Navigating Algorithmic Fairness: Exploring and Addressing Bias through Latent Structures

Nirmal Solanki
Indian Institute of Technology,
Mumbai, India

***Abstract*—Recent investigations have underscored the susceptibilities of contemporary machine learning systems to bias, particularly in relation to demographics that lack representation in the training data. This study introduces an innovative and adjustable algorithm crafted to alleviate concealed biases within training data, which may be unidentified. Our approach integrates the original learning task with a variational autoencoder, enabling the discovery of latent structures within the dataset. Subsequently, it dynamically utilizes the acquired latent distributions to recalibrate the significance of specific data points during the training process. While our methodology is versatile and applicable to diverse data modalities and learning tasks, our focus in this research is on mitigating racial and gender biases in facial detection systems. We assess the effectiveness of our algorithm using the dataset designed explicitly for evaluating biases in computer vision systems. Our findings showcase enhanced overall performance and a reduction in categorical bias through our debiasing strategy.**

1. Introduction

Machine learning (ML) systems are increasingly shaping decisions that have wide-ranging implications for individuals and society. They influence processes like loan eligibility assessments, criminal sentencing, news presentation sequencing, and medical diagnoses. The development and deployment of fair and unbiased AI systems are crucial to prevent unintended consequences and foster long-term acceptance of these algorithms. Facial recognition, once considered a straightforward task, has exposed significant algorithmic biases, particularly affecting certain demographics. Ensuring the fairness of these systems is vital, especially in applications such as law enforcement's face detection, where disparities in accuracy among different groups raise concerns. This issue becomes more complex when facial recognition systems are integrated into broader surveillance or criminal detection pipelines.

In this work, we address the intricate task of seamlessly integrating debiasing capabilities into the model training process. Our approach employs an end-to-end deep learning algorithm that not only learns the intended task, such as facial detection, but also grasps the underlying latent structure of the training data. By autonomously and unsupervisedly learning latent distributions, our algorithm adeptly uncovers hidden biases within the training data. Leveraging a variational autoencoder (VAE) as its foundation, our algorithm identifies under-represented examples in the training dataset and adjusts the sampling probabilities during training, thereby mitigating biases (Fig. 1).

We showcase the effectiveness of our algorithm in debiasing a facial detection system trained on a biased dataset. Additionally, our algorithm provides insights into the learned latent variables actively debiased against. To assess its performance, we compare the debiased model against a standard deep learning classifier, evaluating racial and gender bias using the Pilot Parliaments Benchmark (PPB) dataset (Buolamwini and Gebru 2018).

The key contributions of this work are threefold:
1. Introducing a novel and customizable debiasing algorithm that employs learned latent variables to dynamically adjust sampling probabilities during training.
2. Proposing a semi-supervised model capable of simultaneously learning a debiased classifier and the underlying latent variables governing the given classes.
3. Conducting a comprehensive analysis of our method in the context of facial detection with biased training data, and evaluating its fairness across race and gender on the PPB dataset.

The structure of this paper is organized as follows: we provide a brief overview of related work in Sec. 2, articulate the model and debiasing algorithm in Sec. 3, present experimental results in Sec. 4, and conclude with final remarks in Sec. 5.

2. Methodology

The task at hand involves binary classification based on a dataset $D_{\text{train}} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, where $x \in \mathbb{R}^m$ represents features, and $y \in \mathbb{R}^d$ denotes labels. The objective is to find a parameterized functional mapping $f : X \to Y$ represented by θ that minimizes the loss $L(\theta)$ over the entire training dataset. The optimization problem is formalized as:

$$\theta^* = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L_i(\theta) \quad (1)$$

In the context of a new test example (x, y), the classifier should ideally output $= \mathrm{f}_\theta(x)$, where is close to y,

considering the original loss function. Each data point is associated with a continuous latent vector $z \in \mathbb{R}^k$ capturing hidden, sensitive features.

A classifier $f_\theta(x)$ is considered biased if its decision changes after exposure to additional sensitive feature inputs. Formally, a classifier is fair concerning a set of latent features z if $f_\theta(x) = f_\theta(x, z)$. For instance, in facial detection, skin color, gender, and age are latent variables that should not impact the classifier's decision.

To ensure fairness across latent variables, the dataset should have roughly uniform samples over the latent space. This implies that the training distribution should not be biased to overrepresent a particular category while underrepresenting others. Fairness, in this context, is not just about balancing classes but also ensuring balance within a single class over unobserved latent variables.

The bias of a classifier can be measured by computing its accuracy across different sensitive categories. The overall accuracy is the mean accuracy over all sensitive categories, while bias is the variance in accuracies across realizations of these categories. If a classifier performs equally well across different realizations of a specific latent variable, it is considered unbiased; otherwise, it exhibits bias.

To address potential biases, the paper proposes an unsupervised approach to learning latent variables, avoiding manual annotation and potential human bias. The architecture used for learning these latent variables is outlined in the subsequent subsection.

Unsupervised Learning of Latent Variables and Adaptive Resampling with Debiasing-VAE

In this investigation, we tackle the challenge of autonomously learning latent variables and employing them for dynamic dataset resampling during training. To achieve this, we introduce an extension of the variational autoencoder (VAE) architecture known as debiasing-VAE (DB-VAE). Unlike conventional VAEs, our model incorporates supplementary output variables, transforming it into a semi-supervised framework.

The encoder of the DBVAE acquires an approximation, $q_\phi(x|z)$, of the latent variable distribution, where z denotes the latent variables, and $\hat{y}$ encompasses additional output variables. By explicitly supervising a subset of output variables, we convert the VAE into a semi-supervised model while still retaining unsupervised learning of other latent variables.

The decoder, mirroring the encoder, reconstructs the input from the latent space. Throughout training, we employ reparameterization to facilitate unsupervised learning of latent variables. The loss function amalgamates a supervised latent loss, a reconstruction loss, and a latent loss for unsupervised variables. The overall loss is a weighted sum of these components, allowing for flexible adjustment of their relative significance.

The loss function for the proposed debiasing-VAE (DB-VAE) is composed of three key components, contributing to a comprehensive objective during training:

Supervised Latent Loss (Ly) :

$$L_y(y, \hat{y}) = \sum_{i \in \{0,1\}} y_i \log\left(\frac{1}{\hat{y}_i}\right)$$

Reconstruction Loss (Lx) :

$$L_x(x, \hat{x}) = \|x - \hat{x}\|_p$$

Unsupervised Latent Loss (LKL) :

$$L_{KL}(\mu, \sigma) = \frac{1}{2} \sum_{j=0}^{k-1} (\sigma_j + \mu_j^2 - 1 - \log(\sigma_j))$$

Here, y represents the ground truth, $\hat{y}$ is the predicted output, x is the input, and $\hat{x}$ is the reconstructed output. The latent variables are denoted by μ (mean) and σ (standard deviation). The loss function L_{TOTAL} is a weighted combination of these components:

$$L_{TOTAL} = c_1 \cdot L_y + c_2 \cdot L_x + c_3 \cdot L_{KL}$$

The coefficients c_1, c_2, and c_3 allow for the adjustment of the relative importance of each loss component during training. This comprehensive loss function guides the training process, ensuring effective debiasing and supervised learning of latent variables in the DBVAE.

For comparison, we introduce a baseline model lacking unsupervised latent variables and a decoder network, trained exclusively on the supervised loss function. It is imperative to handle negative samples with care, concentrating on enhancing the supervised loss without backpropagating gradients to prevent inadvertent debiasing.

Our methodology aims to guarantee fairness and alleviate biases in tasks like facial detection by autonomously learning latent variables in an adaptive manner. The experiments showcase the efficacy of the proposed DBVAE architecture in achieving these objectives while retaining the original set of equations.

Algorithm for Automated Debiasing

In this section, we outline the procedure for dynamically adjusting the training data through adaptive resampling, leveraging the latent structure acquired by our DB-VAE model. By excluding frequently occurring segments of the latent space, we enhance the likelihood of selecting less common data for training. This adaptation occurs concurrently with the learning of latent variables, ensuring that our debiasing strategy comprehensively considers the entire distribution of underlying features in the training dataset.

The training dataset undergoes encoding through the network, yielding an estimate $Q(z|X)$ of the latent distribution. The objective is to amplify the presence of infrequent data points by intensifying the sampling of underrepresented areas within the latent space. To achieve this, we approximate the latent space distribution using a histogram $Q^\dagger(z|X)$ with a dimensionality determined by the number of latent variables k. To mitigate the challenges posed by high-dimensionality as the latent space complexity increases, we

further simplify the approach by employing independent histograms to approximate the joint distribution. Specifically, we define an independent histogram, $Q_i^\dagger(zi|X)$, for each latent variable zi:

$$Q^\dagger(z|X) \propto \prod_i Q_i^\dagger(zi|X) \quad (3)$$

This enables a concise approximation of $Q(z|X)$ based on the frequency distribution of each learned latent variable. Introducing a single parameter, α, allows us to modulate the degree of debiasing during training. We define the probability distribution for selecting a datapoint x as $W(z(x)|X)$, parametrized by the debiasing factor α:

$$W(z(x)|X) \propto \prod_i \Big(1 + \alpha Q_i^\dagger(zi(x)|X)\Big) \quad (4)$$

Algorithm 1 provides pseudocode for training the DB-VAE. In each epoch, all inputs x from the original dataset X are propagated through the model to assess the corresponding latent variables $z(x)$. Subsequently, the histograms $Q_i^\dagger(zi(x)|X)$ are updated accordingly. During training, a new batch is drawn by selectively retaining inputs x from the original dataset X based on the likelihood $W(z(x)|X)$. Training on the debiased data batch compels the classifier to optimize parameters for improved performance in rare cases, without significant degradation in performance for common training examples. Importantly, the debiasing process is not predetermined manually; instead, it is informed by the learned latent variables.

Algorithm 1: Adaptive Re-sampling for Automated Debiasing of the DB-VAE Architecture

Require: Training data X, Y, batch size b
1: Initialize weights ϕ, θ
2: for each epoch, E_t do
3: Sample $z \sim q_\phi(z|X)$
4: Update $Q_i^\dagger(zi(x)|X)$
5: $W(z(x)|X) \leftarrow \prod_i \Big(1 + \alpha Q_i^\dagger(zi(x)|X)\Big)$
6: while $iter < n_b$ do
7: Sample $x_{\text{batch}} \sim W(z(x)|X)$
8: $L(\phi, \theta) \leftarrow \frac{1}{b} \sum_{i \in x_{\text{batch}}} L_i(\phi, \theta)$
9: Update: $[w \leftarrow w - \eta \nabla_{\phi,\theta} L(\phi, \theta)]_{w \in \{\phi,\theta\}}$
10: end while
11: end for

3. Experiment

Dataset

Our classifier undergoes training on an extensive dataset comprising $n = 4 \times 10^5$ images. This dataset is balanced, consisting of 2×10^5 positive examples (images of faces) and 2×10^5 negative examples (images of non-faces). The dataset is partitioned into training and validation sets, with an 80-20 split, respectively. Positive examples are sourced from the CelebA dataset (Liu et al. 2015), cropped to a square based on annotated face bounding boxes. Negative examples are extracted from the ImageNet dataset (Deng et al. 2009), encompassing a diverse array of non-human categories. All images are uniformly resized to 64×64.

Post-training, our debiasing algorithm undergoes evaluation on the PPB test dataset (Buolamwini and Gebru 2018). This dataset features images of 1270 male and female parliamentarians from diverse African and European countries. The dataset exhibits uniformity in pose, illumination, and facial expression, with balanced representation in both skin tone and gender. Gender annotations include the labels "Male" and "Female," while skin tone annotations adhere to the Fitzpatrick skin type classification system (Fitzpatrick 1988), categorizing each image as "Lighter" or "Darker."

Training the Models

For the traditional facial detection task, we employ a convolutional neural network (CNN) featuring four consecutive convolutional layers with 5x5 filters and 2x2 strides for efficient feature extraction. The final classification stage involves two fully connected layers, each with 1000 and 1 hidden neurons, respectively. ReLU activation and batch normalization (Ioffe and Szegedy 2015) are applied across all layers in the network. Our DB-VAE architecture shares the same classification network for the encoder, with the exception of the final fully connected layer, which now yields an additional k latent variables, resulting in a total of $2k + 1$ activations. A decoder, mirroring the encoder with 2 fully connected layers and 4 de-convolutional layers, is then employed to reconstruct the original input image. Training is conducted by minimizing the empirical training loss, as defined in Eq. 2, with L2 reconstruction loss.

In our experiments, we strategically block all gradients from the decoder network when $y = 0$ (i.e., for negative examples), focusing debiasing efforts exclusively on positive face examples. Alongside training the standard classification network with no debiasing, we train DB-VAE models with varying degrees of debiasing, controlled by the parameter α, for 50 epochs. The models are subsequently evaluated on the validation set. To ensure the robustness of results, models are re-trained from scratch five times, adding a layer of statistical reliability to the findings.

Automated debiasing of Facial Detection Systems

We delve into the outcomes of the debiasing algorithm and conduct a comprehensive assessment of our trained models on the PPB dataset. Our focus extends to the resampling probabilities, $W(z(x)|X)$, derived from the debiased model. And with an increase in the probability of resampling, there is a corresponding decrease in the number of data points within the associated bin. This trend suggests that images more likely to be resampled are those characterized by 'rare' features.

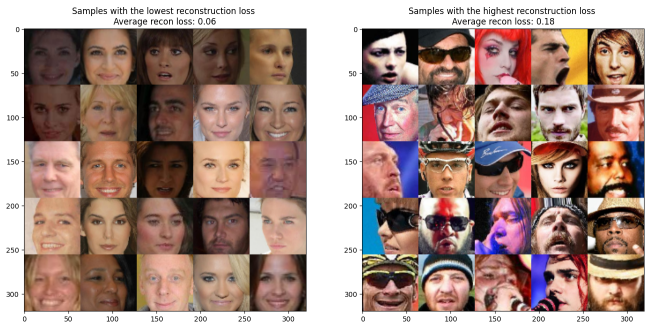

Figure 1. :. The top faces with the lowest (left) and highest (right) probabilities of being sampled.

As the probability of resampling rises, the images become more diverse. Further validation comes from examining the few faces in the training data with the lowest and highest resampling probabilities (Fig.1). Faces with the lowest resampling probability exhibit a high degree of uniformity in skin tone, hair color, forward gaze, and background color. In contrast, faces with the highest resampling probability showcase rarer features such as headwear or eyewear, tilted gaze, shadowing, and darker skin. Collectively, these findings indicate that our algorithm effectively identifies and actively resamples data points with rarer, more distinctive features.

4. Conclusion

In this paper, we present a groundbreaking debiasing algorithm designed to dynamically adjust the sampling probabilities of individual data points during training. The innovation lies in its unsupervised learning approach, allowing for scalability to large datasets without the need for manual labeling of latent variables. Applied to the realm of facial detection, our method not only enhances classification accuracy but also mitigates categorical biases, particularly across race and gender, when compared to conventional classifiers. We provide a comprehensive algorithm for debiasing and share an open-source implementation of our model. Our work contributes to the ongoing effort to develop and deploy fair and unbiased AI systems, playing a crucial role in preventing unintended discrimination and fostering long-term acceptance of these algorithms in various domains.

Acknowledgments

This work is mostly under the study of the research paper published ”Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure” and the course Advanced Machine Learning by Prof. Amit Sethi. The author thank Prof. Alexander Amini (MIT), Prof. Amit Sethi (IIT Bombay), Joy Buolamwini, IBM Research.

References

[1] Amini, Alexander et al. ”Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure.” Proceedings of the 2019 AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES), 27-28 January, 2019, Honolulu, Hawaii, United States, AAAI/ACM, 2019.

[2] Aleksandra Mojsilovic et al. ”Understanding Unequal Gender Classification Accuracy from Face Images”, arXiv: 1812.00099.

[3] Alexandra Chouldechova, ”Fair prediction with disparate impact: A study of bias in recidivism prediction instruments”,2018

[4] [Abdullah et al. 2017] Abdullah, N. A.; Saidi, M. J.; Rahman, N. H. A.; Wen, C. C.; and Hamid, I. R. A. 2017. Face recognition for criminal identification: An implementation of principal component analysis for face recognition. In AIP Conference Proceedings, volume 1891, 020002.

[5] Adams, R. P. and Ghahramani, Z. (2009). Archipelago: nonparametric Bayesian semi-supervised learning. In Proceedings of the International Conference on Machine Learning (ICML).

[6] Diederik P. Kingma, Danilo J. Rezende†, Shakir Mohamed, Max Welling: Semi-supervised Learning with Deep Generative Models

[7]