Supplemental Materials: Learning Bias-Invariant Representation by Cross-Sample Mutual Information Minimization

Wei Zhu[1], Haitian Zheng[1], Haofu Liao [*2], Weijian Li[1], and Jiebo Luo[1]

[1]University of Rochester
[2]Amazon Web Services
{*wzhu15,hzheng15,wli69*}*@ur.rochester.edu, liaohaofu@gmail.com, jluo@cs.rochester.edu*

1. Discussion on I_{CS} Eq. (2)

The proposed I_{CS} is a lower bound for I_{JSD} with same definition of joint distribution and product of marginal distribution. Specifically, following the definition used in I_{CS}, I_{JSD} could be rewritten as

$$I_{JSD}(h^y, h^b) = \sup -E_{(i,j)\in\Omega}\log(1+\exp(-M(h_i^y, h_j^b))) - E_{(i,j)\notin\Omega}\log(1+\exp(M(h_i^y, h_j^b)))$$

Lemma 1. *I_{CS} is a lower bound for I_{JSD}.*

Proof. Since $-\log(1+x)$ is convex, based on Jensen's inequality, we know

$$-E_{(i,j)\in\Omega}\log(1+g_{(i,j)}) \geq -\log(1+E_{(i,j)\in\Omega}g_{(i,j)}).$$

By replacing $g_{(i,j)}$ with $\exp(-M(h_i^y, h_j^b))$ and $\exp(M(h_i^y, h_j^b))$ respectively, and substituting them to I_{CS} and I_{JSD}, we directly get $I_{CS} \leq I_{JSD}$. We thus complete our proof. □

2. Algorithm for CSAD

CSAD requires pretraining for target classifier, bias classifier, and mutual information estimator. We provide the complete algorithm for CSAD in Algorithm 1.

3. Algorithm for Optimizing Eq. (2)

We provide the details on optimizing Eq. (2) with joint content and local structural learning in Algorithm 2.

[*]This work was done when Haofu Liao was at the University of Rochester.

Algorithm 1 Learning Bias-Invariant Representation

Input: Training data $x = \{(x_i, y_i, b_i)\}$;
 # STEP 1: Pretrain Feature Extractor and Target Branch
Pretrain F, D^y, and P^y by minimizing the target prediction loss until convergence;
 # STEP 2: Pretrain Bias Branch
Pretrain D^b and P^b by minimizing the bias prediction loss until convergence;
 # STEP 3: Pretrain Mutual Information Estimator
Pretrain M to maximize Eq. (2) until convergence;
 # STEP 4: Iteratively Update
while not converge **do**
  Sample a minibatch of data;
  Update F, D^y, and P^y to minimize the target predication loss;
  for k = 1, . . . , K **do**
    Update D^b and P^b to minimize the bias predication loss;
  end for
  for k = 1, . . . , K **do**
    Update M to maximize Eq. (2);
  end for
  # Adversarial Debiasing
  Update F to minimize Eq. (2);
end while

4. Dataset and Network Structure

We provide detail structure of used network for different datasets. We omit the activation function (ReLU Layer) of the network for convenience.

Algorithm 2 Cross sample Mutual Information Estimator M

Input: Target representation $h^y = \{h_i^y\}$, Bias representation $h^b = \{h_i^b\}$;

STEP I: Content Similarity Learning
Calculate content similarity as Eq. (3)
STEP II: Structural Similarity Learning
Calculate the pairwise similarity matrices for h^y and h^b by Eq. (4), and then normalize the matrices by Eq. (5) to obtain the edge E^y and E^b.
Conduct RWR on the obtained graph G^y and G^b respectively to obtain r_i^y and r_i^b for the i-th sample by Eq. (7);
Normalize the r_i^b and r_i^y for the i-th sample;
Calculate the structural similarity $s_s^y(i,j)$ by Eq. (8);
STEP III: Joint Similarity Learning
Obtain the joint similarity by Eq. (9);
STEP IV: Cross Sample Mutual Information Estimation
Update M to maximize Eq. (2).

4.1. Colored Mnist

The Colored MNIST dataset [4] introduces color bias to the standard MNIST dataset [6], and the digits are class-wisely colored for the training set following [4]. We assign a mean color for each class of digit. Then, for each training image, its color is sampled from a normal distribution with the mean set as the class-wise mean color and a predefined variance σ^2. We vary the variance σ^2 from 0.02 to 0.05 to have a different amount of bias in the training data, and smaller σ^2 represents more color bias. To have a bias-free testing set, testing images are generated similarly to the training ones but with the mean randomly sampled from ten mean colors. The color label is grouped into eight different categories for each RGB channel following [4].

The used network for target and bias task follows [9]. For detail we adopt two convolutional layers with kernel size as 5 and 64 filters as feature extractor. The disentangler is implemented with a fully connected layer (1024-128). Class Predictor is implemented with two fully connected layer as (128-64-10). Bias Predictor is also implemented with two fully connected layer for each color as (128-64-8). The mutual information estimator is a three-layer fully connected network as (128-64-32-32).

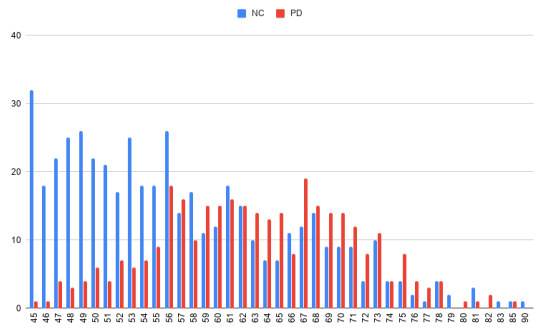

Figure 1. The age distribution of PD and non-PD groups for the mPower dataset. The dataset is biased on age, as PD patients are typically elder.

4.2. IMDB face

The IMDB face dataset [10] is a face image dataset that contains $460,723$ face images of $20,284$ celebrities with their information regarding age and gender. Following [9], a pretrained network on Image [3] for age and gender annotation is used to filter out the misannotated images, resulting a cleaned dataset of 112,340 samples. We aim to conduct gender prediction on an age-biased training set. Likewise, the cleaned images are divided into three subsets, namely: Extreme bias 1 (EB1): women aged 0-29, men aged 40+; Extreme bias 2 (EB2): women aged 40+, men aged 0-29; Test set: 20% of the cleaned images aged 0-29 or 40+. As a result, EB1 and EB2 are biased towards the age, since EB1 consists of younger females and older males and EB2 consists of younger males and older females.

Follow [4], feature extractor is implemented with a pretrained ResNet-18 by modifying the last fc layer as (512-256). The disentangler is a fully connected layer as (256-64). The class predictor is a fully connected layer as (64-1) and the bias predictor is also a fully connected layer as (64-12). The mutual information estimator is a three-layer fully connected network as (64-64-32-32).

4.3. CelebA

Follow [8], feature extractor is implemented with a pretrained ResNet-18 by modifying the last fc layer as (512-256). The disentangler is a fully connected layer as (256-64). The class predictor is a fully connected layer as (64-1) and the bias predictor is also a fully connected layer as (64-1). The mutual information estimator is a three-layer fully connected network as (64-64-32-32). We train the model with a balanced batch to alleviate the unbalance problem.

4.4. mPower

We illustrate the age bias for mPower dataset in Fig. 1. We can observe that most PD's patients are the elder, and it is thus necessary to remove the age bias for PD's diagnosis. We conduct adversarial debiasing for the finger tapping task, where patients will tap their phones alternatively with two fingers. To evaluate the debiasing methods, we contrive a bias-free testing set following the settings of Colored MNIST and IMDB face. For detail, we divide the age into 6 different intervals $\{45-49, 50-54, 55-59, 60-67, 65-69, 70+\}$, and then draw 30 PD and NC subjects from each interval as the testing set (360 samples in total). The training set contains all the other 1044 patients.

Feature extractor is implemented with a 6-layer TCN with kernel size as 5 and 64 filters. The disentangler is a fully connected layer as (64-64). The class predictor is a fully connected layer as (64-1) and the bias predictor is also a fully connected layer as (64-8). The mutual information estimator is a three-layer fully connected network as (64-64-32-32).

4.5. Adult

To comprehensively evaluate the performance, various metrics have been applied following [11]. First, Balanced accuracy (BA) is used for imbalance data. Moreover, we evaluate the model with counterfactual samples by flipping the attributes of spouse (gender&Race) for testing records, and calculate spouse (gender&Race) consistency S-Con (GR-Con) by the predication consistency between original and altered samples [11]. We also report group fair metrics provided by AIF360 [1] with respect to race or gender, including $\text{Gap}_G^{\text{RMS}}$, $\text{Gap}_R^{\text{RMS}}$, $\text{Gap}_G^{\text{max}}$, and $\text{Gap}_R^{\text{max}}$, and please refer to [11] for their definitions.

The baseline used by other methods is a two-layer MLP as (41-100-2) [11]. We adopt a three-layer MLP as (41-64-32-2). We note that the three-layer MLP has fewer parameters than the two-layer baseline and achieves competitive performance. For the proposed methods, we discompose the three-layer baseline into three modules. Feature extractor is a fully connected layer as (41-64). The disentangler is a fully connected layer as (64-32). The class predictor is a fully connected layer as (32-1) and the bias predictor is also a fully connected layer as (32-2). The mutual information estimator is a two-layer fully connected network as (32-32-32). We construct a balanced minibatch for Adult which contains same number of samples from each target class following [11].

5. Implementation Details

We use Adam to train our model [5]. We set $K = 10$, $\tau = 10$ Eq.(5), $c = 0.5$ Eq. (6), and $\alpha = 1$ Eq. (9), and search $\lambda \in \{0.1, 0.5, 1, 10\}$. We would like to emphasize that λ is particularly important for our method to achieve a fairness-accuracy balance. Specifically, over large λ would force the feature extractor to learn little information, and small ones would lead to little influence on the pretrained target classifier. We conduct experiments on Adult to show the trade-off for our method in Table 1. The λ is default set to 10 for Adult. As we increase λ, our model focuses more on fairness with reduced accuracy. We note that the fairness-accuracy trade-off is still an open and significant problem for debiasing and fairness [7, 2]. We will study the problem comprehensively in the future.

Table 1. We vary the λ to study the accuracy-fairness trade-off of our method on Adult dataset. BA is balanced accuracy, while the other four metrics are used to evaluate the fairness.

λ	BA ↑	$\text{Gap}_G^{\text{RMS}}$ ↓	$\text{Gap}_R^{\text{RMS}}$ ↓	$\text{Gap}_G^{\text{max}}$ ↓	$\text{Gap}_R^{\text{max}}$ ↓
1	**81.4**	.118	.061	.130	.067
2	80.7	.080	.053	.109	.055
10	80.4	.060	.042	.066	.058
20	78.9	**.058**	.035	**.065**	.050
40	78.5	.063	**.030**	.088	**.042**

References

[1] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, Oct. 2018. 3

[2] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and KR Varshney. Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing. In *Proceedings of the 37 th International Conference on Machine Learning. Vienna. https://proceedings. icml. cc/static/paper_files/icml/2020/2831-Paper. pdf (PMLR 119, 2020)*, 2020. 3

[3] Tal Hassner et al. Age and gender classification using convolutional neural networks. 2

[4] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9012–9020, 2019. 2

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[6] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 2

[7] Suyun Liu and Luis Nunes Vicente. Accuracy and fairness trade-offs in machine learning: A stochastic multi-objective approach. *arXiv preprint arXiv:2008.01132*, 2020. 3

[8] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. *arXiv preprint arXiv:2007.02561*, 2020. 2

[9] Ruggero Ragonesi, Riccardo Volpi, Jacopo Cavazza, and Vittorio Murino. Learning unbiased representations via mutual information backpropagation. *CoRR*, abs/2003.06430, 2020. 2

[10] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 10–15, 2015. 2

[11] Mikhail Yurochkin, Amanda Bower, and Yuekai Sun. Training individually fair ml models with sensitive subspace robustness. In *International Conference on Learning Representations*, 2019. 3