Bias in Machine Learning

Sarayu V

Contains content from paper titled ‘Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure’, http://hdl.handle.net/1721.1/121101

Outline

- Biases in machine learning algorithms
- A Simple Discrete Choice model to detect bias
- A debiasing algorithm based on VAE to debias training data (Amini et al 2019, ACM conference)

An Example of Bias

- nutrition study- body mass index (BMI) as a function of daily pasta calorie intake
- positive relationship in the population (red solid line)
- Data, especially big data, is often heterogeneous, generated by subgroups with their own characteristics and behaviors. The heterogeneities bias the data.
- data is disaggregated by fitness level

BIAS

- Suppose you have a sample with people only from one fitness group, can you use the same model to make conclusions for the population?

Omitted Variable □ Model misspecification

Aggregation bias effect □ no relationship

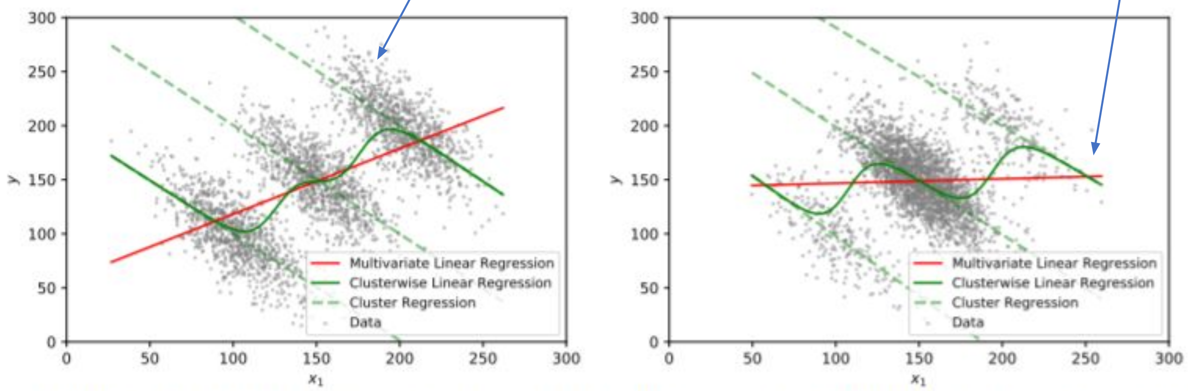


Fig. 1. Illustration of biases in data. Red line shows the regression (MLR) for the entire population, while dashed green lines are regressions for each subgroup, and the solid green line is the unbiased regression. (a) When all subgroups are of equal size, then MLR shows a positive relationship between the outcome and the independent variable. (b) Regression shows almost no relationship in less balanced data. The relationships between variables within each subgroup, however, remain the same. (Credit: Nazanin Alipourfard)

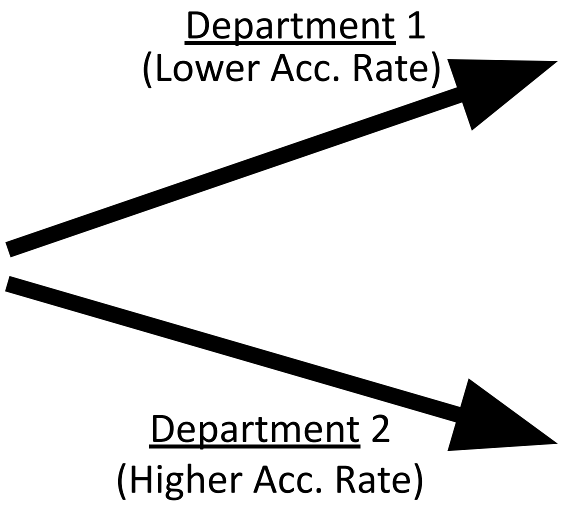
Another Example of Bias: Simpson’s Paradox

According to Simpson’s Paradox, a trend, association, or a characteristic observed in underlying subgroups may be quite different from association or characteristic observed when these subgroups are aggregated.

Gender bias lawsuit in university admissions against UC Berkeley

Aggregate Data

	Women	Men
# applicants	30	120
# admits	4	20
Acc. Rate	13.33%	16.67%



Department 1
(Lower Acc. Rate)

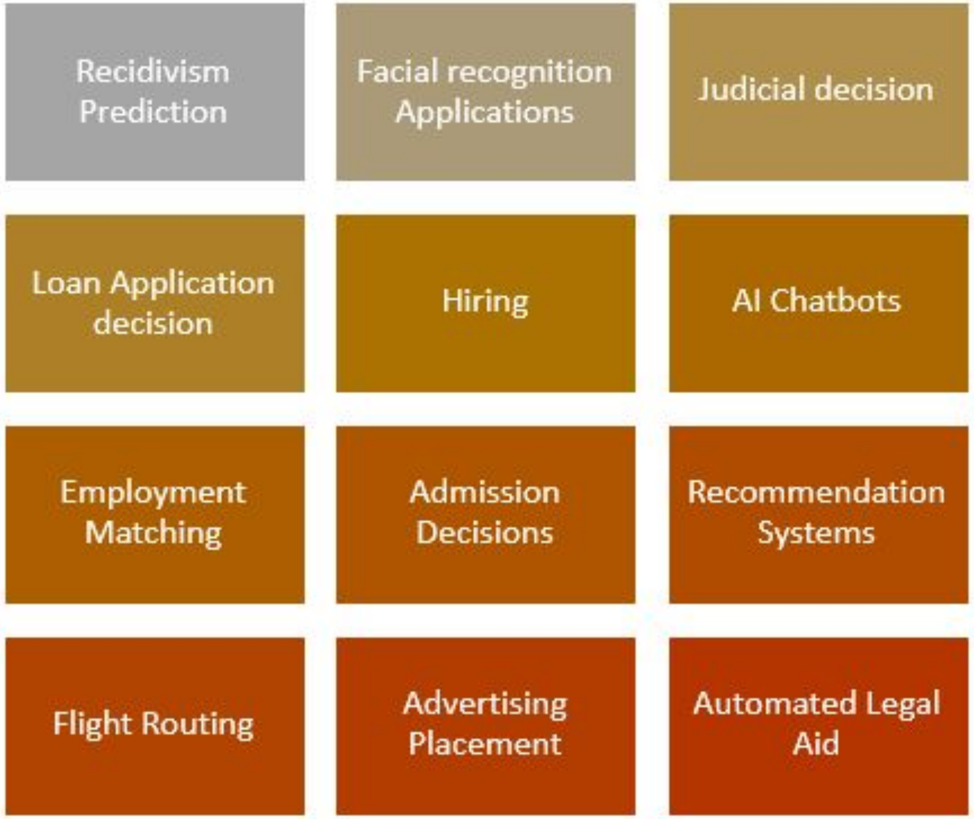
	Women	Men
# applicants	20	40
# admits	2	4
Acc. Rate	10%	10%

Department 2
(Higher Acc. Rate)

	Women	Men
# applicants	10	80
# admits	2	16
Acc. Rate	20%	20%

Why Are AI Applications Becoming Increasingly Popular?

- Benefits of using AI systems (AI-driven algorithms) is that they are highly effective
  1. Productivity improves significantly (machines don’t get tired)
  2. Machines account for many more factors than people can (big data)



Recidivism Prediction

Facial recognition Applications

Judicial decision

Loan Application decision

Hiring

AI Chatbots

Employment Matching

Admission Decisions

Recommendation Systems

Flight Routing

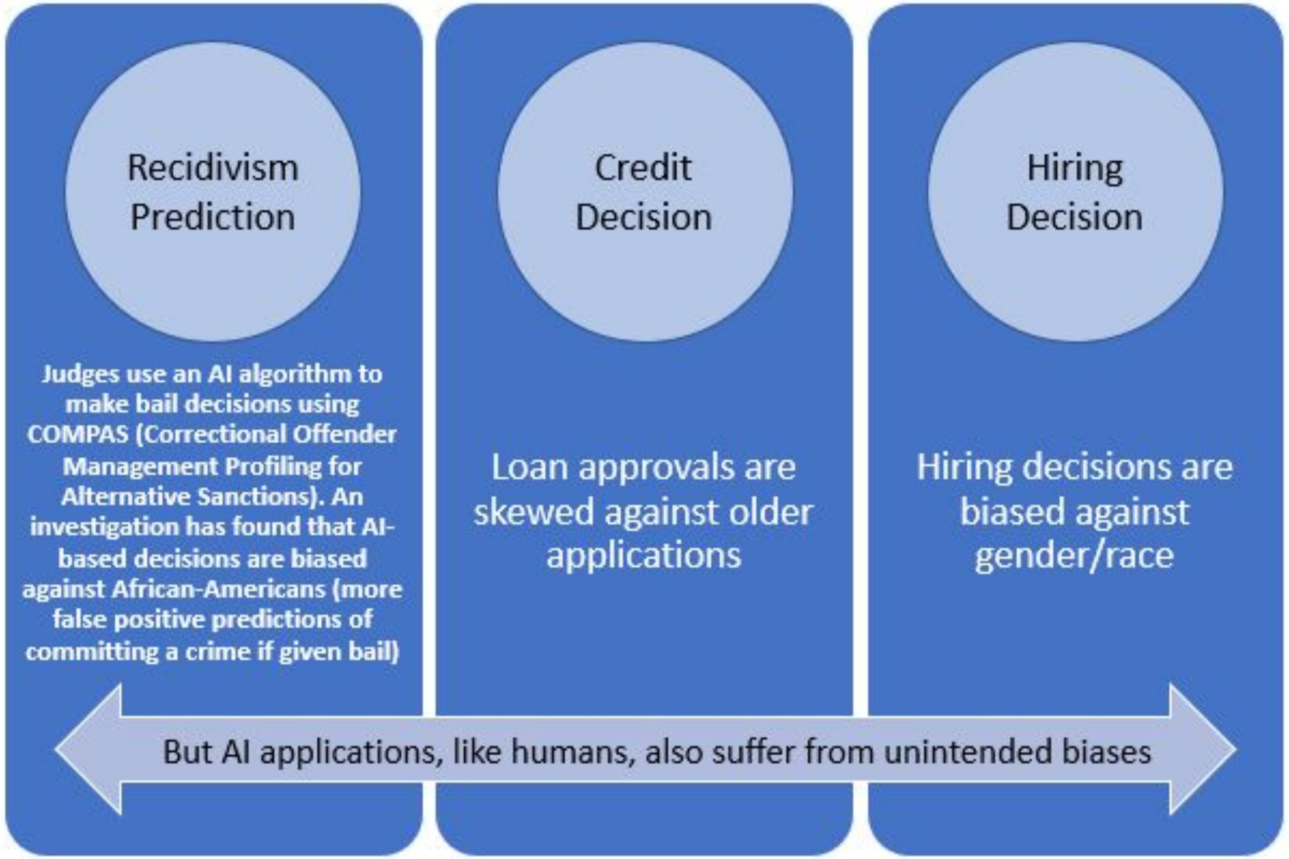
Advertising Placement

Automated Legal Aid

AI Applications Can Also Suffer From Biases

- Many sensitive environments to make important and life-changing decisions and biases outcomes due to these algorithm can have serious consequences

- It is important to take bias and fairness issues into consideration while designing and engineering these types of systems.



Recidivism Prediction

Judges use an AI algorithm to make bail decisions using COMPAS (Correctional Offender Management Profiling for Alternative Sanctions). An investigation has found that AI-based decisions are biased against African-Americans (more false positive predictions of committing a crime if given bail)

Credit Decision

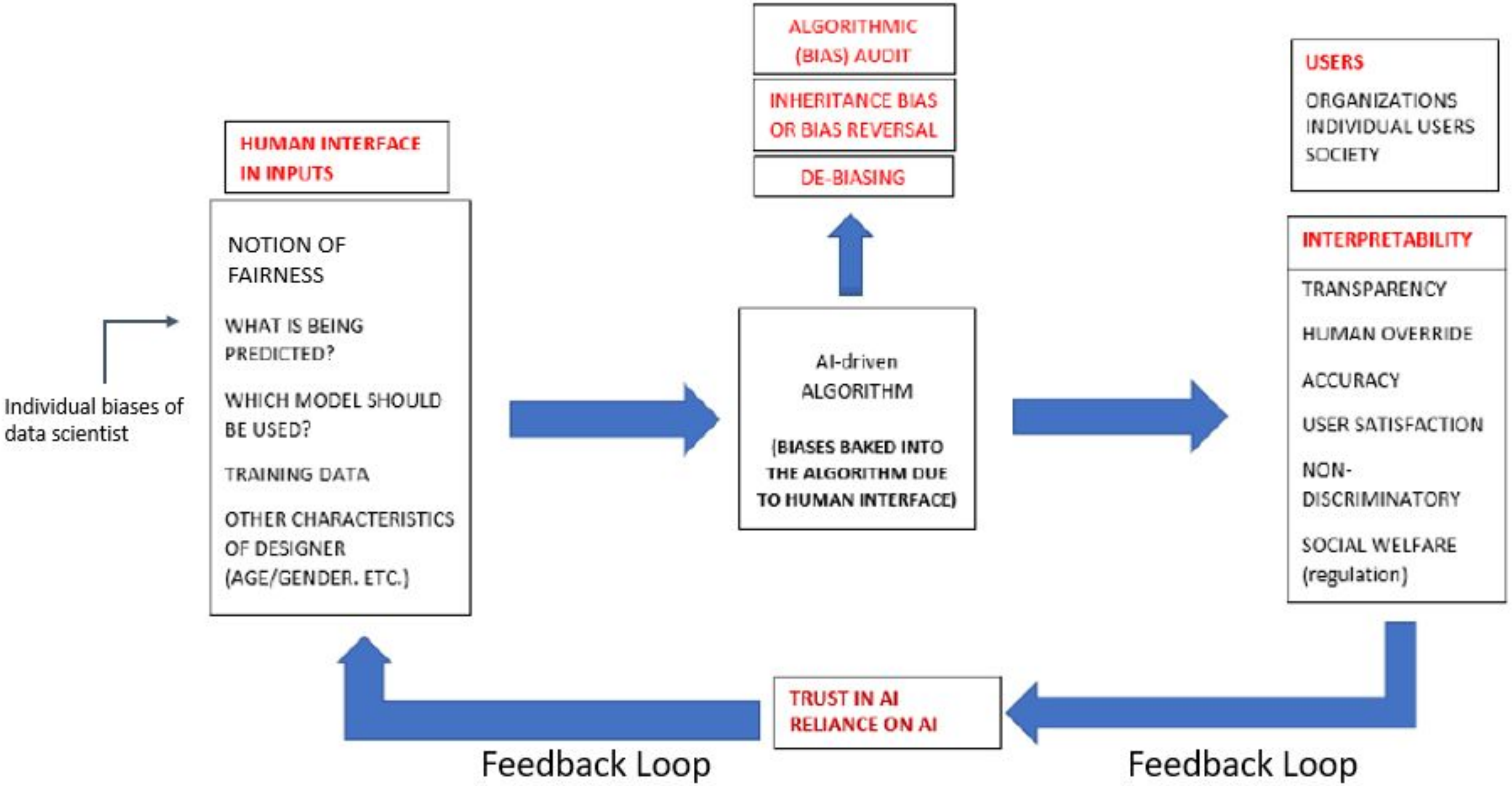
Loan approvals are skewed against older applications

Hiring Decision

Hiring decisions are biased against gender/race

But AI applications, like humans, also suffer from unintended biases

ADOPTION OF AI ALGORITHMS: THE IMPACT OF BIASES



Individual biases of data scientist

HUMAN INTERFACE IN INPUTS

NOTION OF FAIRNESS

WHAT IS BEING PREDICTED?

WHICH MODEL SHOULD BE USED?

TRAINING DATA

OTHER CHARACTERISTICS OF DESIGNER (AGE/GENDER. ETC.)

ALGORITHMIC (BIAS) AUDIT

INHERITANCE BIAS OR BIAS REVERSAL

DE-BIASING

AI-driven ALGORITHM

(BIASES BAKED INTO THE ALGORITHM DUE TO HUMAN INTERFACE)

USERS

ORGANIZATIONS
INDIVIDUAL USERS
SOCIETY

INTERPRETABILITY

TRANSPARENCY

HUMAN OVERRIDE

ACCURACY

USER SATISFACTION

NON-DISCRIMINATORY

SOCIAL WELFARE (regulation)

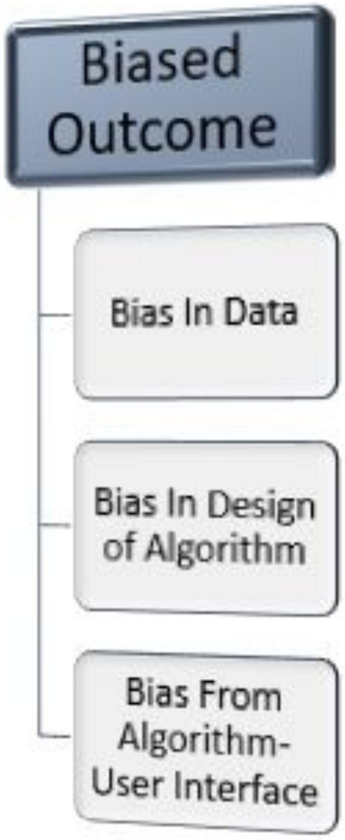
TRUST IN AI
RELIANCE ON AI

Feedback Loop

Feedback Loop

Sources of Bias

- These biases (resulting in unfair outcomes) stem from hidden biases, either in the training data or the design of the algorithm or the user interface
  - It is extremely important for society that AI-driven algorithms are properly tuned to deliver **fair outcomes** and are seen to be fair (**interpretability**)
  - i.e., they do not exhibit biases against sub-groups of the population, i.e., they are non-discriminatory and users can **trust** and **rely** on the system



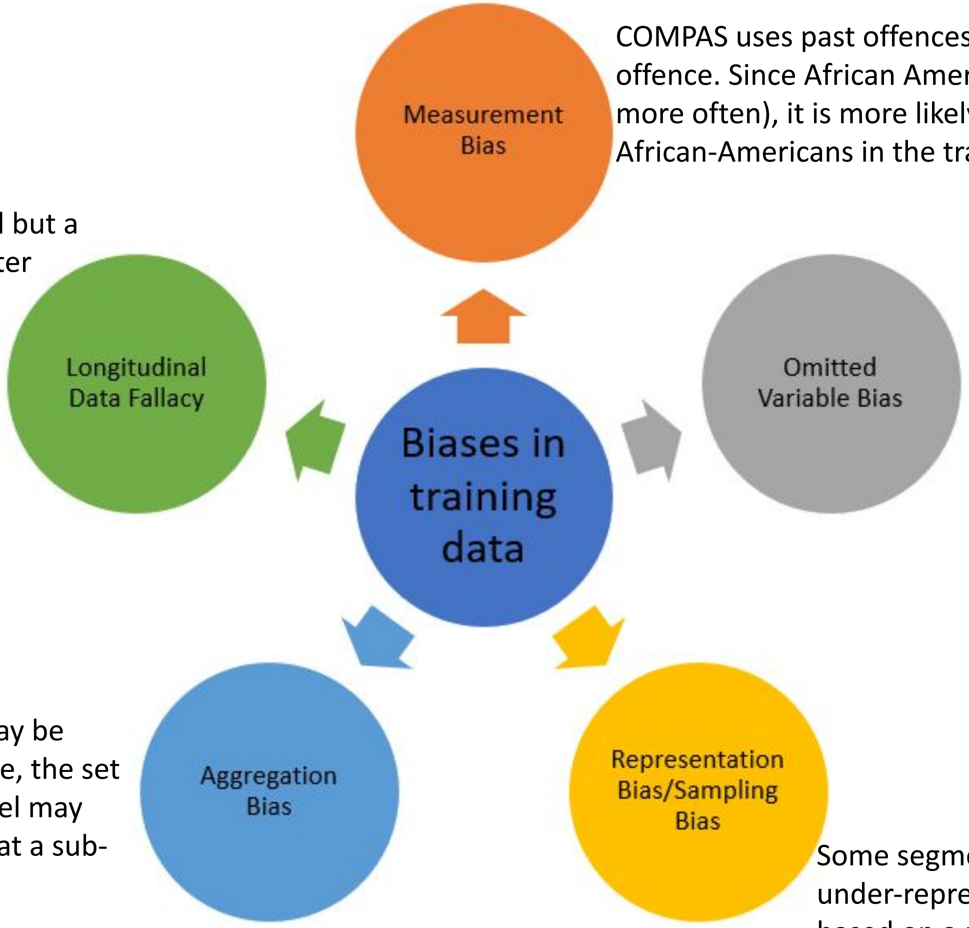
Biased Outcome

Bias In Data

Bias In Design of Algorithm

Bias From Algorithm-User Interface

Biases In Training Data



COMPAS uses past offences to estimate the likelihood of committing a future offence. Since African Americans are more likely to be under scrutiny (searched more often), it is more likely that they will be more occurrences of past offences by African-Americans in the training data. This induces a bias in the estimate.

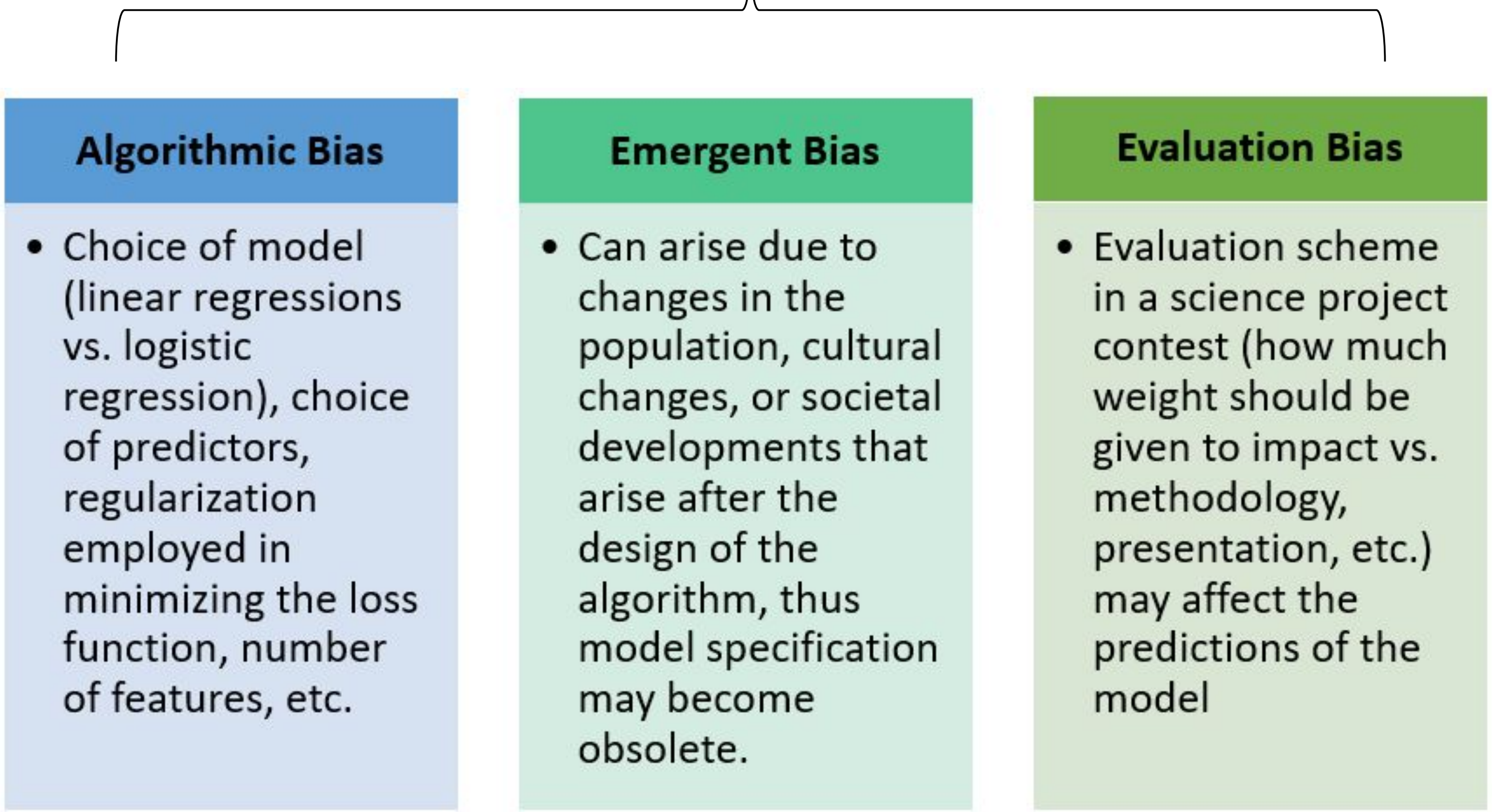
Longitudinal Times Series Analysis may be required but a Cross-Sectional Analysis is conducted. The parameter estimates from the latter will be then biased.

The training data does not include an important feature (for instance, the BMI-pasta intake study ignored the fitness level of individuals in the training date; the estimates are therefore biased).

Models that describes the overall population may be unsuitable for some groups of the population, i.e, the set of features that are relevant at an aggregate level may differ from the set of features that are relevant at a sub-group level.

Some segments of the population are under-represented (the training data is thus not based on a random sample)

Biases in the Design of Algorithms



Algorithmic Bias

- Choice of model (linear regressions vs. logistic regression), choice of predictors, regularization employed in minimizing the loss function, number of features, etc.

Emergent Bias

- Can arise due to changes in the population, cultural changes, or societal developments that arise after the design of the algorithm, thus model specification may become obsolete.

Evaluation Bias

- Evaluation scheme in a science project contest (how much weight should be given to impact vs. methodology, presentation, etc.) may affect the predictions of the model

Biases at the Algorithm-User Interface (with user-generated training data)

Due to the existence of the feedback loop phenomenon, which is a situation in which the trained machine learning model makes decisions that produce outcomes, and these very outcomes affect future data that will be collected for subsequent training rounds or models.

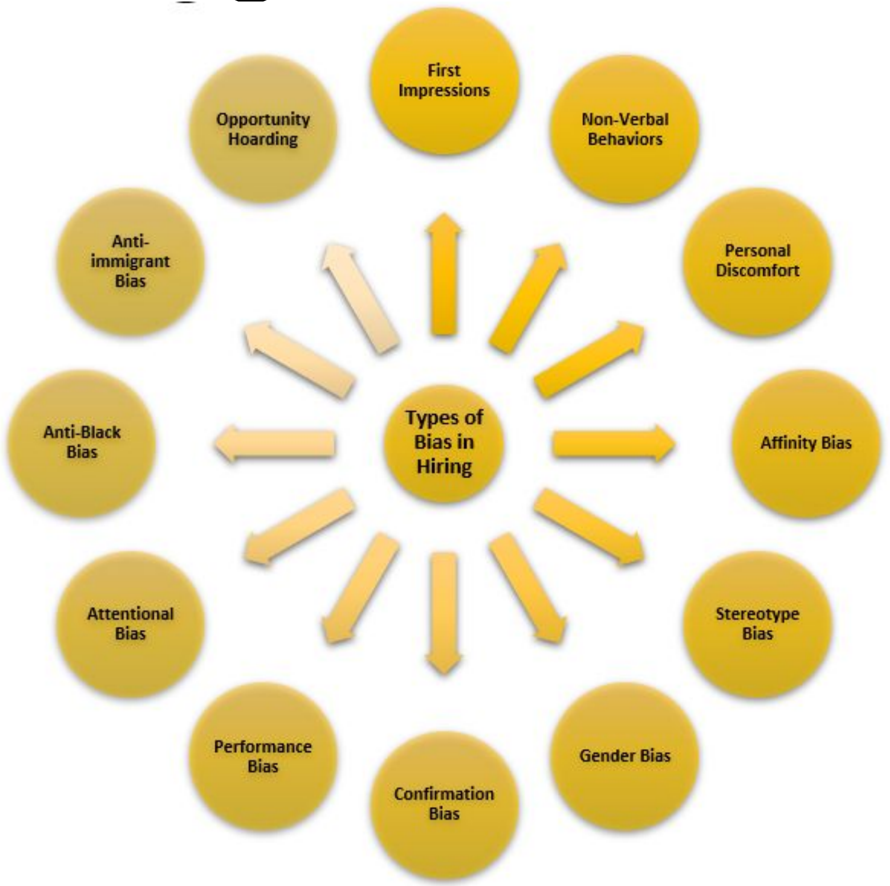
Social bias:
When a users sees that other users have given a high rating to a restaurant, they may be hesitant to voice their own opinion and may prefer to mimic previous user responses.

Self selection bias:
More affected users (with extremely positive of negative opinions) are more likely to respond to a survey, the intermediate opinion individuals often don't respond. Thus, user responses may be skewed in either direction

A Simple Model to Detect Bias: Estimating Affinity Bias in Hiring Decisions

Affinity Bias

The tendency to want to work with someone who is like us culturally, someone we like, and who we can socialize with. Our similarity and comfort level with the candidate can then override our assessment of the candidate's skills and the abilities to do the job.



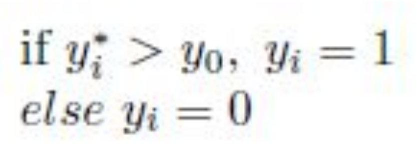
Modeling a Simple Shortlisting decision

Let $\mathbf{y_i}$ represent the shortlisting decision, which can take one of two discrete values (0=not selected, 1=shortlisted/hired).

The agent who makes the shortlisting decision prefers one alternative over the other because it provides
more utility (or a gain measure). Thus, economic logic is absorbed into the classification model by using a latent variable.

A candidate is shortlisted by a recruiter if his productivity exceeds a certain standard or benchmark ($\mathbf{y_0}$). This benchmark acts as a threshold for a positive or negative decision. Consequently, the value $\mathbf{y_i}$ is contingent on $\mathbf{y_i* > y_0}$.

We model productivity (yi*) as a latent variable that is unobserved, but is known to be related to the candidate's educational background, relevant work experience, etc. For convenience, we employ one feature X1, but the model can be extended to include several features



$$\text{if } y_i^* > y_0,\ y_i = 1$$
$$else\ y_i = 0$$

Modeling a Shortlisting decision

$$y_i^* = \beta_0' + \beta_1 X_1 + u$$

ith candidate's productivity **(yi*)** is a linear function of **X$_1$**

intercept

u captures the random error term

β1 is the marginal effect of X$_1$ on productivity

$$\begin{aligned}&\text{if } y_i^* > y_0,\ \ y_i = 1\\ &else\ y_i = 0\end{aligned}$$

Evaluating the Probability of Shortlisting (without Bias)

The probability (p_i) of selection of an i^{th} candidate with characteristic $X_1 = x_1$ is

$$p_i = Prob\ (y_i = 1|X_1 = x_1) = Prob\ (y_i^* > y_0\ |X_1 = x_1)$$

i.e.,

$$\begin{aligned} Prob\ (\beta_0' + \beta_1 X_1 + u > y_0|X_1 = x_1) &= Prob\ (u > [y_0 - (\beta_0' + \beta_1 x_1)]\ ) \\ &= Prob\ (u < [(\beta_0' + \beta_1 x_1) - y_0]\ ) \\ &= F(\beta_0' + \beta_1 x_1 - y_0) \end{aligned}$$

where F is the cumulative distribution function that is assumed to be symmetric, e.g., u could arise from a logistic distribution. Then, the probability of shortlisting is modeled as a logit function, as given below:

$$ln\left(\frac{p_i}{1 - p_i}\right) = E[y_i^*] = \beta_0' + \beta_1 x_1 - y_0$$

$$p_i = \frac{1}{1 + e^{-(\beta_0' + \beta_1 x_1 - y_0)}}$$

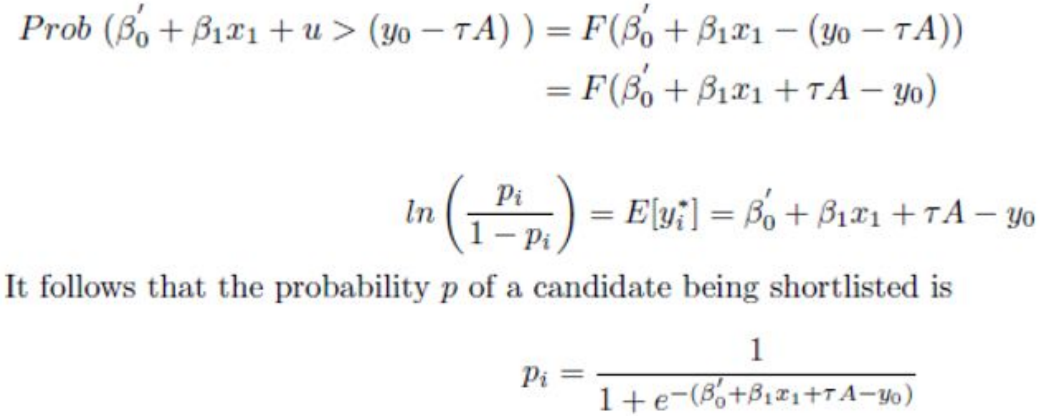
Introducing Affinity Bias in the Model (1)

Overestimate productivity; A=1 -> yi*+ τ

Shortlist if $(y_i^* + \tau) > y_0$; A=1

Shortlist if $y_i^* > (y_0 - \tau)$; A=1

In general, shortlist if yi*$> (y_0 - \tau A)$



$$\begin{aligned} Prob\ (\beta_0' + \beta_1 x_1 + u > (y_0 - \tau A)\ ) &= F(\beta_0' + \beta_1 x_1 - (y_0 - \tau A)) \\ &= F(\beta_0' + \beta_1 x_1 + \tau A - y_0) \end{aligned}$$

$$ln\left(\frac{p_i}{1 - p_i}\right) = E[y_i^*] = \beta_0' + \beta_1 x_1 + \tau A - y_0$$

It follows that the probability p of a candidate being shortlisted is

$$p_i = \frac{1}{1 + e^{-(\beta_0' + \beta_1 x_1 + \tau A - y_0)}}$$

Introducing Affinity Bias in the Model (2)

Let $\beta_0(\tau) = \beta_0' + \tau A - y_0$

Then $ln\left(\frac{p_i}{1-p_i}\right) = \beta_0(\tau) + \beta_1 x_1$

Logit model

probability of a candidate being shortlisted is

$$p_i(\tau) = \frac{1}{1+e^{-(\beta_0(\tau)+\beta_1 x_1)}}$$

The probability of shortlisting is a function of the **affinity bias parameter (τ)**

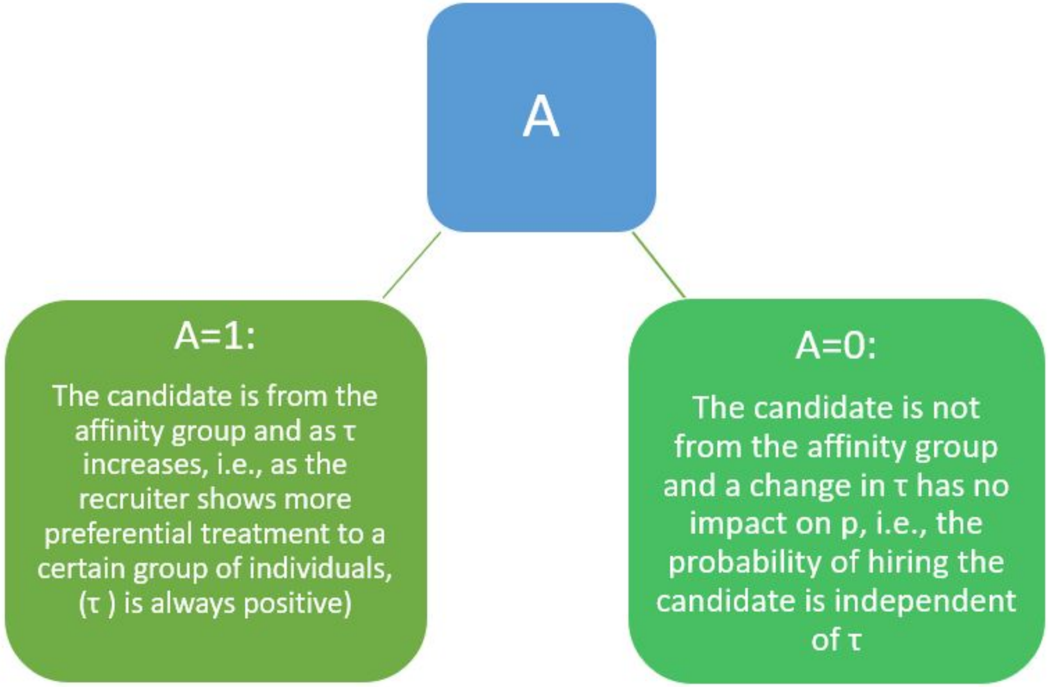
How does Affinity Bias Parameter (τ) Affect the Probability of Shortlisting

Differentiating p(τ), we get

$$p_i'(\tau) = \frac{-1}{[1+e^{-(\beta_0(\tau)+\beta_1 x_1)}]^2}.\ e^{-(\beta_0(\tau)+\beta_1 x_1)}.\ (-\beta_0'(\tau))$$

Since $\beta_0 = \beta_0' + \tau A - y_0$, it follows that $\beta_0'(\tau) = A$ and

$$p_i'(\tau) = \frac{A.\ e^{-(\beta_0(\tau)+\beta_1 x_1)}}{[1+e^{-(\beta_0(\tau)+\beta_1 x_1)}]^2}$$



A=1:

The candidate is from the affinity group and as τ increases, i.e., as the recruiter shows more preferential treatment to a certain group of individuals, (τ) is always positive)

A=0:

The candidate is not from the affinity group and a change in τ has no impact on p, i.e., the probability of hiring the candidate is independent of τ

Note, p(τ) is increasing in τ for individuals in the preferred group, as expected

Estimate τ by running simple logistic regression.

We run a logistic regression with the shortlisting decision (y_i= 1/0) on the feature X_1 for the entire sample (consisting of candidates in the preferred group with A =1 and candidates from the non-preferred group (A = 0).

Response variable y_i
Independent Variables: X_1 and A

Logit model: log (p_i /1-p_i) = $\beta_0 + \beta_1 X_1 + \tau A$
Chi square

Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure

- The problem of severely imbalanced training datasets and the question of how to integrate debiasing capabilities into AI algorithms still remains largely unsolved.
- This paper tackles the challenge of integrating debiasing directly into a model training process that adapts automatically to the shortcoming of the training data.
- The latent structure of the data is learned in an unsupervised manner to uncover hidden and implicit biases in the data

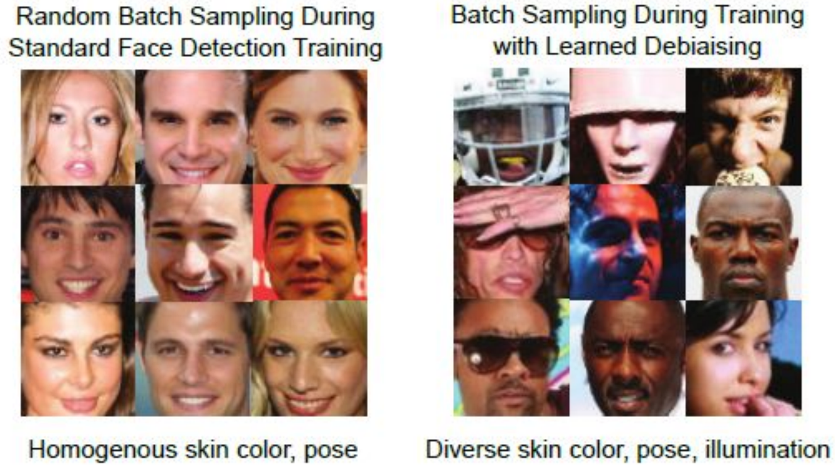

Random Batch Sampling During Standard Face Detection Training
Batch Sampling During Training with Learned Debiaising
Homogenous skin color, pose
Diverse skin color, pose, illumination

Figure 1: **Batches sampled for training without (left) and with (right) learned debiasing.** The proposed algorithm identifies, in an unsupervised manner, under-represented parts of training data and subsequently increases their respective sampling probability. The resulting batch (right) from the CelebA dataset shows increased diversity in features such as skin color, illumination, and occlusions.

Authors: Alexander Amini, Ava Soleimany, Wilko Schwarting, Sangeeta Bhatia and Daniela Rus
Published: Proceedings of the 2019 AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)

Classification Problem: Facial Detection

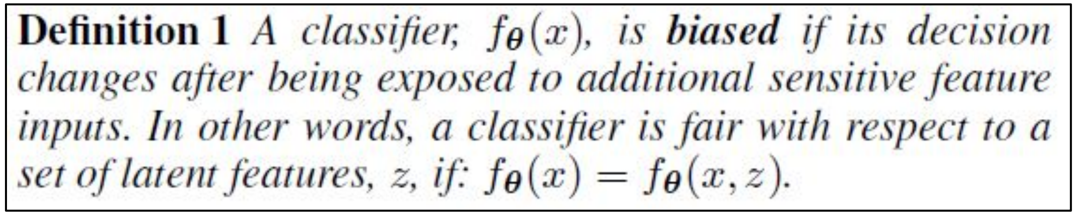
$\mathcal{D}_{train} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{n}$ → $\boldsymbol{\theta}^{*} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\theta})$

features $\boldsymbol{x} \in \mathbb{R}^m$

labels $\boldsymbol{y} \in \mathbb{R}^d$

new point $(\boldsymbol{x}, \boldsymbol{y})$ → $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ where $\hat{\boldsymbol{y}}$ is "close" to $\boldsymbol{y}$.

Each data point x_i → continuous latent vector $z \in \mathbb{R}^{\kappa}$

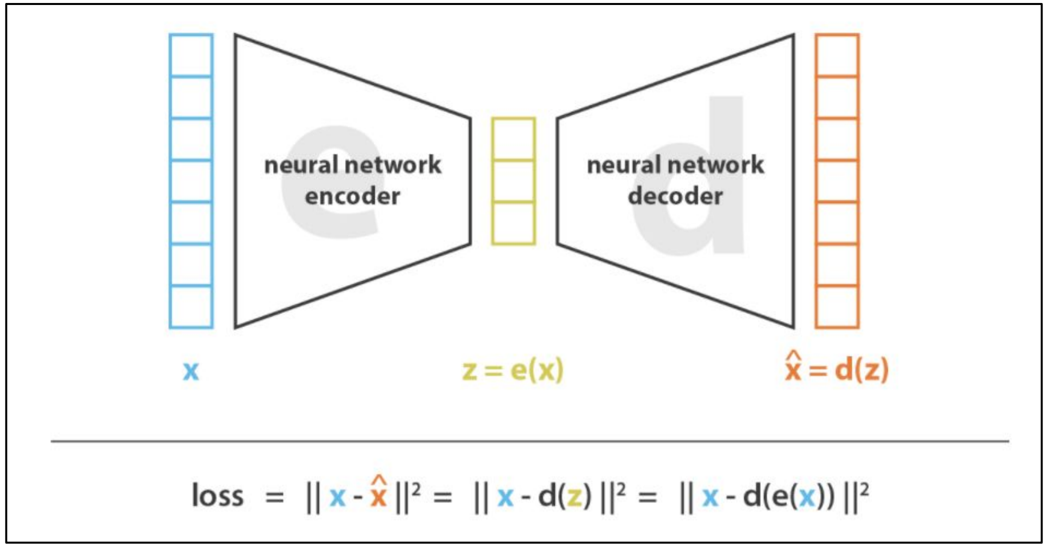


Definition 1 *A classifier, $f_{\boldsymbol{\theta}}(x)$, is* ***biased*** *if its decision changes after being exposed to additional sensitive feature inputs. In other words, a classifier is fair with respect to a set of latent features, z, if: $f_{\boldsymbol{\theta}}(x) = f_{\boldsymbol{\theta}}(x, z)$.*

- **To ensure fairness of classifier across these latent variables, data set should contain roughly uniform samples over the latent space**

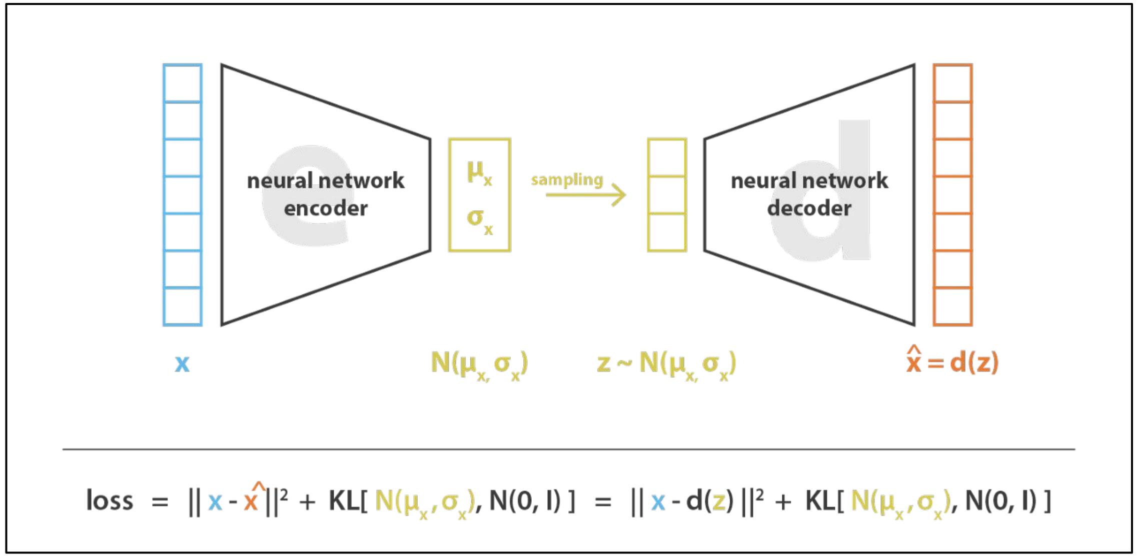
De-biasing Variational Autoencoder…(1)

- Standard Variational Autoencoder



$$\text{loss} = \|x - \hat{x}\|^2 = \|x - d(z)\|^2 = \|x - d(e(x))\|^2$$

autoencoder



$$\text{loss} = \|x - \hat{x}\|^2 + \mathrm{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,] = \|x - d(z)\|^2 + \mathrm{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,]$$

Variational autoencoder

a variational autoencoder can be defined as being an autoencoder whose training is regularized to avoid overfitting and ensure that the latent space has good properties that enable generative process

De-biasing Variational Autoencoder…(2)

- Learn the latent variables of the class in an entirely unsupervised manner and proceed to adaptively resample the dataset while training

- The encoder portion of the VAE learns an approximation q_Φ $(z|x)$ of the true distribution of the latent variables given a data point

- A decoder network mirroring the encoder is then used to reconstruct the input back from the latent space by approximating $p_\theta(x|z)$

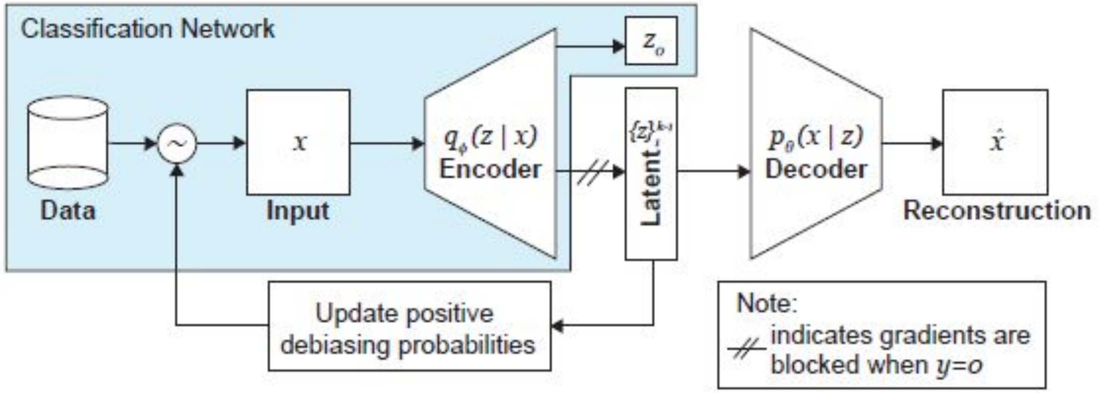
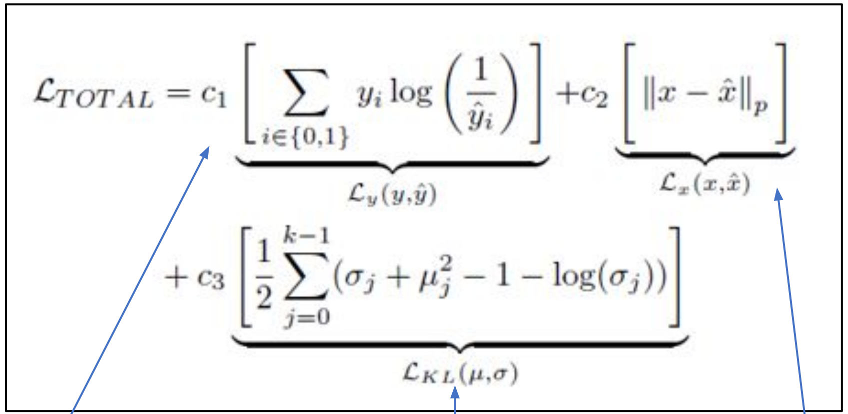


Figure 2: **Debiasing Variational Autoencoder.** Architecture of the semi-supervised DB-VAE for binary classification (blue region). The unsupervised latent variables are used to adaptively resample the dataset while training.

De-biasing Variational Autoencoder…(3)



$$\mathcal{L}_{TOTAL} = c_1 \underbrace{\left[\sum_{i \in \{0,1\}} y_i \log\left(\frac{1}{\hat{y}_i}\right) \right]}_{\mathcal{L}_y(y,\hat{y})} + c_2 \underbrace{\left[\|x - \hat{x}\|_p \right]}_{\mathcal{L}_x(x,\hat{x})}$$

$$+ c_3 \underbrace{\left[\frac{1}{2} \sum_{j=0}^{k-1} (\sigma_j + \mu_j^2 - 1 - \log(\sigma_j)) \right]}_{\mathcal{L}_{KL}(\mu,\sigma)}$$

Supervised latent loss

KL Divergence

Reconstruction loss

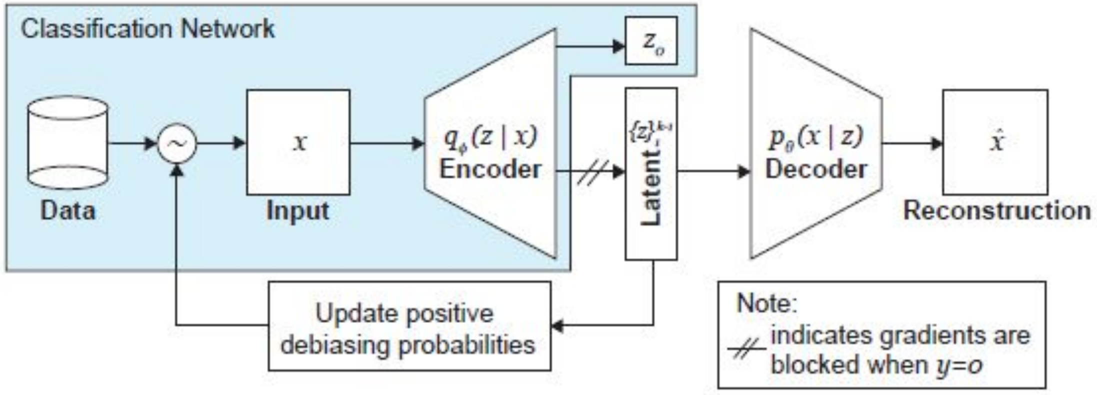


Figure 2: **Debiasing Variational Autoencoder.** Architecture of the semi-supervised DB-VAE for binary classification (blue region). The unsupervised latent variables are used to adaptively resample the dataset while training.

De-biasing Variational Autoencoder…(4)

- **Goal**
  1. **By reducing the over-represented regions of the latent space according to frequency of occurrence, we increase the probability of selecting rarer data for training.**
  2. **This is done adaptively as the latent variables themselves are being learned during training.**

- The training dataset is fed through the encoder network, which provides an initial estimate Q(z|X) of the latent distribution.

- approximate the distribution of the latent space with a histogram $\hat{Q}(z|X) \propto \prod_i \hat{Q}_i(z_i|X)$

Independent histogram for each latent variable z_i

Resampling frequency is inversely related to the histogram frequency

$$\boxed{\mathcal{W}(z(x)|X) \propto \prod_i \frac{1}{\hat{Q}_i(z_i(x)|X) + \alpha}}$$

The probability distribution of selecting a data point x

Debiasing parameter

As α->∞
- no debiasing
- subsampled training set -> original training dataset

As α->0
- Debiasing
- subsampled training set -> uniform over latent variables

Results...(1)

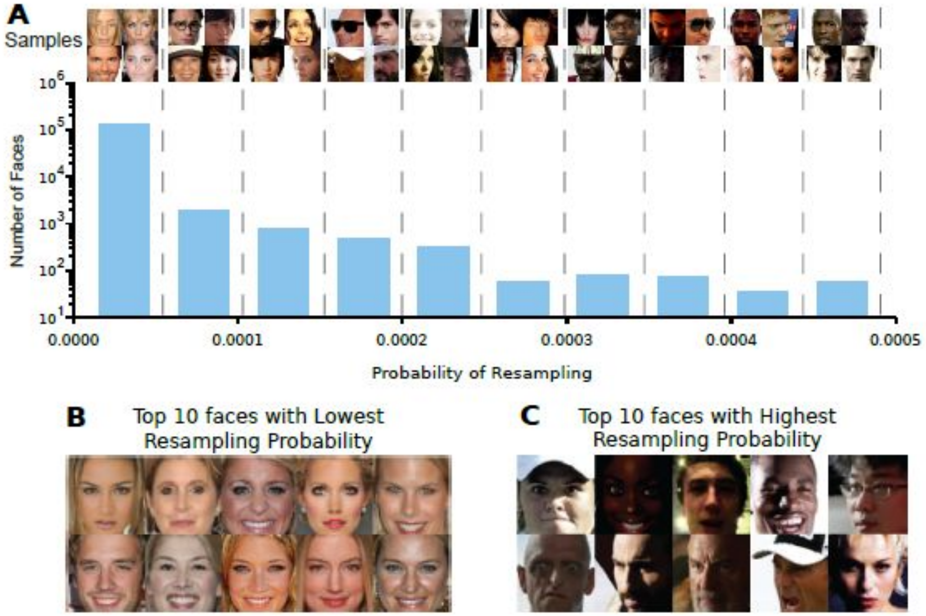


Figure 4: **Sampling probabilities over the training dataset.** Histogram over resampling probabilities showing four samples from each bin (**A**). The top ten faces with the lowest (**B**) and highest (**C**) probabilities of being sampled.

Results…(2)

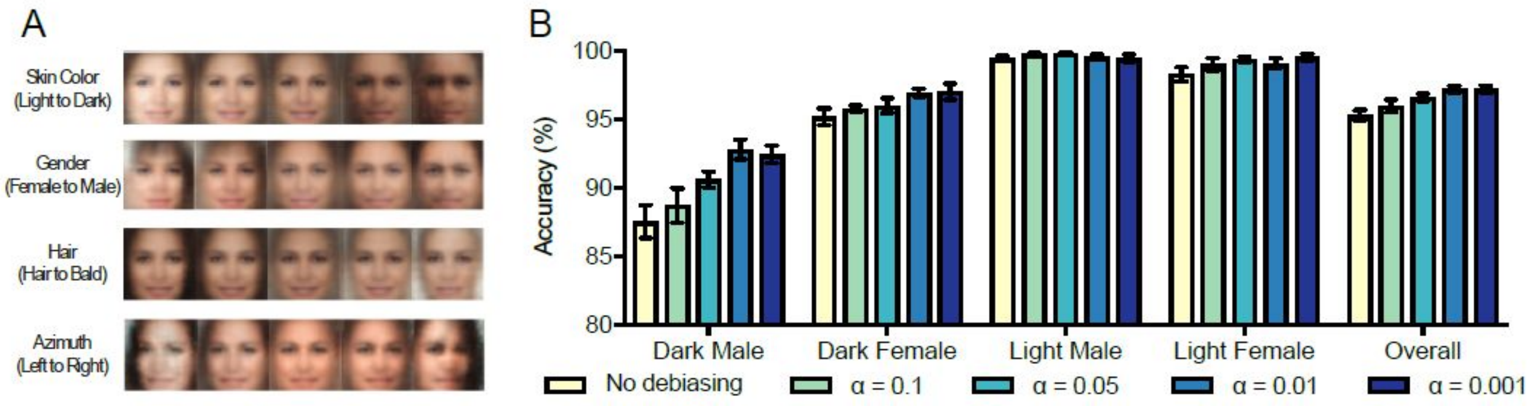


Figure 5: **Increased performance and decreased categorical bias with DB-VAE.** The model learns latent features such as skin color, gender, hair (**A**) and demonstrates increased performance and decreased categorical bias with learned debiasing (**B**).

- Measure overall accuracy of classifier -> mean accuracy over all sensitive categories
- Measure bias of the classifier-> variance in accuracies across all realizations of these categories

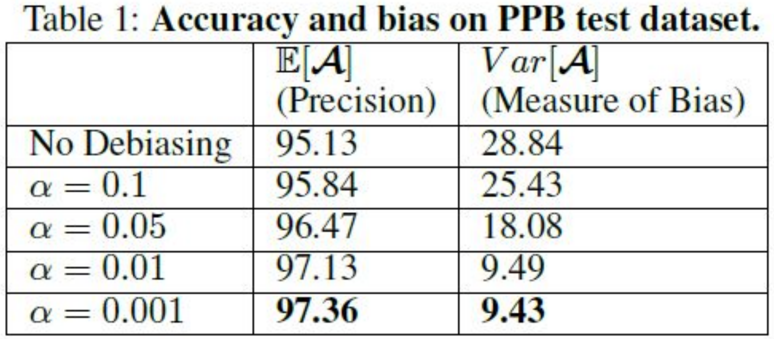


Table 1: **Accuracy and bias on PPB test dataset.**

	$\mathbb{E}[\boldsymbol{\mathcal{A}}]$ (Precision)	$Var[\boldsymbol{\mathcal{A}}]$ (Measure of Bias)
No Debiasing	95.13	28.84
$\alpha = 0.1$	95.84	25.43
$\alpha = 0.05$	96.47	18.08
$\alpha = 0.01$	97.13	9.49
$\alpha = 0.001$	**97.36**	**9.43**

Thank You