight Redistribution, Modular Debiasing Networks
	Post-processing	Keyword Replacement, Machine Translation, Other Neural Rewriters

Table 2: Overview of the proposed Method Types from the surveys of Caton and Haas (2024); Bartl et al. (2024); Chu, Wang, and Zhang (2024).

Source	Context	ML Tasks
(Bartl et al. 2024)	NLP	Occupation Classification, Sentiment Analysis, Machine Translation
(Mehrabi et al. 2021)	General	Classification, Regression, Community detection, Clustering, Machine translation, Semantic role labeling, Named Entity Recognition
(Laclau, Largeron, and Choudhary 2024)	Graphs	Node classification, Edge prediction, Community detection, Graph property prediction
(Gallegos et al. 2024)	LLMs	Classification, Question-answering, Logical reasoning, Fact retrieval, Information extraction

Table 3: Overview of discussed ML tasks in the surveys of Bartl et al. (2024); Mehrabi et al. (2021); Laclau, Largeron, and Choudhary (2024); Gallegos et al. (2024).

Pipeline architecture

One of the most used properties to differentiate bias mitigation methods is their **Pipeline Location**, i.e. in which part of the machine learning pipeline the intervention occurs. This influences the capabilities of a method (Defrance, Buyl, and Bie 2024). Additionally, pipeline location is an important compatibility constraint in practical settings (Richardson et al. 2021), since it is not always possible to intervene at any location. Historically, the prevailing division was pre-, in-, and post-processing (Bartl et al. 2024; Caton and Haas 2024; Ekstrand et al. 2022; Laclau, Largeron, and Choudhary 2024; Mehrabi et al. 2021). Novel research on large language models introduced a fourth pipeline location, namely intra-processing (Chu, Wang, and Zhang 2024; Gallegos et al. 2024).

Research on the needs of practitioners with regards to fairness toolkits has shown that an important feature of a toolkit is how easily it integrates in existing workflows (Deng et al. 2022; Lee and Singh 2021). Knowing the **Compatible Models** is an important constraint determining whether or not the method would fit in the existing workflow. Pre-processing methods are often method independent as their intervention occurs before data is processed by the model. Even so, Reweighing—a pre-processing method from AIF360 (Bellamy et al. 2018)—requires a model that can incorporate sample weights. Post-processing methods process the output of a model. Therefore, the compatibility of a model is not dependent on the technique, but on the output format. This results in most post-processing classification methods requiring probabilistic classifiers.

More information on the constraints imposed by the method on the pipeline is described in the free text of this section. This additional information could for example list the models with which a model independent method has been tested. Another possible use is for mentioning particular assumptions, such as for Deterministic Reranking from AIF360 (Bellamy et al. 2018), which assumes that the provided rankings are ordered by descending score.

Label Information

- **Pipeline Location.** The location in the pipeline signifies where the method intervenes. This property affects the capabilities of the bias mitigation method. An overview of the possible pipeline locations and their description can be found in Table 4.
- **Compatible Model.** States the models which are technically compatible and have been shown to have acceptable performance after applying the bias mitigation method.
- **Model description.** In the model description, more detailed information is provided on the theoretically compatible models. For example, the pre-processing method CorrelationRemover from Fairlearn (Weerts et al. 2023) is model independent, however it has been shown that it is most appropriate to be combined with linear models. This is an example of additional information that should be available when assessing the compatibility with a use case.

Fairness type

Bias occurs when people with differing sensitive attributes are treated differently. Most older bias mitigation methods can only account for binary attributes, with gender as prototypical example. However, this is often insufficient to mitigate biases in practice. Today, it is understood that there is a need to account for more complex sensitive attributes, including **Compositions of Sensitive Attributes**. For example, the works of Buolamwini and Gebru (2018) call for intersectional fairness, meaning that each combination of sensitive attributes denotes one sensitive group. This can be translated to one categorical attribute which contains this concatenation of sensitive attributes.

A problem arises when many sensitive attributes are present or when certain combinations are rare. Due to a lack of statistical power for small groups, it might be impossible to calculate a fairness measure for each group. An alternative solution is to account for parallel attributes, meaning that per set of sensitive attributes fairness is ensured (Defrance, Buyl, and Bie 2024). This is a weaker fairness notion compared to intersectional fairness, but still accounts for multiple axes of sensitive attributes.

Fairness in AI is often described as satisfying a specific fairness definition. While such a **Fairness Guarantee** of satisfying a definition may seem desirable, providing a guarantee is often accompanied by direct implications on other performance metrics. Sylvester and Raff (2018) have found that practitioners would sometimes prefer applying a method that improves fairness, but does not fully guarantee fairness in order to achieve their design goals.

Several interpretations of fairness exist in literature. Bias mitigation methods target one or more of these interpretations. We call these interpretations **Fairness Types** to emphasize the link with the specific **Fairness Definitions** that fall under these higher-order fairness types. Binary classification has a well-known split based on fairness type, namely between *Group Fairness* and *Individual Fairness*. Other fairness concepts exist such as egalitarianism, Rawlsian justice, or Nozick's entitlement theory, however no practical bias mitigation methods have been proposed for these concepts. Still, the flexibility of the labels would allow for these concepts to be stated as a fairness type.

The previously mentioned labels cannot convey the full fairness notion that a bias mitigation method enforces: a **Fairness Description** text area is therefore provided to allow for further details. This description aims to help against the 'portability trap' of Selbst et al. (2019). The 'portability trap' discusses how enforcing fairness is highly contextual and can only be reused after an informed decision. The fairness description provides this detailed description on which compatibility with a new case can be gauged.

Label Information

- **Composition Sensitive Attributes.** This label signifies which combinations of sensitive attributes can be handled by the method. An overview of common compositions can be found in Table 5. Binary attributes are a subset of categorical attributes and categorical attributes are a subset of parallel attributes.

Location	Description
Pre-processing	Pre-processing techniques try to transform the data so the underlying discrimination is removed (d'Alessandro, O'Neil, and LaGatta 2017). This data includes both training data and prompts (Chu, Wang, and Zhang 2024).
In-processing	In-processing techniques try to modify and change state-of-the-art learning algorithms to remove discrimination during the model training process (d'Alessandro, O'Neil, and LaGatta 2017). This includes making modifications to the optimization process by adjusting the loss function and incorporating auxiliary modules (Chu, Wang, and Zhang 2024).
Intra-processing	The Intra-processing focuses on mitigating bias in pretrained or fine-tuned models during the inference stage without requiring additional training (Chu, Wang, and Zhang 2024).
Post-processing	Post-processing approaches modify the results generated by the model to mitigate biases (Chu, Wang, and Zhang 2024).

Table 4: Descriptions of pipeline locations by Chu, Wang, and Zhang (2024); d'Alessandro, O'Neil, and LaGatta (2017).

- **Fairness Guarantee.** The fairness guarantee states to what degree a fairness method can guarantee fairness. The three possible types of fairness guarantee for a method can be found in Table 6. A method that provides fairness guarantees fails if the chosen fairness constraint cannot be achieved.
- **Fairness Type.** The fairness type denotes the fairness interpretation that a method can satisfy. Depending on the context different fairness types exist. An example of fairness types can be found in Table 7.
- **Fairness Definition.** Fairness definitions are the mathematical formulas a method can enforce on a pipeline. A fairness definition is associated with a fairness type. A vast assortment of fairness definitions have been proposed in the literature. We refer to the surveys of Caton and Haas (2024); Laclau, Largeron, and Choudhary (2024), and Gallegos et al. (2024) for examples of fairness definitions.
- **Fairness Description** The fairness description provides a more detailed explanation how fairness is envisioned and enforced in the method. This includes the mathematical approach for calculating the fairness difference. It also provides the necessary information surrounding the parameters for tuning fairness. If there are any constraints or relaxations with regards to the sensitive attributes, this must be noted here.

Implementation

BiMi Sheets focus on a specific implementation of a bias mitigation method, an important feature of which being its **Programming Language**. A users' unfamiliarity with a specific language or its incompatibility with an existing pipeline might pose a burden for adoption.

Tied to the programming language are the **Compatible Packages** of a specific implementation. Compatibility with well-known packages greatly improves the usability of an implementation. Practitioners have also noted the compatibility with other packages as an important element when choosing a bias mitigation method (Deng et al. 2022; Lee and Singh 2021).

Label Information

- **Programming Language.** States the programming language in which the method is provided.
- **Compatible Package.** List of packages for which the compatibility with the method implementation has been shown. These can include data related packages such as pandas (pandas development team 2020), folktables (Ding et al. 2024), or model related packages like scikit-learn (Pedregosa et al. 2011), Tensorflow (Developers 2024) or PyTorch (Paszke et al. 2019).
- **Description.** The description includes the limitations of the implementation compared to the method's theoretical capabilities. It further includes practical information for integrating the method into a pipeline, such as the expected data format and hyperparameter information.

Use case

The performance of a method is an important aspect in determining its usability. Performance is often shown through **Use Cases** which show the performance in a wide range of applications. As these use cases are strongly connected to their datasets, we report the datasets on which the method was tested. The description section allows providing initial performance parameters of the method.

Recent work showed that the popular Adult, COMPAS, and German Credit datasets are not appropriate to benchmark bias mitigation methods (Fabris et al. 2022; Ding et al. 2024). The use case labels might help other researchers find appropriate sources.

Label Information

- **Use Case.** The use case labels report the datasets on which the method has been tested, indicating its performance with regards to different tasks. An extensive overview of fairness datasets for a range of different tasks can be found in Fabris et al. (2022).
- **Description.** The description allows reporting the performance of the method on its use cases, indicating its capabilities and allowing for easy comparison with other methods.

Composition	Description
Binary Attribute	The method can only take into account one binary attribute whether an individual belongs to a specific group.
Categorical Attributes	The method can encode categorical attributes. This means that it there are a certain number of groups for which an individual can only belong into one of these groups.
Parallel Attributes	The method can handle a set of categorical attributes, where it aims to enforce fairness along the axis of every categorical attribute.
Numerical Attribute	The method can handle continuous valued attributes.

Table 5: Overview of Compositions of Sensitive Attributes.

Fairness Guarantee	Description
Fairness Guaranteed	The method can guarantee the satisfaction of a fairness constraint. This fairness constraint is often configurable through the method's hyperparameters.
Tunable Fairness Strength	The method provides a hyperparameter where the strength of the fairness intervention relative to other performance such as performance can be set.
No Fairness Guarantee	The method's fairness intervention strength cannot be tuned and no formal fairness guarantee is provided by the method.

Table 6: Overview of the possible Fairness Guarantee a method could provide.

Context	Existing fairness types
Binary classification	Group Fairness, Individual Fairness, Subgroup Fairness (Mehrabi et al. 2021)
	Group Fairness, Individual Fairness, Counterfactual Fairness (Caton and Haas 2024)
Large Language Models	Group Fairness, Individual Fairness (Chu, Wang, and Zhang 2024)
	Embedding-based Fairness, Probability-based Fairness, and Generated Text-based Fairness (Gallegos et al. 2024)
Recommender Systems	Consumer Fairness, Provider Fairness (Ekstrand et al. 2022)
Graphs	Structural Metrics, Representation Fairness, Fair Prediction (Laclau, Largeron, and Choudhary 2024)

Table 7: Examples of fairness types.

Ongoing challenges

Having detailed the structure and content of BiMi Sheets in the previous section, let us now discuss two ongoing challenges: the robustness of attempting to partially standardize labels describing bias limitation methods, and the sheets' adoption in practice. Note that these challenges are not unique to BiMi Sheets but hold for most fairness documentation initiatives.

Standardizing language BiMi Sheets' structure aims to find a balance between harmonizing the language used to describe bias mitigation methods and retaining enough flexibility to account for various types of methods across different domains of machine learning.

In particular, building on taxonomies and terms introduced in surveys, the sheets' proposed structure uses a set of labels to homogenize the lexicon used to characterize the properties of bias mitigation methods. The use of a common set of labels eases method comparison, and could help people less familiar with the broader field of AI fairness in understanding its intricacies.

However, we acknowledge that evolution in the field of AI fairness may require adjusting the labels used in BiMi Sheets. For instance, the introduction of new bias mitigation methods (e.g. the rise of methods providing some explainability in tandem with bias mitigation) may require the adjustment of method type labels; or the range of fairness type labels might require expansion as research attempts to account for a wider range of philosophical fairness notions.

Adoption of BiMi Sheets Besides providing a structured approach for documenting the properties of bias mitigation methods, the value of BiMi Sheets lies in standardizing the communication of these properties, and generally easing method comparisons. Fully realizing this potential would require their widespread adoption by both researchers and practitioners.

As a first step towards that goal, we have created a platform to host BiMi Sheets, which can be found at bimisheet.com. It is populated with 24 BiMi Sheets, and allows the creation of novel sheets to be included in the platform after an acceptance process. The platform includes search capabilities and facilitates the comparison of hosted methods. This platform and the underlying sheets are fully open-source, allowing for other hosting initiatives. In the Appendix you can find an offline version of the BiMi Sheets currently available on the platform.

Besides adoption from individual researchers or practitioners, adoption from fairness frameworks such as AIF360 (Bellamy et al. 2018) and Fairlearn (Weerts et al. 2023) would be beneficial for both parties. The goal of BiMi Sheets is complementary to that of fairness frameworks. This means that integrating with BiMi Sheets could improve the findability of methods provided by fairness frameworks, but also increase the coverage of BiMi Sheets, improving on its usefulness.

Conclusion

Motivated by previous research highlighting the challenges that the diversity of bias mitigation methods poses to both researchers and practitioners, this paper sought to tackle the problem of documentation debt in bias mitigation methods. BiMi Sheets propose to *document* and *structure* bias mitigation methods, by systematically reporting a method's description, pipeline constraints, fairness formalization, implementation details and use cases, empowering researchers and practitioners alike to better understand whether a method is applicable to their context.

Nevertheless, it should be emphasized that *we did not 'solve' the portability trap in fairness*. Documentation and structure alone cannot replace the holistic, interdisciplinary analysis required to understand how bias is best addressed in a decision process. Indeed, a technical solution may not necessarily be required, as also argued by Selbst et al. (2019) who discuss four other 'traps'; and several works point to the inherent limitations of technical solutions to address biases in AI (Wachter, Mittelstadt, and Russell 2021; John-Mathews, Cardon, and Balagué 2022; Buyl and De Bie 2024).

References

Arnold, M.; Bellamy, R. K. E.; Hind, M.; Houde, S.; Mehta, S.; Mojsilovic, A.; Nair, R.; Ramamurthy, K. N.; Reimer, D.; Olteanu, A.; Piorkowski, D.; Tsay, J.; and Varshney, K. R. 2019. FactSheets: Increasing Trust in AI Services through Supplier's Declarations of Conformity. (arXiv:1808.07261). ArXiv:1808.07261 [cs].

Bartl, M.; Mandal, A.; Leavy, S.; and Little, S. 2024. Gender Bias in Natural Language Processing and Computer Vision: A Comparative Survey. *ACM Comput. Surv.* Just Accepted.

Bellamy, R. K. E.; Dey, K.; Hind, M.; Hoffman, S. C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; Mojsilovic, A.; Nagar, S.; Ramamurthy, K. N.; Richards, J.; Saha, D.; Sattigeri, P.; Singh, M.; Varshney, K. R.; and Zhang, Y. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias.

Buolamwini, J.; and Gebru, T. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, 77–91. PMLR.

Buyl, M.; and De Bie, T. 2024. Inherent limitations of AI fairness. *Communications of the ACM*, 67(2): 48–55.

Buyl, M.; Defrance, M.; and Bie, T. D. 2024. fairret: a Framework for Differentiable Fairness Regularization Terms. (arXiv:2310.17256). ArXiv:2310.17256 [cs].

Caton, S.; and Haas, C. 2024. Fairness in Machine Learning: A Survey. *ACM Computing Surveys*, 56(7): 1–38.

Chu, Z.; Wang, Z.; and Zhang, W. 2024. Fairness in Large Language Models: A Taxonomic Survey. *ACM SIGKDD Explorations Newsletter*, 26(1): 34–48.

Cruz, A.; and Hardt, M. 2024. Unprocessing Seven Years of Algorithmic Fairness. In *The Twelfth International Conference on Learning Representations*.

Defrance, M.; Buyl, M.; and Bie, T. D. 2024. ABCFair: an Adaptable Benchmark approach for Comparing Fairness Methods. (arXiv:2409.16965). ArXiv:2409.16965 [cs].

Delaney, E.; Fu, Z.; Wachter, S.; Mittelstadt, B.; and Russell, C. 2024. OxonFair: A Flexible Toolkit for Algorithmic Fairness. (arXiv:2407.13710). ArXiv:2407.13710 [cs].

Deng, W. H.; Nagireddy, M.; Lee, M. S. A.; Singh, J.; Wu, Z. S.; Holstein, K.; and Zhu, H. 2022. Exploring How Machine Learning Practitioners (Try To) Use Fairness Toolkits. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, 473–484. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393522.

Developers, T. 2024. TensorFlow.

Ding, F.; Hardt, M.; Miller, J.; and Schmidt, L. 2024. Retiring adult: new datasets for fair machine learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713845393.

d'Alessandro, B.; O'Neil, C.; and LaGatta, T. 2017. Conscientious Classification: A Data Scientist's Guide to Discrimination-Aware Classification. *Big Data*, 5(2): 120–134. ArXiv:1907.09013 [stat].

Ekstrand, M. D.; Das, A.; Burke, R.; Diaz, F.; et al. 2022. Fairness in information access systems. *Foundations and Trends® in Information Retrieval*, 16(1-2): 1–177.

Fabris, A.; Messina, S.; Silvello, G.; and Susto, G. A. 2022. Algorithmic fairness datasets: the story so far. *Data Mining and Knowledge Discovery*, 36(6): 2074–2152.

Face, H. 2025. The AI community building the future.

Gallegos, I. O.; Rossi, R. A.; Barrow, J.; Tanjim, M. M.; Kim, S.; Dernoncourt, F.; Yu, T.; Zhang, R.; and Ahmed, N. K. 2024. Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics*, 50(3): 1097–1179.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; III, H. D.; and Crawford, K. 2021. Datasheets for Datasets. (arXiv:1803.09010). ArXiv:1803.09010 [cs].

Green, B. 2019. "Good" isn't good enough.

Gursoy, F.; and Kakadiaris, I. A. 2022. System Cards for AI-Based Decision-Making for Public Policy. (arXiv:2203.04754). ArXiv:2203.04754 [cs].

Han, X.; Chi, J.; Chen, Y.; Wang, Q.; Zhao, H.; Zou, N.; and Hu, X. 2024. FFB: A Fair Fairness Benchmark for In-Processing Group Fairness Methods. (arXiv:2306.09468). ArXiv:2306.09468 [cs].

John-Mathews, J.-M.; Cardon, D.; and Balagué, C. 2022. From reality to world. A critical perspective on AI fairness. *Journal of Business Ethics*, 178(4): 945–959.

Laclau, C.; Largeron, C.; and Choudhary, M. 2024. A Survey on Fairness for Machine Learning on Graphs. (arXiv:2205.05396). ArXiv:2205.05396 [cs].

Lee, M. S. A.; and Singh, J. 2021. The Landscape and Gaps in Open Source Fairness Toolkits. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380966.

Li, Y.; Chen, H.; Xu, S.; Ge, Y.; Tan, J.; Liu, S.; and Zhang, Y. 2023. Fairness in recommendation: Foundations, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, 14(5): 1–48.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.*, 54(6): 115:1–115:35.

Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I. D.; and Gebru, T. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, 220–229. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6125-5.

Pagano, T. P.; Loureiro, R. B.; Lisboa, F. V. N.; Peixoto, R. M.; Guimarães, G. A. S.; Cruz, G. O. R.; Araujo, M. M.; Santos, L. L.; Cruz, M. A. S.; Oliveira, E. L. S.; Winkler, I.; and Nascimento, E. G. S. 2023. Bias and Unfairness in Machine Learning Models: A Systematic Review on Datasets, Tools, Fairness Metrics, and Identification and Mitigation Methods. *Big Data and Cognitive Computing*, 7(11): 15.

pandas development team, T. 2020. pandas-dev/pandas: Pandas.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Richardson, B.; Garcia-Gathright, J.; Way, S. F.; Thom, J.; and Cramer, H. 2021. Towards Fairness in Practice: A Practitioner-Oriented Rubric for Evaluating Fair ML Toolkits. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380966.

Saleiro, P.; Kuester, B.; Hinkson, L.; London, J.; Stevens, A.; Anisfeld, A.; Rodolfa, K. T.; and Ghani, R. 2019. Aequitas: A Bias and Fairness Audit Toolkit. (arXiv:1811.05577). ArXiv:1811.05577 [cs].

Selbst, A. D.; Boyd, D.; Friedler, S. A.; Venkatasubramanian, S.; and Vertesi, J. 2019. Fairness and Abstraction in Sociotechnical Systems. FAT* '19, 59–68. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6125-5.

Staufer, L.; Yang, M.; Reuel, A.; and Casper, S. 2025. Audit Cards: Contextualizing AI Evaluations. (arXiv:2504.13839). ArXiv:2504.13839 [cs].

Sylvester, J.; and Raff, E. 2018. What About Applied Fairness? *ArXiv*.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2021. Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI. *Computer Law & Security Review*, 41: 105567.

Wang, A.; Hertzmann, A.; and Russakovsky, O. 2024. Benchmark suites instead of leaderboards for evaluating AI fairness. *Patterns*, 5(11).

Weerts, H.; Dudík, M.; Edgar, R.; Jalali, A.; Lutz, R.; and Madaio, M. 2023. Fairlearn: Assessing and Improving Fairness of AI Systems.

Weerts, H.; Pfisterer, F.; Feurer, M.; Eggensperger, K.; Bergman, E.; Awad, N.; Vanschoren, J.; Pechenizkiy, M.; Bischl, B.; and Hutter, F. 2024. Can fairness be automated? Guidelines and opportunities for fairness-aware AutoML. *Journal of Artificial Intelligence Research*, 79: 639–677.

BiMi Sheet examples

On the following pages we provide 24 offline examples of BiMi Sheets. These methods either originate from well-known fairness toolkits or were proposed recently in top ML conferences. These choice criteria resulted in 16 BiMi Sheets from methods in AIF360 (Bellamy et al. 2018), 4 from FairLearn (Weerts et al. 2023), 1 for error-parity (Cruz and Hardt 2024), 2 for fairret (Buyl, Defrance, and Bie 2024), and 1 for OxonFair (Delaney et al. 2024).

Adversarial Debiasing: BiMi Sheet

Metadata

Name: Adversarial Debiasing
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Mitigating Unwanted Biases with Adversarial Learning* [3]

Method Description

Adversarial Learning | Regularization | Binary Classification | Tabular Datasets

Adversarial debiasing is an in-processing technique that learns a classifier to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions. This approach leads to a fair classifier as the predictions cannot carry any group discrimination information that the adversary can exploit.

Pipeline Architecture

In-Processing | Neural Networks

The method is compatible with any gradient-based learning model, however the current implementation only allows for a neural network with one hidden layer.

Fairness Type

Binary Attribute | Tunable Fairness Strength
Group Fairness | Demographic Parity

The adversarial model tries to predict the sensitive attributes from the prediction of the model. The method tries to prevent this from happening, effectively making the prediction independent of the sensitive attribute, which is equivalent to Demographic Parity.

Implementation Constraints

Python

The model runs a Tensorflow neural network with one hidden layer. Input data has to be formatted as an AIF360 dataset.

Tested Use Cases

Adult [1]

Additional experiments were done in Zhang et al. [3] on the Google analogy data set [2].

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://papers.nips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html`.

[3] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 335–340, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360128. doi: 10.1145/3278721.3278779. URL `https://doi.org/10.1145/3278721.3278779`.

Calibrated Equalized Odds Post-processing: BiMi Sheet

Metadata

Name: Calibrated Equalized Odds Post-processing
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *On fairness and calibration* [4]

Method Description

Calibration | Binary Classification | Dataset Independent

Equalized Odds Post-Processing is a post-processing technique that seeks to attain scores that optimize (a relaxation of) equalized odds while the classifier remains calibrated. This is achieved by changing the probabilities of certain samples to the base rate of the group rather than the classifier's prediction.

Pipeline Architecture

Post-Processing | Calibrated Classifier

Calibrated Equalized Odds Post-processing is compatible with any underlying learner that outputs a calibrated classifier with regards to the sensitive groups.

Fairness Type

Binary Attribute | No Fairness Guarantee
Group Fairness | Equal Opportunity | Predictive Equality | Calibration

The method aims to satisfy either the chosen fairness measure, while maintaining calibration between satisfiers. This is achieved by sacrificing performance for one group. Experimental results show no guarantee on the performance of the method.

Implementation Constraints

Python | scikit-learn

While the concept of the method is compatible with any calibrated classifier, this implementation uses a set model type. The dataset must be passed in the AIF360 used format.

Tested Use Cases

Adult [1] German [2] Compas [3]

The method does not seem to improve the fairness of a model when experimented on Adult.

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[3] Jeff Larson, Julia Angwin, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

[4] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Deterministic Reranking - Conservative: BiMi Sheet

Metadata

Name: Deterministic Reranking - Conservative
Authors: AIF360
Version: 0.6.1   License: Apache 2.0
Proposed in *Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search* [1]

Method Description

Re-ranking | Ranking | Dataset Independent

If there are any groups for which the minimum representation constraint is violated, choose the element with the highest score among those groups. Otherwise, among groups that do not violate the maximum constraint, pick the group that minimizes $\frac{\lceil p_a * k \rceil}{p_a}$. From this group, choose the element with the highest score.

Pipeline Architecture

Post-Processing | Score-based ranker

Deterministic Reranking takes as input a list of output scores (for a each member of a list of candidates) of a ranker. Candidates are by convention assumed to be ordered by decreasing score by the initial ranker.

Fairness Type

Categorical Attributes | Fairness Guaranteed
Group Fairness | Adjusted Demographic Parity

The Deterministic Reranker allows to set a desired proportion p_a per demographic group a. Given k samples, the algorithm ensures that the number of selected candidates from each group n_a satisfies $\lfloor p_a * k \rfloor \leq n_a \leq \lceil p_a * k \rceil$.

Implementation Constraints

Python | scikit-learn

The dataset must be passed in the AIF360 used format.

Tested Use Cases

Synthetic Dataset

The original paper also tested the method on a proprietary dataset.

References

[1] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pages 2221–2231.

Deterministic Reranking - Constrained: BiMi Sheet

Metadata

Name: Deterministic Reranking - Constrained
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search* [1]

Method Description

Re-ranking | Ranking | Dataset Independent

Starting with 0, increase the value of k until the minimum representation constraint is increased for at least one group. If there are more than one such groups, order them according to the descending score of their highest-scoring candidates not yet in the ranking. For each group in the list above: 1. Insert the next candidate from the group to the next empty index in the ranking Swap the candidate towards earlier indices until: Either the score of the candidate in the earlier index is larger, or, 2. Swapping will violate the minimum condition for the group of the candidate in the earlier index.

Pipeline Architecture

Post-Processing | Score-based ranker

Deterministic Reranking takes as input a list of output scores (for a each member of a list of candidates) of a ranker. Candidates are by convention assumed to be ordered by decreasing score by the initial ranker.

Fairness Type

Categorical Attributes | Fairness Guaranteed
Group Fairness | Adjusted Demographic Parity

The Deterministic Reranker allows to set a desired proportion p_a per demographic group a. Given k samples, the algorithm ensures that the number of selected candidates from each group n_a satisfies $\lfloor p_a * k \rfloor \leq n_a \leq \lceil p_a * k \rceil$.

Implementation Constraints

Python | scikit-learn

The dataset must be passed in the AIF360 used format.

Tested Use Cases

Synthetic Dataset

The original paper also tested the method on a proprietary dataset.

References

[1] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pages 2221–2231.

Deterministic Reranking - Greedy: BiMi Sheet

Metadata

Name: Deterministic Reranking - Greedy
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search* [1]

Method Description

Re-ranking | Ranking | Dataset Independent

The attribute value with the highest next score among those that have not yet met their desired proportion is chosen. However, if there are any attribute values for which the desired proportion is about to be violated, the one with the highest next score among them is chosen.

Pipeline Architecture

Post-Processing | Score-based ranker

Deterministic Reranking takes as input a list of output scores (for a each member of a list of candidates) of a ranker. Candidates are by convention assumed to be ordered by decreasing score by the initial ranker.

Fairness Type

Categorical Attributes | Fairness Guaranteed
Group Fairness | Adjusted Demographic Parity

The Deterministic Reranker allows to set a desired proportion p_a per demographic group a. Given k samples, the algorithm ensures that the number of selected candidates from each group n_a satisfies $\lfloor p_a * k \rfloor \leq n_a \leq \lceil p_a * k \rceil$.

Implementation Constraints

Python | scikit-learn

The dataset must be passed in the AIF360 used format.

Tested Use Cases

Synthetic Dataset

The original paper also tested the method on a proprietary dataset.

References

[1] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pages 2221–2231.

Deterministic Reranking - Relaxed: BiMi Sheet

Metadata

Name: Deterministic Reranking - Relaxed
Authors: AIF360
Version: 0.6.1   License: Apache 2.0
Proposed in *Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search* [1]

Method Description

Re-ranking | Ranking | Dataset Independent

If there are any groups for which the minimum representation constraint is violated, choose the element with the highest score among those groups. Otherwise, among groups that do not violate the maximum constraint, pick the group that minimizes $\lceil \frac{\lceil p_a * k \rceil}{p_a} \rceil$. From this group, choose the element with the highest score.

Pipeline Architecture

Post-Processing | Score-based ranker

Deterministic Reranking takes as input a list of output scores (for a each member of a list of candidates) of a ranker. Candidates are by convention assumed to be ordered by decreasing score by the initial ranker.

Fairness Type

Categorical Attributes | Fairness Guaranteed
Group Fairness | Adjusted Demographic Parity

The Deterministic Reranker allows to set a desired proportion p_a per demographic group a. Given k samples, the algorithm ensures that the number of selected candidates from each group n_a satisfies $\lfloor p_a * k \rfloor \leq n_a \leq \lceil p_a * k \rceil$.

Implementation Constraints

Python | scikit-learn

The dataset must be passed in the AIF360 used format.

Tested Use Cases

Synthetic Dataset

The original paper also tested the method on a proprietary dataset.

References

[1] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pages 2221–2231.

Disparate Impact Remover: BiMi Sheet

Metadata

Name: Disparate Impact Remover
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Certifying and Removing Disparate Impact* [3]

Method Description

Transformation | Binary Classification | Tabular Datasets

Disparate Impact Remover changes the values of the labels while preserving ranking.

Pipeline Architecture

Pre-Processing | Model Independent

The original paper tested with Logistic Regression, Support Vector Machines, and Gaussian Naïve Bayes models.

Fairness Type

Binary Attribute | Tunable Fairness Strength
Group Fairness | Balanced Error Rate Parity

The method aims to have the balanced error rate higher than a specific threshold. This follows

Implementation Constraints

Python | scikit-learn

The implementation of the method allows only binary attributes, the original paper states that the method can be applied for parallel attributes. Input data has to be formatted as an AIF360 dataset

Tested Use Cases

Adult [2]

Experiments were conducted on the Ricci Data [1], German credit [4], and Adult income [2] in Feldman et al. [3].

References

[1] Ricci v. destefano, 557 u.s. 557, 2009.

[2] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[3] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783311. URL https://doi.org/10.1145/2783258.2783311.

[4] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

Grid Search Reduction: BiMi Sheet

Metadata

Name: Grid Search Reduction
Authors: AIF360
Version: 0.6.1   License: Apache 2.0
Proposed in *A Reductions Approach to Fair Classification* [1]

Method Description

Constraint Optimization | Binary Classification | Dataset Independent

Exponentiated gradient reduction is an in-processing technique that reduces fair classification to a sequence of cost-sensitive classification problems, returning a randomized classifier with the lowest empirical error subject to fair classification constraints.

Pipeline Architecture

In-Processing | Model Independent

Grid Search Reduction is compatible with any underlying learner that can produce 0-1 predictions.

Fairness Type

Categorical Attributes | Fairness Guaranteed | Group Fairness | Demographic Parity | Equalized Odds | Equal Opportunity | Predictive Equality

A difference bound is passed to the system which details the allowed fairness violation. The difference between two groups will be at most twice the value of this difference bound hyperparameter.

Implementation Constraints

Python | scikit-learn

Input data has to be formatted as an AIF360 dataset. Users are required to provide a classifier or regression model satisfying scikit-learn conventions for the 'fit' and 'predict' methods (wth 0-1 outputs for 'predict').

Tested Use Cases

Adult [2]

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018.

[2] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

GerryFair Classifier: BiMi Sheet

Metadata

Name: GerryFair Classifier
Authors: AIF360
Version: 0.6.1   License: Apache 2.0
Proposed in *Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness* [2] and *An Empirical Study of Rich Subgroup Fairness for Machine Learning* [3]

Method Description

Subgroup Analysis | Calibration | Binary Classification | Tabular Datasets

The algorithm is based on a formulation of subgroup fairness as a zero-sum game between a Learner (the primal player) and an Auditor (the dual player). The Learner is trying to minimize the sum of the prediction error and a fairness penalty term (given by the Lagrangian), and the Auditor is trying to penalize the fairness violation of the Learner by first identifying the subgroup with the greatest fairness violation and putting all the weight on the dual variable corresponding to this subgroup.

Pipeline Architecture

In-Processing | Linear Regression | Support Vector Machines | Decision Trees | Kernel Regression

Fairness Type

Categorical Attributes | Fairness Guaranteed | Group Fairness | Demographic Parity | Equal Opportunity | Predictive Equality

The method stops when the difference in the subgroup's metric with the overall value of the metric weighted with their proportion is smaller than a chosen hyperparameter gamma.

Implementation Constraints

Python | scikit-learn

Tested Use Cases

Adult [1]

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2564–2572. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/kearns18a.html`.

[3] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 100–109, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287592. URL `https://doi.org/10.1145/3287560.3287592`.

Grid Search Reduction: BiMi Sheet

Metadata

Name: Grid Search Reduction
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *A Reductions Approach to Fair Classification*, *Fair Regression: Quantitative Definitions and Reduction-based Algorithms* [1, 2]

Method Description

Constraint Optimization | Binary Classification | Hard Labels | Regression | Dataset Independent

Grid Search Reduction reduces the task to a sequence of cost-sensitive classification or regression problems, returning the deterministic classifier/regressor with the lowest empirical error subject to fairness constraints.

Pipeline Architecture

In-Processing | Model Independent

Grid Search Reduction is compatible with any underlying learner that can produce 0-1 predictions in the classification case, and any regression model in the regression case.

Fairness Type

Categorical Attributes | No Fairness Guarantee | Group Fairness | Demographic Parity | Equalized Odds | Error Rate Parity | True and False Positive Rate Parity | Bounded Group Loss | Moment based fairness definitions

The approach applies to fairness definitions covered by *fairlearn.reductions.Moment* (including demographic parity, equalized odds, error rate parity, true and false positive rate parity for classification; and bounded group loss for regression).

Implementation Constraints

Python | scikit-learn | aif360

Input data has to be formatted as an AIF360 dataset. Users are required to provide a classifier or regression model satisfying scikit-learn conventions for the 'fit' and 'predict' methods. In the classification case, 'predict' should yield 0-1 outputs.

Tested Use Cases

Adult [3] | Law School [4]

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018.

[2] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019.

[3] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[4] Linda Wightman. Lsac national longitudinal bar passage study. lsac research report series.

LFR: BiMi Sheet

Metadata

Name: LFR
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Learning Fair Representations* [4]

Method Description

Adversarial Learning | Binary Classification | Tabular Datasets

LFR maps each individual, represented as a data point in a given input space, to a probability distribution in a new representation space. The aim of this new representation is to lose any information that can identify whether the person belongs to the protected subgroup, while retaining as much other information as possible.

Pipeline Architecture

Pre-Processing | Model Independent

Fairness Type

Binary Attribute | Tunable Fairness Strength
Group Fairness | Individual Fairness | Demographic Parity

The method minimizes an objection which is a combination of demographic parity, information loss in the representation, and accuracy loss of the prediction label. The strength for each of these elements can be tuned.

Implementation Constraints

Python

Input data has to be formatted as an AIF360 dataset.

Tested Use Cases

Adult [1]

Experiments were conducted on Adult [1], German credit [3], and Heritage Health Prize milestone 1 challenge [2] in Zemel et al. [4].

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Phil Brierley, David Vogel, and Randy Axelrod. Heritage provider network health prize round 1 milestone prize how we did it – team 'market makers'.

[3] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[4] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `https://proceedings.mlr.press/v28/zemel13.html`.

MetaFair Classifier: BiMi Sheet

Metadata

Name: MetaFair Classifier
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees* [2]

Method Description

Constraint Optimization | Binary Classification | Tabular Datasets

The MetaFair Classifier reduces classification with linear-fractional constraints to solving a small number of linear classification problems for carefully chosen parameters.

Pipeline Architecture

In-Processing | Bayesian model

The method has an accompanying model implementation used for predictions.

Fairness Type

Categorical Attributes | Tunable Fairness Strength
Group Fairness | Demographic Parity | Predictive Parity

The original model can satisfy any linear-fractional statistic, however only two are implemented here. The difference in fairness measure is noted as a fraction of the measures.

Implementation Constraints

Python | scikit-learn

Input data has to be formatted as an AIF360 dataset.

Tested Use Cases

Adult [1]

Experiments on Adult [1], German credit [3] and COMPAS [4] were conducted in the original paper of Celis et al. [2].

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 319–328, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287586. URL `https://doi.org/10.1145/3287560.3287586`.

[3] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[4] Jeff Larson, Julia Angwin, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

Optimized Preprocessing : BiMi Sheet

Metadata

Name: Optimized Preprocessing
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Optimized Pre-Processing for Discrimination Prevention* [2]

Method Description

Transformation | Binary Classification | Tabular Datasets

Optimized Preprocessing is a preprocessing technique that learns a probabilistic transformation that edits the features and labels in the data with group fairness, individual distortion, and data fidelity constraints and objectives.

Pipeline Architecture

Pre-Processing | Model Independent

The original paper conducted experiments with Logistic Regression and Random Forest. [2]

Fairness Type

Categorical Attributes | Fairness Guaranteed
Group Fairness | Demographic Parity

The fairness guarantee is provided on the dataset, not the model outcome. The method ensures that the fraction between positive rates is smaller than a hyperparameter epsilon, while keeping the distortion on the features under a given threshold.

Implementation Constraints

Python | scikit-learn

The implementation does not use the initialized privileged and unprivileged groups in the initialization but the info on demographic attributes in the pre-processing of the dataset. The Optimizer dict contains the tunable parameters of the function.

Tested Use Cases

Adult [1] | German [3] | COMPAS [4]

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://papers.nips.cc/paper_files/paper/2017/hash/9a49a25d845a483fae4be7e341368e36-Abstract.html`.

[3] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[4] Jeff Larson, Julia Angwin, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

Prejudice Remover: BiMi Sheet

Metadata

Name: Prejudice Remover
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Fairness-aware classifier with prejudice remover regularizer* [2]

Method Description

Regularization | Binary Classification | Dataset Independent

The Prejudice Remover adds a penalty to the learning objective that aims to reduce 'indirect prejudice' - the mutual information between the protected attribute and the model's predictions.

Pipeline Architecture

In-Processing | Logistic Regression

This bias mitigation method is currently only implemented for logistic regression.

Fairness Type

Binary Attribute | Tunable Fairness Strength
Group Fairness | Demographic Parity

The penalization terms aims to reduce 'indirect prejudice', minimizing mutual information between model predictions and sensitive attributes.

Implementation Constraints

Python

Input data has to be formatted as an AIF360 dataset.

Tested Use Cases

Experiments on Adult [1] were conducted in Kamishima et al. [2], it is not certain if this is the same implementation.

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pages 35–50. Springer, 2012.

Reject Option Classification: BiMi Sheet

Metadata

Name: Reject Option Classification
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Decision theory for discrimination-aware classification* [3]

Method Description

Thresholding | Binary Classification | Hard Labels | Dataset Independent

Reject option classification is a postprocessing technique that gives favorable outcomes to unpriviliged groups and unfavorable outcomes to priviliged groups in a confidence band around the decision boundary with the highest uncertainty. This is done by flipping the labels for the samples of these groups close to the decision boundary.

Pipeline Architecture

Post-Processing | Probabilistic Classifier

Reject Option Classification is compatible with any underlying learner that can produce scores of predicted probabilities.

Fairness Type

Binary Attribute | No Fairness Guarantee | Group Fairness
Demographic Parity | Equalized Odds | Equal Opportunity

The critical region size is determined based on optimizing the chosen fairness guarantee.

Implementation Constraints

Python | scikit-learn

Input data has to be formatted as an AIF360 dataset.

Tested Use Cases

Adult [1] | German [2] | Compas [4]

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[3] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th international conference on data mining*, pages 924–929. IEEE, 2012.

[4] Jeff Larson, Julia Angwin, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

Reweighing: BiMi Sheet

Metadata

Name: Reweighing
Authors: AIF360
Version: 0.6.1 License: Apache 2.0
Proposed in *Data Preprocessing Techniques for Classification without Discrimination* [3]

Method Description

Reweighing | Binary Classification | Tabular Datasets

The tuples in the training dataset are assigned weights. By carefully choosing the weights, the training dataset can be made discrimination-free w.r.t. S without having to change any of the labels. The weights on the tuples can be used directly in any method based on frequency counts.

Pipeline Architecture

Pre-Processing | Model which incorporates weigths

The method is effectively model independent, however for the method to be useful then the model must be able to incorporate the weights produced by the method.

Fairness Type

Binary Attribute | No Fairness Guarantee
Group Fairness | Demographic Parity

The reweighing factor is the fraction of the expected frequency to the observed frequency wrt the combination of sensitive attribute and its class.

Implementation Constraints

Python | scikit-learn

Input data has to be formatted as an AIF360 dataset.

Tested Use Cases

Adult [1], German [4], COMPAS [2]

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[3] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33 (1):1–33, October 2012. ISSN 0219-3116. doi: 10.1007/s10115-011-0463-8.

[4] Jeff Larson, Julia Angwin, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

Error-parity: BiMi Sheet

Metadata

Name: Error-parity
Authors: André Cruz and Moritz Hardt
Version: 0.3.11 License: MIT License
Proposed in *Unprocessing Seven Years of Algorithmic Fairness.* [1]

Method Description

Thresholding | Binary Classification | Hard Labels | Dataset Independent

Error-parity sets groupspecific acceptance thresholds so as to **minimize risk while achieving an equality in error rates** across a desired set of groups. It is both simple and computationally efficient. Error-parity achieves exact error rate equality, unlike many preprocessing and inprocessing, which achieve some relaxation of the constraint. The method uses the output scores and returns hard prediction labels.

Pipeline Architecture

Post-Processing | Probabilistic Classifier

Error-parity is compatible with any underlying learner that can produce scores of predicted probabilities.

Fairness Type

Categorical Attributes | Fairness Guaranteed | Group Fairness | Demographic Parity | Equalized Odds | Equal Opportunity | Predictive Equality

Implementation Constraints

Python 3.8-3.12 | scikit-learn | fairlearn

The implementation requires a trained score predictor that takes in samples, X, in shape (num_samples, num_features), and outputs real-valued scores, R, in shape (num_samples,) as the model that feeds into error-parity.

Tested Use Cases

Synthetic | Adult | Folktables

References

[1] André Cruz and Moritz Hardt. Unprocessing seven years of algorithmic fairness. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=jr03SfWsBS`.

Adversarial Mitigation: BiMi Sheet

Metadata

Name: Adversarial Mitigation
Authors: Fairlearn
Version: 0.11.0 License: MIT
Proposed in *Mitigating Unwanted Biases with Adversarial Learning* [2]

Method Description

Adversarial Learning | Classification | Regression | Tabular Datasets

The predictor network is constructed to solve the underlying supervised learning task, without considering fairness, by minimizing the predictor loss. However, to improve fairness, we do not only minimize the predictor loss, but we also want to decrease the adversary's ability to predict the sensitive features from the predictor's predictions (when implementing demographic parity), or jointly from the predictor's predictions and true labels (when implementing equalized odds).

Pipeline Architecture

In-Processing | Neural Networks

The neural network models cannot have an activation function or discrete prediction function on the final layer.

Fairness Type

Categorical Attributes | Numerical Attribute | Tunable Fairness Strength
Group Fairness | Demographic Parity | Equalized Odds

Fairness is achieved by minimizing the adversary's ability to predict the sensitive attribute. This is trade-off with the regular loss of the system, the strength of the fairness intervention can be tuned through a hyperparameter.

Implementation Constraints

Python | scikit-learn | PyTorch | Tensorflow Keras

The data needs to be provided as a 2D-arraylike of floats. The labels can be binary, categorical or float.

Tested Use Cases

Adult [1]

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 335–340, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360128. doi: 10.1145/3278721.3278779. URL `https://doi.org/10.1145/3278721.3278779`.

Correlation Remover: BiMi Sheet

Metadata

Name: Correlation Remover
Authors: Fairlearn
Version: 0.11.0   License: MIT

Method Description

Transformation | Task Independent | Tabular Datasets

Sensitive features can be correlated with non-sensitive features in the dataset. By applying the CorrelationRemover, these correlations are projected away while details from the original data are retained as much as possible (as measured by the least-squares error).

Pipeline Architecture

Pre-Processing | Model Independent

We expect the CorrelationRemover to be most appropriate as a preprocessing step for (generalized) linear models.

Fairness Type

Categorical Attributes | No Fairness Guarantee
Dataset Unbiasing | Removing Dataset Correlations

The degree to which the correlation is removed can be tuned with the α parameter.

Implementation Constraints

Python | scikit-learn | PyTorch | Tensorflow Keras

Tested Use Cases

Diabetes 130-Hospitals [1, 2]

References

[1] Cios Krzysztof DeShazo Jon Clore, John and Beata Strack. Diabetes 130-US Hospitals for Years 1999-2008. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5230J.

[2] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore. Impact of hba1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014(1):781670, 2014. ISSN 2314-6141. doi: 10.1155/2014/781670.

Reductions: BiMi Sheet

Metadata

Name: Reductions
Authors: Fairlearn
Version: 0.11.0 License: MIT
Proposed in *A reductions approach to fair classification.* [1] and
Fair regression: quantitative definitions and reduction-based algorithms. [2]

Method Description

Constraint Optimization | Binary Classification | Regression | Tabular Datasets

A learning reduction takes as input complex examples, transforms them into simpler examples, invokes an appropriate learning algorithm on the simpler examples, then transforms predictions on these simpler examples to a prediction on the complex examples [3]. A fairness constraint is used to transform a binary classification or regression problem to a cost-sensitive problem.

Pipeline Architecture

In-Processing | Probabilistic Classifier

Reductions is compatible with any underlying learner that can produce scores of predicted probabilities.

Fairness Type

Categorical Attributes | Fairness Guaranteed | Group Fairness | Demographic Parity | Equalized Odds | Equal Opportunity | Predictive Equality | Overall Accuracy Equality | Bounded Group Loss

A difference bound is passed to the system which details the allowed fairness violation. The difference between two groups will be at most twice the value of this difference bound hyperparameter.

Implementation Constraints

Python | scikit-learn | Tensorflow Keras

The reduction algorithms in Fairlearn only require a wrapper access to any 'base' learning algorithm, meaning that the 'base' algorithm only needs to implement fit and predict methods, as any standard scikit-learn estimator, but it does not need to have any knowledge of the desired fairness constraints or sensitive features.

Tested Use Cases

Credit Card Clients [4]

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/agarwal18a.html`.

[2] Alekh Agarwal, Miroslav Dudik, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 120–129. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/agarwal19d.html`.

[3] Alina Beygelzimer, Hal Daumé, John Langford, and Paul Mineiro. Learning reductions that really work. *Proceedings of the IEEE*, 104(1):136–147, 2016. doi: 10.1109/JPROC.2015.2494118.

[4] I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C55S3H.

Threshold Optimizer: BiMi Sheet

Metadata

Name: Threshold Optimizer
Authors: Fairlearn
Version: 0.11.0   License: MIT
Proposed in *Equality of Opportunity in Supervised Learning* [2]

Method Description

Thresholding | Binary Classification | Hard Labels | Dataset Independent

For each sensitive feature value, ThresholdOptimizer creates separate thresholds and applies them to the predictions of the user-provided estimator. To decide on the thresholds it generates all possible thresholds and selects the best combination in terms of the objective and the fairness constraints.

Pipeline Architecture

Post-Processing | Probabilistic Classifier

ThresholdOptimizer expects an estimator that provides it with scores. While the output of ThresholdOptimizer is binary, the input is not limited to scores derived from binary classifiers. In fact, real valued input, e.g. from a regressor, provides it with many more options to create thresholds.

Fairness Type

Categorical Attributes | Fairness Guaranteed
Group Fairness | Demographic Parity | Equalized Odds
Equal Opportunity | Predictive Equality | Predictive Parity
False Omission Rate Parity

ThresholdOptimizer is built to satisfy the specified fairness criteria exactly and with no remaining disparity. In many cases this comes at the expense of performance, for example, with significantly lower accuracy. Regardless, it provides an interesting data point for comparison with other models.

Implementation Constraints

Python | scikit-learn

By default, ThresholdOptimizer trains the passed estimator using its fit() method. If prefit is set to True, ThresholdOptimizer does not call fit() on the estimator and assumes that it is already trained.

Tested Use Cases

Diabetes 130-Hospitals [1, 3]

References

[1] Cios Krzysztof DeShazo Jon Clore, John and Beata Strack. Diabetes 130-US Hospitals for Years 1999-2008. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5230J.

[2] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html`.

[3] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore. Impact of hba1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014(1):781670, 2014. ISSN 2314-6141. doi: 10.1155/2014/781670.

Fairret - Projection: BiMi Sheet

Metadata

Name: Fairret - Projection
Authors: Maarten Buyl, MaryBeth Defrance, and Tijl De Bie
Version: 0.1.3 License: MIT
Proposed in *fairret: a Framework for Differentiable Fairness Regularization Terms* [1]

Method Description

Regularization | Binary Classification | Soft Labels | Tabular Datasets | Image Datasets

A FAIRRET quantifies a model's unfairness as a single value that is minimized like any other objective through automatic differentiation.

Pipeline Architecture

In-Processing | Neural Networks

The method is designed to be implemented in neural networks, however any method that uses gradients could be used.

Fairness Type

Parallel Attributes | Numerical Attribute | Tunable Fairness Strength | Group Fairness | Demographic Parity | Equal Opportunity | Predictive Equality | Predictive Parity | False Omission Rate Parity | Overall Accuracy Equality | Treatment Equality | F1-score Equality

The method calculates the loss term by determining the distance of the model to the c-fixed fair set of models. Instead of using the linear-fractional models directly, the measures are set to the average value of the measure in a batch, allowing to transform them to linear constraints. These linear constraints constitute the fair set. The method can account for numerical attributes, however linear behavior is expected.

Implementation Constraints

Python | PyTorch

The package is implemented to work with PyTorch. No experiments were conducted to confirm the performance on image datasets.

Tested Use Cases

Bank [3] CreditCard [4] LawSchool (SEAPHE project)
ACS Datasets [2]

References

[1] Maarten Buyl, Marybeth Defrance, and Tijl De Bie. fairret: a framework for differentiable fairness regularization terms. In *International Conference on Learning Representations*, 2024.

[2] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62: 22–31, June 2014. ISSN 0167-9236. doi: 10.1016/j.dss.2014.03.001.

[4] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2, Part 1):2473–2480, March 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2007.12.020.

Fairret - Violation: BiMi Sheet

Metadata

Name: Fairret - Violation
Authors: Maarten Buyl, MaryBeth Defrance, and Tijl De Bie
Version: 0.1.3 License: MIT
Proposed in *fairret: a Framework for Differentiable Fairness Regularization Terms* [1]

Method Description

Regularization | Binary Classification | Soft Labels | Tabular Datasets | Image Datasets

A FAIRRET quantifies a model's unfairness as a single value that is minimized like any other objective through automatic differentiation.

Pipeline Architecture

In-Processing | Neural Networks

The method is designed to be implemented in neural networks, however any method that uses gradients could be used.

Fairness Type

Parallel Attributes | Numerical Attribute | Tunable Fairness Strength | Group Fairness | Demographic Parity | Equal Opportunity | Predictive Equality | Predictive Parity | False Omission Rate Parity | Overall Accuracy Equality | Treatment Equality | F1-score Equality

A violation FAIRRET uses the proportion to which the fairness measure differs from the c-fixed measure as an added regularization term. This c-fixed measure is the average value for the measure within a batch, as this will be an integer it allows for the linear-fractional fairness measures to be transformed into linear equations. The method can account for numerical attributes, however linear behavior is expected.

Implementation Constraints

Python | PyTorch

The package is implemented to work with PyTorch. No experiments were conducted to confirm the performance on image datasets.

Tested Use Cases

Bank [3] | CreditCard [4] | LawSchool (SEAPHE project) | ACS Datasets [2]

References

[1] Maarten Buyl, Marybeth Defrance, and Tijl De Bie. fairret: a framework for differentiable fairness regularization terms. In *International Conference on Learning Representations*, 2024.

[2] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62: 22–31, June 2014. ISSN 0167-9236. doi: 10.1016/j.dss.2014.03.001.

[4] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2, Part 1):2473–2480, March 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2007.12.020.

OxonFair: BiMi Sheet

Metadata

Name: OxonFair
Authors: E. Delaney, Z. Fu, S. Wachter, B. Mittelstadt, and C. Russell
Version: NA   License: Apache-2.0 license
Proposed in *OxonFair: A Flexible Toolkit for Algorithmic Fairness* [2]

Method Description

Thresholding | Classification | Hard Labels | Inferring Sensitive Attributes | Dataset Independent

OxonFair is a postprocessing approach for enforcing fairness, with support for a wide range of performance metrics and fairness criteria, and support for inferred attributes, i.e., it does not require access to protected attributes at test time. Under the hood, FairPredictor works by adjusting the decision boundary for each group individually. Where groups are not available, it makes use of inferred group membership to adjust decision boundaries.

Pipeline Architecture

Post-Processing | Probabilistic Classifier

Thresholding can be applied to most pretrained ML algorithms, and optimal thresholds can be selected using held-out validation data unused in training.

Fairness Type

Categorical Attributes | No Attributes at Inference | Fairness Guaranteed | Group Fairness | Demographic Parity | Equal Opportunity | Predictive Equality | Predictive Parity | False Omission Rate Parity | Overall Accuracy Equality | F1-score Equality | Min-Max

OxonFair supports any fairness measure (including conditional fairness measures) that can be expressed per group as a weighted sum of True Positives, False Positives, True Negatives and False Negatives. OxenFair selects the solution that best optimizes the objective (performance) while satisfying the constraint (fairness bound).

Implementation Constraints

Python | scikit-learn | AutoGluon [4] | PyTorch | XGBoost [1]

Oxonfair has its own dataloader object. A notebook is available detailing how to add a dataset not yet in OxonFair.

Tested Use Cases

COMPAS [9] | CelebA [10] | Multilingual Twitter corpus [7] | Jigsaw [8] | Myocardial Infarction [5] | XGBoost [1]

Oxonfair has shown to work with a Resnet-50 backbone [6] trained on ImageNet [3], decision trees, random forests, and XGBoost [1].

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, August 2016. Association for Computing Machinery. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL `https://dl.acm.org/doi/10.1145/2939672.2939785`.

[2] Eoin Delaney, Zihao Fu, Sandra Wachter, Brent Mittelstadt, and Chris Russell. Oxonfair: A flexible toolkit for algorithmic fairness. (arXiv:2407.13710), November 2024. doi: 10.48550/arXiv.2407.13710. URL `http://arxiv.org/abs/2407.13710`. arXiv:2407.13710 [cs].

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, page 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. URL `https://ieeexplore.ieee.org/document/5206848`.

[4] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[5] Shulman V.A. Rossiev D.A. Shesternya P.A. Nikulina S.Yu. Orlova Yu.V. Golovenkin, S.E. and V.F. Voino-Yasenetsky. Myocardial infarction complications. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C53P5M.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), page 770–778, June 2016. doi: 10.1109/CVPR.2016.90. URL `https://ieeexplore.ieee.org/document/7780459`.

[7] Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J. Paul. Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, page 1440–1448, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.180`.

[8] Jigsaw. Jigsaw unintended bias in toxicity classification, 2019. URL `https://kaggle.com/jigsaw-unintended-bias-in-toxicity-classification`.

[9] Jeff Larson, Julia Angwin, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

[10] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, page 3730–3738, December 2015. doi: 10.1109/ICCV.2015.425. URL `https://ieeexplore.ieee.org/document/7410782`.