Democratizing the Development of Fair and Effective Machine Learning Systems

Giordano d'Aloisio

Università degli Studi dell'Aquila / Italy

AI-Gap

20th November 2023, L'Aquila

Let's formally define bias and fairness

BIAS

Systematic favoritism or discrimination of an ML model toward individuals based on some features called *sensitive variables* (like race or gender)

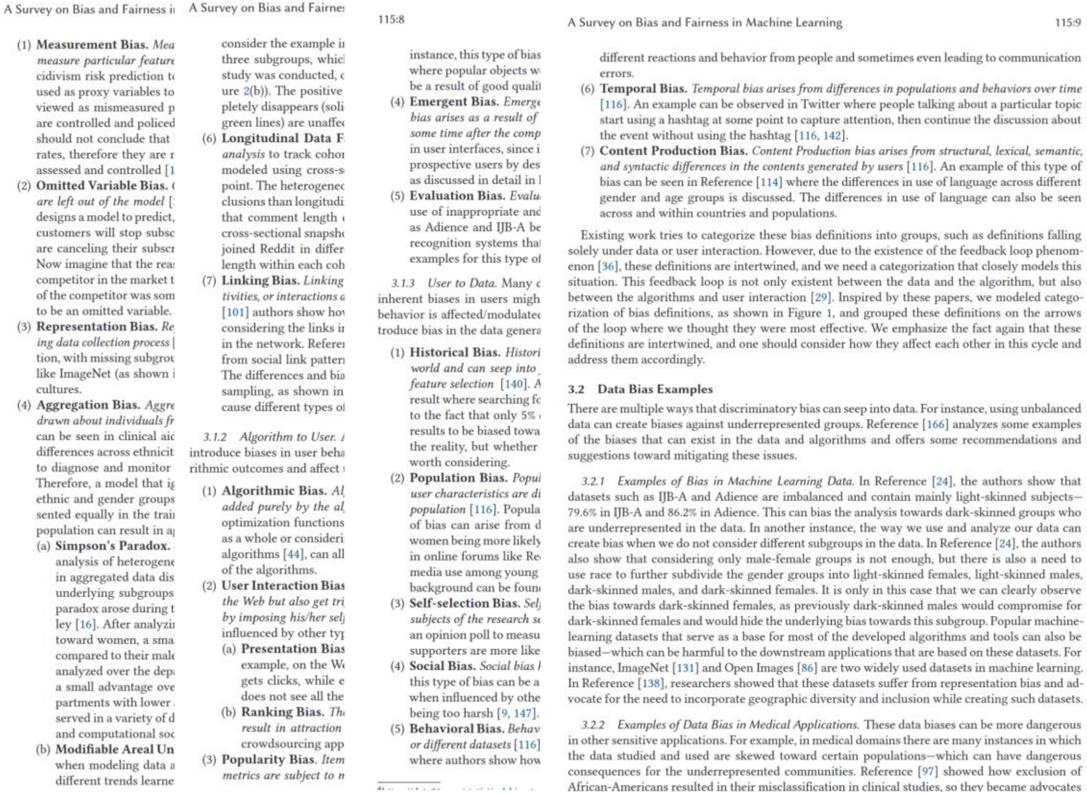
FAIRNESS

Absence of favoritism and discrimination in the predictions of an ML model towards individuals identified by some *sensitive variables* (like race or gender)

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, *54*(6), 1–35. https://doi.org/10.1145/3457607

Is the concept of bias and fairness that simple?

- Actually not...


A Survey on Bias and Fairness in Machine Learning 115:9

different reactions and behavior from people and sometimes even leading to communication errors.

(6) **Temporal Bias.** *Temporal bias arises from differences in populations and behaviors over time* [116]. An example can be observed in Twitter where people talking about a particular topic start using a hashtag at some point to capture attention, then continue the discussion about the event without using the hashtag [116, 142].

(7) **Content Production Bias.** *Content Production bias arises from structural, lexical, semantic, and syntactic differences in the contents generated by users* [116]. An example of this type of bias can be seen in Reference [114] where the differences in use of language across different gender and age groups is discussed. The differences in use of language can also be seen across and within countries and populations.

Existing work tries to categorize these bias definitions into groups, such as definitions falling solely under data or user interaction. However, due to the existence of the feedback loop phenomenon [36], these definitions are intertwined, and we need a categorization that closely models this situation. This feedback loop is not only existent between the data and the algorithm, but also between the algorithms and user interaction [29]. Inspired by these papers, we modeled categorization of bias definitions, as shown in Figure 1, and grouped these definitions on the arrows of the loop where we thought they were most effective. We emphasize the fact again that these definitions are intertwined, and one should consider how they affect each other in this cycle and address them accordingly.

3.2 Data Bias Examples

There are multiple ways that discriminatory bias can seep into data. For instance, using unbalanced data can create biases against underrepresented groups. Reference [166] analyzes some examples of the biases that can exist in the data and algorithms and offers some recommendations and suggestions toward mitigating these issues.

3.2.1 Examples of Bias in Machine Learning Data. In Reference [24], the authors show that datasets such as IJB-A and Adience are imbalanced and contain mainly light-skinned subjects—79.6% in IJB-A and 86.2% in Adience. This can bias the analysis towards dark-skinned groups who are underrepresented in the data. In another instance, the way we use and analyze our data can create bias when we do not consider different subgroups in the data. In Reference [24], the authors also show that considering only male-female groups is not enough, but there is also a need to use race to further subdivide the gender groups into light-skinned females, light-skinned males, dark-skinned males, and dark-skinned females. It is only in this case that we can clearly observe the bias towards dark-skinned females, as previously dark-skinned males would compromise for dark-skinned females and would hide the underlying bias towards this subgroup. Popular machine-learning datasets that serve as a base for most of the developed algorithms and tools can also be biased—which can be harmful to the downstream applications that are based on these datasets. For instance, ImageNet [131] and Open Images [86] are two widely used datasets in machine learning. In Reference [138], researchers showed that these datasets suffer from representation bias and advocate for the need to incorporate geographic diversity and inclusion while creating such datasets.

3.2.2 Examples of Data Bias in Medical Applications. These data biases can be more dangerous in other sensitive applications. For example, in medical domains there are many instances in which the data studied and used are skewed toward certain populations—which can have dangerous consequences for the underrepresented communities. Reference [97] showed how exclusion of African-Americans resulted in their misclassification in clinical studies, so they became advocates

At least 23 different definitions of bias and fairness are available from the literature

From many definitions come many metrics

Generic metrics

`metrics.num_samples` (y_true[, y_pred, ...])	Compute the number of samples.
`metrics.num_pos_neg` (y_true[, y_pred, ...])	Compute the number of positive and negative samples.
`metrics.specificity_score` (y_true, y_pred, *)	Compute the specificity or true negative rate.
`metrics.sensitivity_score` (y_true, y_pred[, ...])	Alias of `sklearn.metrics.recall_score()` for binary classes only.
`metrics.base_rate` (y_true[, y_pred, ...])	Compute the base rate, $Pr(Y = \text{pos\_label}) = \frac{P}{P+N}$.
`metrics.selection_rate` (y_true, y_pred, *[, ...])	Compute the selection rate, $Pr(\hat{Y} = \text{pos\_label}) = \frac{TP+FP}{P+N}$.
`metrics.smoothed_base_rate` (y_true[, y_pred, ...])	Compute the smoothed base rate, $\frac{P+\alpha}{P+N+\lvert R_Y \rvert \alpha}$.
`metrics.smoothed_selection_rate` (y_true, ...)	Compute the smoothed selection rate, $\frac{TP+FP+\alpha}{P+N+\lvert R_Y \rvert \alpha}$.
`metrics.generalized_fpr` (y_true, probas_pred, *)	Return the ratio of generalized false positives to negative examples in the dataset, $GFPR = \frac{GFP}{N}$.
`metrics.generalized_fnr` (y_true, probas_pred, *)	Return the ratio of generalized false negatives to positive examples in the dataset, $GFNR = \frac{GFN}{P}$.

Individual fairness metrics

`metrics.generalized_entropy_index` (b[, alpha])	Generalized entropy index measures inequality over a population.
`metrics.generalized_entropy_error` (y_true, y_pred)	Compute the generalized entropy.
`metrics.theil_index` (b)	The Theil index is the `generalized_entropy_index()` with $\alpha = 1$.
`metrics.coefficient_of_variation` (b)	The coefficient of variation is the square root of two times the `generalized_entropy_index()` with $\alpha = 2$.
`metrics.consistency_score` (X, y[, n_neighbors])	Compute the consistency score.

Group fairness metrics

`metrics.statistical_parity_difference` (y_true)	Difference in selection rates.
`metrics.mean_difference` (y_true[, y_pred, ...])	Alias of `statistical_parity_difference()`.
`metrics.disparate_impact_ratio` (y_true[, ...])	Ratio of selection rates.
`metrics.equal_opportunity_difference` (y_true, ...)	A relaxed version of equality of opportunity.
`metrics.average_odds_difference` (y_true, ...)	A relaxed version of equality of odds.
`metrics.average_odds_error` (y_true, y_pred, *)	A relaxed version of equality of odds.
`metrics.class_imbalance` (y_true[, y_pred, ...])	Compute the class imbalance, $\frac{N_u - N_p}{N_u + N_p}$.
`metrics.kl_divergence` (y_true[, y_pred, ...])	Compute the Kullback-Leibler divergence, $KL(P_p \| P_u) = \sum_y P_p(y) \log\left(\frac{P_p(y)}{P_u(y)}\right)$
`metrics.conditional_demographic_disparity` (y_true)	Conditional demographic disparity, $CDD = \frac{1}{\sum_i N_i} \sum_i N_i \cdot DD_i$
`metrics.smoothed_edf` (y_true[, y_pred, ...])	Smoothed empirical differential fairness (EDF).
`metrics.df_bias_amplification` (y_true, y_pred, *)	Differential fairness bias amplification.
`metrics.between_group_generalized_entropy_error` (...)	Compute the between-group generalized entropy.
`metrics.mdss_bias_scan` (y_true, probas_pred)	DEPRECATED: Change to new interface - aif360.sklearn.detectors.mdss_detector.bias_scan by version 0.5.0.
`metrics.mdss_bias_score` (y_true, probas_pred)	Compute the bias score for a prespecified group of records using a given scoring function.

At least 29 different bias and fairness metrics are available in the AIF360 repository

Bias mitigation methods

`aif360.algorithms.preprocessing`

`algorithms.preprocessing.DisparateImpactRemover` ([...])	Disparate impact remover is a preprocessing technique that edits feature values increase group fairness while preserving rank-ordering within groups [1]_.
`algorithms.preprocessing.LFR` (...[, k, Ax, ...])	Learning fair representations is a pre-processing technique that finds a latent representation which encodes the data well but obfuscates information about protected attributes [2]_.
`algorithms.preprocessing.OptimPreproc` (...[, ...])	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 [3]_.
`algorithms.preprocessing.Reweighing` (...)	Reweighing is a preprocessing technique that Weights the examples in each (group, label) combination differently to ensure fairness before classification [4]_.

`aif360.algorithms.postprocessing`

`algorithms.postprocessing.CalibratedEqOddsPostprocessing` (...)	Calibrated equalized odds postprocessing is a post-processing technique that optimizes over calibrated classifier score outputs to find probabilities with which to change output labels with an equalized odds objective [7]_.
`algorithms.postprocessing.EqOddsPostprocessing` (...)	Equalized odds postprocessing is a post-processing technique that solves a linear program to find probabilities with which to change output labels to optimize equalized odds [8]_ [9]_.
`algorithms.postprocessing.RejectOptionClassification` (...)	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 [10]_.

`aif360.algorithms.inprocessing`

`algorithms.inprocessing.AdversarialDebiasing` (...)	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 [5]_.
`algorithms.inprocessing.ARTClassifier` (...)	Wraps an instance of an `art.classifiers.Classifier` to extend `Transformer`.
`algorithms.inprocessing.GerryFairClassifier` ([...])	Model is an algorithm for learning classifiers that are fair with respect to rich subgroups.
`algorithms.inprocessing.MetaFairClassifier` ([...])	The meta algorithm here takes the fairness metric as part of the input and returns a classifier optimized w.r.t.
`algorithms.inprocessing.PrejudiceRemover` ([...])	Prejudice remover is an in-processing technique that adds a discrimination-aware regularization term to the learning objective [6]_.
`algorithms.inprocessing.ExponentiatedGradientReduction` (...)	Exponentiated gradient reduction for fair classification.
`algorithms.inprocessing.GridSearchReduction` (...)	Grid search reduction for fair classification or regression.

14 bias mitigation methods are available in the AIF360 repository... but many more are available from the literature!

What does it mean?

At least 23 different definitions of bias and fairness

+

At least 29 different bias and fairness metrics

+

At least 14 different bias mitigation algorithms

How can we solve this issue?

- Software engineering approaches can help us to formalise and standardise the development of fair ML systems
- Having a more formal and standard workflow will ease the development of fair ML systems and make it accessible also to non-expert users

To this aim we propose MANILA, a web-based application to *democratize* the development of fair and effective (i.e., correct) ML systems

MANILA

- MANILA is a tool that guides users in defining and performing fairness and effectiveness evaluations of different ML models and fairness enhancing methods
- Automatically disables methods and metrics that are not compatible with other selected features
- Eventually selects and returns the setting having the best fairness and effectiveness trade-off, based on the selected metrics
- Freely available in the SoBigData RI: https://sobigdata.d4science.org/group/sobigdata.it/manila-univaq

SCAN ME

MANILA in action

- We train a Logistic Regression and a Random Forest classifier to predict the recidivism of condemned people using the COMPAS dataset
- We evaluate the fairness and effectiveness of different settings against *non-white* people

SCAN ME

ML Model

Logistic Regression
Random Forest

Fairness Method

No method
Reweighing
DEMV

Metrics

Disparate Impact
Equalized Odds
Accuracy

Aggregation Function

Harmonic Mean

Demo


Thank you for your attention!
UNIVERSITÀ DEGLI STUDI DELL'AQUILA
DISIM
Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica