Learning Unbiased Representations via Rényi Minimization

Vincent Grari[1,3]✉, Oualid El Hajouji[2], Sylvain Lamprier[1], and Marcin Detyniecki[3]

[1] Sorbonne Université LIP6/CNRS Paris, France
[2] Ecole polytechnique Palaiseau, France
[3] AXA REV Research Paris, France
{vincent.grari,sylvain.lamprier}@lip6.fr
oualid.el-hajouji@polytechnique.edu ; marcin.detyniecki@axa.com

Abstract. In recent years, significant work has been done to include fairness constraints in the training objective of machine learning algorithms. Differently from classical prediction retreatment algorithms, we focus on learning fair representations of the inputs. The challenge is to learn representations that capture most relevant information to predict the targeted output Y, while not containing any information about a sensitive attribute S. We leverage recent work which has been done to estimate the Hirschfeld-Gebelein-Renyi (HGR) maximal correlation coefficient by learning deep neural network transformations and use it as a min-max game to penalize the intrinsic bias in a multi dimensional latent representation. Compared to other dependence measures, the HGR coefficient captures more information about the non-linear dependencies, making the algorithm more efficient in mitigating bias. After providing a theoretical analysis of the consistency of the estimator and its desirable properties for bias mitigation, we empirically study its impact at various levels of neural architectures. We show that acting at intermediate levels of neural architectures provides best expressiveness/generalization abilities for bias mitigation, and that using an HGR based loss is more efficient than more classical adversarial approaches from the literature.

1 Introduction

This recent decade, deep learning models have shown very competitive results by learning representations that capture relevant information for the learning task. However, the representation learnt by the deep model may contain some bias from the training data. This bias can be intrinsic to the training data, and may therefore induce a generalisation problem due to a distribution shift between training and testing data. For instance, the color bias in the colored MNIST data set [25] can make models focus on the color of a digit rather than its shape for the classification task. The bias can also go beyond training data, so that inadequate representations can perpetuate or even reinforce some society biases [10]

[1] https://github.com/axa-rev-research/unbiased_representations_renyi.

(e.g. gender or age). Since the machine learning models have far-reaching consequences in our daily lives (credit rating, insurance pricing, recidivism score, etc.), we need to make sure that the representation data contains as little bias as possible. A naive method to mitigate bias could be to simply remove sensitive attributes from the training data set [36]. However, this concept, known as "fairness through unawareness", is highly insufficient because any other non-sensitive attribute might indirectly contain significant sensitive information reflected in the deep learning representation. For example, the height of an adult could provide a strong indication about the gender. A new research field has emerged to find solutions to this problem: fair machine learning. Its overall objective is to ensure that the prediction model is not dependent on a sensitive attribute [46]. Many recent papers tackle this challenge using an adversarial neural architecture, which can successfully mitigate the bias. We distinguish two adversarial mitigation families. While prediction retreatment methods apply mitigation on the output prediction [48], fair representation methods consider sensitive bias in intermediary latent representations [1]. Our claim is that mitigating at intermediate levels of neural architectures allows a greater stability at test time, which we observe in our experiments (section 6.3).

In this paper, we propose a new fair representation architecture by leveraging the recent Renyi neural estimator, previously used in a prediction retreatment algorithm [19] and we propose to study why such an architecture outperforms the state of the art. The contributions of this paper are:

- We provide a theoretical analysis of the consistency of the HGR estimator, along its nice properties compared to state-of-the-art for fair representation;
- We propose a neural network architecture which creates a fair representation by minimizing the HGR coefficient. The HGR network is trained to discover non-linear transformations between the multidimensional latent representation and the sensitive feature. Note that this is also the first use of a neural HGR estimator for multidimensional variables;
- We empirically demonstrate that our neural HGR-based approach is able to identify the optimal transformations with multidimensional features and present very competitive results for fairness learning;
- To the best of our knowledge, this is the first work to compare mitigation at different levels of neural architectures. We show that acting at intermediary levels of neural representations allows the best trade-off between expressiveness and generalisation for bias mitigation.

2 Related Work

Significant work has been done in the field of fair machine learning recently, in particular when it comes to quantifying and mitigating undesired bias. For the mitigation approaches, three distinct strategy groups exist. While pre-processing [23, 7, 11] and post-processing [21, 13] approaches respectively act on the input or the output of a classically trained predictor, in-processing approaches mitigate the undesired bias directly during the training phase [46, 12, 48, 30]. In this paper we focus on in-processing fairness, which proves to be the most powerful framework for settings where acting on the training process is an option.

Among the in-processing approaches, some of them, referred to as prediction retreatment, aim at directly modifying the prediction output by adversarial training. To ensure independence between the output and the sensitive attribute, Zhang et al. [48] feed the prediction output as input to an adversary network (upper right in Fig 1 in appendix), whose goal is to predict the sensitive attribute, and update the predictor weights to fool the adversary. Grari et al. [19] minimize the HGR correlation between the prediction output and the sensitive attribute in an adversarial learning setting (middle right in Figure 1 in appendix).

On the other hand, several research sub-fields in the in-processing family tackle the problem of learning unbiased representations. Domain adaptation [14, 9] and domain generalization [35, 28] consist in learning representations that are unbiased with respect to a source distribution, and can therefore generalize to other domains. Some of the works in these fields involve the use of adversarial methods [16, 17], close to our work. Several strategies mitigate bias towards a sensitive attribute through representation. One approach [47] relies on a discriminative clustering model to learn a multinomial representation that removes information regarding a binary sensitive attribute. A different approach [2] consists in learning an unbiased representation by minimizing a confusion loss. Invariant representations can also be learnt using Variational Auto-Encoders [26], by adding a mutual information penalty term [34]. One of the first proposition by adversarial neural network for fair representation has been proposed by [32] by mitigating the bias on the latent space with an adversarial and decoding of X from Z and A. Adel et al. [1] learn also a fair representation by inputting it to an adversary network, which is prevented from predicting the sensitive attribute (upper left in Figure 1 in appendix). Other papers minimize the mutual information between the representation and the sensitive attribute: Kim et al. [25] rely on adversarial training with a discriminator detecting the bias, while Ragonesi et al. [38] rely on an estimation by neural network of mutual information [6] (lower left in Figure 1 in appendix). A kernelized version of such adversarial debiasing approach for fair representation is provided in [40].

3 Problem Statement

Throughout this document, we consider a supervised algorithm for regression or classification problems. The training data consists of n examples $(x_i; s_i; y_i)_{i=1}^{n}$, where x_i 2 R^p is the feature vector with p predictors of the i-th example, s_i is its continuous sensitive attribute and y_i its continuous or discrete outcome. We address a common objective in fair machine learning, *Demographic Parity*, which ensures that the sensitive attribute S is independent of the prediction $\hat{Y}$.

3.1 Metrics for Continuous Statistical Dependence

In order to assess this fairness definition in the continuous case, it is essential to look at the concepts and measures of statistical dependence. Simple ways of measuring dependence are Pearson's rho, Kendall's tau or Spearman's rank. Those types of measure have already been used in fairness, with the example of mitigating the conditional covariance for categorical variables [46]. However, the major problem with these measures is that they only capture a limited class of association patterns, like linear or monotonically increasing functions. For example, a random variable with standard normal distribution and its cosine (non-linear) transformation are not correlated in the sense of Pearson.

Over the last few years, many non-linear dependence measures have been introduced like the Kernel Canonical Correlation Analysis (KCCA) [20], the Distance or Brownian Correlation (dCor) [41], the Hilbert-Schmidt Independence Criterion (HSIC and CHSIC) [37]or the Hirschfeld-Gebelein-Rényi (HGR) [39]. Comparing those non-linear dependence measures [29], the HGR coefficient seems to be an interesting choice: it is a normalized measure which is capable of correctly measuring linear and non-linear relationships, it can handle multi-dimensional random variables and it is invariant with respect to changes in marginal distributions.

Definition 1. *For two jointly distributed random variables U 2 U and V 2 V, the Hirschfeld-Gebelein-Rényi maximal correlation is defined as:*

$$HGR(U;V) = \sup_{f:U \; ! \; \mathrm{R}, g:V \; ! \; \mathrm{R}} (f(U);g(V)) = \sup_{\substack{f:U \; ! \; \mathrm{R}, g:V \; ! \; \mathrm{R} \\ E(f(U))=E(g(V))=0 \\ E(f^2(U))=E(g^2(V))=1}} E(f(U)g(V)) \tag{1}$$

where is the Pearson linear correlation coefficient with some measurable functions f and g with positive and finite variance.

The HGR coefficient is equal to 0 if the two random variables are independent. If they are strictly dependent the value is 1. The spaces for the functions f and g are infinite-dimensional. This property is the reason why the HGR coefficient proved difficult to compute.

Several approaches rely on Witsenhausen's linear algebra characterization [44] to compute the HGR coefficient. For discrete features, this characterization can be combined with Monte-Carlo estimation of probabilities [5], or with kernel density estimation (KDE) [33] to compute the HGR coefficient. We will refer to this second metric, in our experiments, as HGR_KDE. Note that this metric can be extended to the continuous case by discretizing the density computation. Another way to approximate this coefficient, Randomized Dependence Coefficient (RDC) [29], is to require that f and g belong to reproducing kernel Hilbert spaces (RKHS) and take the largest canonical correlation between two sets of copula random projections. We will make use of this approximated metric as HGR_RDC. Recently a new approach [19] proposes to estimate the HGR by deep neural network. The main idea is to use two inter-connected neural networks to approximate the optimal transformation functions f and g from 1. The $HGR_{\Theta}(U;V)$ estimator is computed by considering the expectation of the products of standardized outputs of both networks ($\hat{f}_{w_f}$ and $\hat{g}_{w_g}$). The respective parameters w_f and w_g are updated by gradient ascent on the objective function to maximize: $J(w_f, w_g) = E[\hat{f}_{w_f}(U)\hat{g}_{w_g}(V)]$. This estimation has the advantage of being estimated by backpropagation, the same authors therefore present a bias mitigation via a min-max game with an adversarial neural network architecture. However, this attenuation is performed on the predictor output only. Several recent papers [1, 38] have shown that performing the attenuation on a representation tends to give better results in terms of prediction accuracy while remaining fair in complex real-world scenarios. In this work, we are interested in learning fair representations via this Renyi estimator.

4 Theoretical Properties

In this section we study the consistency of the HGR_NN estimator (referred to as $\widehat{HGR}(U, V)_n$), and provide a theoretical comparison with simple adversarial algorithms that rely on an adversary which predicts the sensitive attribute [48, 1]. All the proofs can be found in the Supplementary Material.

4.1 Consistency of the HGR_NN

Definition 2. *(Strong consistency) The estimator $\widehat{HGR}(U, V)_n$ is strongly consistent if for all $\epsilon > 0$, there exists a positive integer N and a choice of statistics network such that:*

$$\forall n \geq N, |HGR(U, V) - \widehat{HGR}(U, V)_n| \leq \epsilon, a.s. \tag{2}$$

As explained in MINE [6], the question of consistency is divided into two problems: a deterministic approximation problem related to the choice of the statistics network, and an estimation problem related to the use of empirical measures.

The first lemma addresses the approximation problem using universal approximation theorems for neural networks [22]:

Lemma 1. *(approximation) Let $\epsilon > 0$. There exists a family of continuous neural networks F_Θ parametrized by a compact domain $\Theta \subset \mathbb{R}^k$, such that*

$$|HGR(U, V) - HGR_\Theta(U, V)| \leq \epsilon. \tag{3}$$

The second lemma addresses the estimation problem, making use of classical consistency theorems for extremum estimators [18]. It states the almost sure convergence of HGR_NN to the associated theoretical neural HGR measure as the number of samples goes to infinity:

Lemma 2. *(estimation) Let $\epsilon > 0$, and F_Θ a family of continuous neural networks parametrized by a compact domain $\Theta \subset \mathbb{R}^k$. There exists an $N \in \mathbb{N}$ such that:*

$$\forall n \geq N, |\widehat{HGR}(U, V)_n - HGR_\Theta(U, V)| \leq \epsilon, a.s. \tag{4}$$

It is implied here that, from rank N, all sample variances are positive in the definition of $\widehat{HGR}(U, V)_n$, which makes the latter well-defined.

We deduce from these two lemmas the following result:

Theorem 1. *$\widehat{HGR}(U, V)_n$ is strongly consistent.*

4.2 Theoretical comparison against simple adversarial algorithms

Given X and Y two one-dimensional random variables, we consider the regression problem:

$$\inf_{f:\mathbb{R}\to\mathbb{R}} E((Y - f(X))^2) \tag{5}$$

The variable that minimizes the quadratic risk is $E(Y|X)$. Thus, prediction retreatment algorithms with predictive adversaries [48], which consider such optimization problems for mitigating biases, achieve the global fairness optimum when $E(S|\hat{Y}) = E(S)$. This does **not generally imply demographic parity** when S is continuous. On the other hand, adversarial approaches based on the HGR_NN [19] achieve the optimum when $HGR(\hat{Y}, S) = 0$, which is equivalent to demographic parity: $P(\hat{Y}|S) = P(\hat{Y})$.

To illustrate this, we consider the maximization problem $\sup_{f:\mathbb{R}\to\mathbb{R}} \rho(f(X), Y)$, which corresponds to the situation where the neural network g is linear in the HGR neural estimator. We have the following result:

Theorem 2. *If $E(Y|X)$ is constant, then $\sup_f \rho(f(X), Y) = 0$. Else, $f^* \in \arg\max_f \rho(f(X), Y)$ iff there exists $a, b \in \mathbb{R}$, with $a > 0$, such that:*

$$f^*(X) = aE(Y|X) + b \tag{6}$$

In other words, the simpler version of the HGR_NN, with g linear, finds the optimal function in terms of regression risk, up to a linear transformation that can be found by simple linear regression. The simplified HGR estimation module therefore captures the exact same non-linear dependencies as the predictive adversary in related work [1, 48]. Thanks to the function g, in cases where Y cannot be expressed as a function of X only, the HGR neural network can capture more dependencies than a predictive NN (or equivalently a simplified HGR neural network).

Specific example to understand the difference: Let us consider the following example below where:

$$Y \sim N(\mu, \sigma^2) \qquad X = \arctan(Y^2) + U \tag{7}$$

where $U \perp Y$ and U follows a Bernoulli distribution with $p = \frac{1}{2}$. In this setting, we have $Y^2 = \tan(X)$, $HGR(X, Y) = 1$ and due to the hidden variable U, neither X nor Y can be expressed as a function of the other. In that case, the simplified maximal correlation, $\rho(E(Y|X);Y)$, has the following bounds, with $\alpha = \frac{\mu}{\sigma}$: $\sqrt{1 - e^{-\frac{\alpha^2}{2}}} \leq \rho(E(Y|X);Y) \leq \sqrt{1 - e^{-\frac{\alpha^2}{2}}(1+\alpha^2)^{-\frac{3}{2}}}$. In the degenerate case $\alpha = 0$, we have $E(Y|X) = 0$: the predictive neural network cannot find any dependence. For non-zero values of α, the distribution of Y is no longer centered around the axis of symmetry of the square function, so that the prediction becomes possible. However, as shown in the inequality above, the simplified maximal correlation is less than 1, and close to 0 when $\alpha \rightarrow 0$.

In Figure 1, we illustrate the bounds (proof in appendix), $\rho(E(Y|X);Y)$ being estimated by Monte-Carlo. First, we note that the upper bound is close to $\rho(E(Y|X);Y)$, whereas the lower bound $\sqrt{1 - e^{-\frac{\alpha^2}{2}}}$ is not as precise. For non-zero values of α, $\rho(E(Y|X);Y)$ is positive, so that a predictive neural network can capture some non-linear dependencies between Y and X.

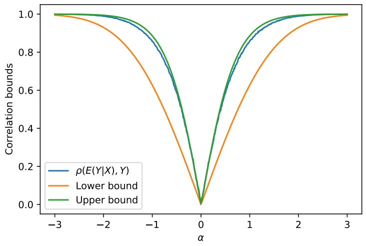


Fig. 1: Simplified HGR w.r.t α

This is due to the fact that, for $\alpha \neq 0$, the square function is bijective when restricted to some open interval containing the mean of Y, whereas when $\alpha = 0$, such an interval cannot be found. When this interval is large and the standard deviation of Y is not too large (which corresponds to high values of $|\alpha|$), $\rho(E(Y|X);Y)$ approaches 1 and the Y prediction error approaches 0. In the opposite case, $\rho(E(Y|X);Y)$ is close to 0 and a predictive neural network cannot capture dependencies.

Therefore, as shown by the example, the bilateral approach of the HGR, as opposed to the unilateral approach of predictive models, can capture more dependencies in complex regression scenarios. In adversarial bias mitigation settings, predictive adversaries might not be able to properly detect bias. Adversarial approaches based on the HGR_NN are better fitted for bias mitigation in such continuous complex settings.

5 Method

The objective is to find a latent representation Z which both minimizes the deviation between the target Y and the output prediction $\hat{Y}$, provided by a function (Z), and does not imply too much dependence with the sensitive S. As explained above in section 3, the HGR estimation by deep neural network [19] is a good candidate for standing as the adversary $HGR(Z;S)$ to plug in the global objective (8). Notice, we can consider the latent representation Z or even the sensitive attribute S as multi-dimensional. This can therefore provide a rich representation of the latent space or even take into account several sensitive features at the same time (for e.g. gender and age or the 3 channels of an image see 6.1). The HGR estimation paper [19] considers only the one-dimensional cases for both U and V but we can generalize to the multidimensional cases.

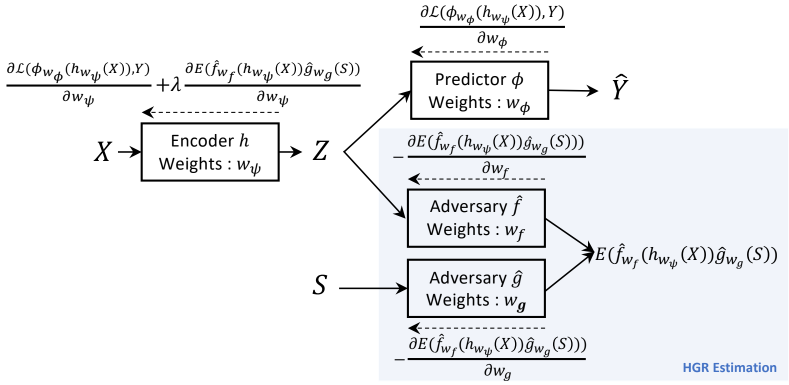

$\frac{\partial \mathcal{L}(\phi_{w_\phi}(h_{w_\psi}(X)),Y)}{\partial w_\phi}$
Predictor ϕ
Weights : w_ϕ
$\hat{Y}$
$\frac{\partial \mathcal{L}(\phi_{w_\phi}(h_{w_\psi}(X)),Y)}{\partial w_\psi} + \lambda \frac{\partial E(\hat{f}_{w_f}(h_{w_\psi}(X))\hat{g}_{w_g}(S))}{\partial w_\psi}$
X
Encoder h
Weights : w_ψ
Z
$-\frac{\partial E(\hat{f}_{w_f}(h_{w_\psi}(X))\hat{g}_{w_g}(S))}{\partial w_f}$
Adversary $\hat{f}$
Weights : w_f
$E(\hat{f}_{w_f}(h_{w_\psi}(X))\hat{g}_{w_g}(S))$
S
Adversary $\hat{g}$
Weights : w_g
$-\frac{\partial E(\hat{f}_{w_f}(h_{w_\psi}(X))\hat{g}_{w_g}(S))}{\partial w_g}$
HGR Estimation

Fig. 2: Learning Unbiased Representations via Rényi Minimization

The mitigation procedure follows the optimization problem:

$$\underset{w\ \ ,w}{\arg\min}\ \underset{w_f,w_g}{\max}\ L(\ _\omega\ (h_\omega\ \ (X));Y) +\ \ E(\hat{f}_{w_f}(h_\omega\ \ (X))\ \ \hat{g}_{w_g}(S)) \qquad (8)$$

where L is the predictor loss function between the output prediction $_\omega\ (h_\omega\ \ (X))\ 2$ R and the corresponding target Y, with $_\omega$ the predictor neural network with parameters $!_\phi$ and $Z = h_\omega\ \ (X)$ the latent fair representation with h_ω the encoder neural network, with parameters $!_\psi$. The second term, which corresponds to the expectation of the products of standardized outputs of both networks ($\hat{f}_{w_f}$ and $\hat{g}_{w_g}$), represents the HGR estimation between the latent variable Z and the sensitive attribute S. The hyperparameter controls the impact of the correlation loss in the optimization.

Figure 2 gives the full architecture of our adversarial learning algorithm using the neural HGR estimator between the latent variable and the sensitive attribute. It depicts the encoder function h_w , which outputs a latent variable Z from X, the two neural networks f_{w_f} and g_{w_g}, which seek at defining the most strongly correlated transformations of Z and S and the neural network $_\omega$ which outputs the prediction $\hat{y}$ from the latent variable Z. Left arrows represent gradients backpropagation. The learning is done via stochastic gradient, alternating steps of adversarial maximization and global loss minimization. The algorithm (more details in the supplementary) takes as input a training set from which it samples batches of size b at each iteration. At each iteration it first standardizes the output scores of networks f_{w_f} and g_{w_g} to ensure 0 mean and a variance of 1 on the batch. Then it computes the HGR neural estimate and the prediction loss for the batch. At the end of each iteration, the algorithm updates the parameters of the prediction parameters $!_\phi$ as well as the encoder parameters $!_\psi$ by one step of gradient descent. Concerning the HGR adversary, the backpropagation of the parameters w_f and w_g is carried by multiple steps of gradient ascent. This allows us to optimize a more accurate estimation of the HGR at each step, leading to a greatly more stable learning process.

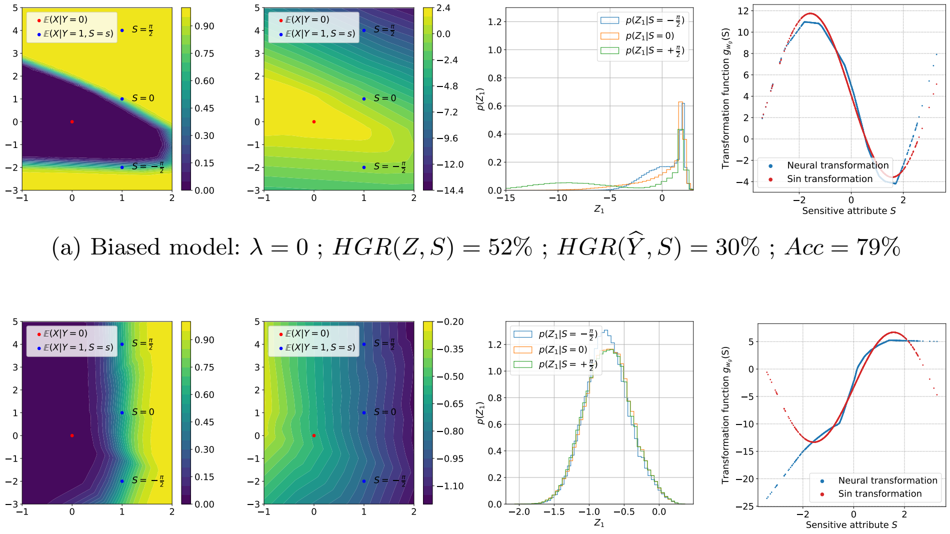
6 Experiments

6.1 Synthetic Scenario

We consider the following toy scenario in a binary target Y and continuous standard gaussian sensitive attribute S setting:

$$X j S = s \quad N \quad \begin{matrix}0\\0\end{matrix} ; \quad \begin{matrix}1 & \frac{1}{2}\\ \frac{1}{2} & 1\end{matrix} \qquad \text{when } Y = 0; \tag{9a}$$

$$X j S = s \quad N \quad \begin{matrix}1\\1 + 3\sin s\end{matrix} ; \quad \begin{matrix}1\,0\\0\,1\end{matrix} \qquad \text{when } Y = 1 \tag{9b}$$



(a) Biased model: $\lambda = 0$; $HGR(Z,S) = 52\%$; $HGR(\widehat{Y},S) = 30\%$; $Acc = 79\%$

(b) Unbiased model: $\lambda = 13$; $HGR(Z,S) = 5\%$; $HGR(\widehat{Y},S) = 4\%$; $Acc = 68\%$

Fig. 3: Toy example. (Left) Decision surface in the (X_1, X_2) plane. The figure (a) shows the decision surface for a biased model focused on a prediction loss. $\widehat{Y}$ values are highly correlated with S, samples with S around $\frac{\pi}{2}$ and $Y = 1$ being easier to classify than those with S between $\frac{\pi}{2}$ and 0. The figure (b) shows decision surfaces for our fair model. These are vertical, meaning that only X_1 influences the classification, and therefore $\widehat{Y}$ is no longer biased w.r.t S. (Middle left) Z_1-slices in the (X_1, X_2) plane. The comparison between the figure below and above highlights the fact that adversarial training allows to create an unbiased representation Z. (Middle right) Conditional probability densities of Z_1 at $S = \frac{\pi}{2}; 0; \frac{\pi}{2}$. With $= 0$, the densities are dependent on S, whereas they are not anymore with adversarial training. (Right) In blue, the function modeled by the neural network g in the HGR Neural Network. In red, the closest linear transformation of $\sin(S)$ to $g(S)$.

Our goal is to learn a representation Z of the input data that is no longer biased w.r.t S, while still accurately predicting the target value Y. Figure 3 compares the results of both a biased model (a) with a hyperparameter $= 0$ and an unbiased model (b) with $= 13$ applied on the toy scenario data. In the context of the Rényi Minimization method, it is interesting to observe the maximal correlation functions learnt by the adversary. When $= 0$, the adversary with sensitive attribute input models the $\sin$ function up to a linear transformation, which also maximizes the correlation with the input data as shown in (9b). In that case, the representation Z still carries the bias of X w.r.t S, in the same $\sin$ shape. When $= 13$, the neural network g is unable to find the $\sin$ function, which seems to indicate that the representation Z does not carry the bias w.r.t S anymore. This is confirmed by the low HGR coefficient between Z and S, the Z_1-slices as well as the conditional densities of Z_1 at different values of S. Not only does the adversarial induces an unbiased representation, it also leads to an almost completely unbiased target $\hat{Y}$, as shown by the vertical decision surfaces and the 4% HGR between $\hat{Y}$ and S. This at the cost of of a slight loss of accuracy, with an 11% decrease.

6.2 MNIST with Continuous Color Intensity

Before considering real-world experiments, we follow the MNIST experimental setup defined by Kim et al. [25], which considers a digit classification task with a color bias planted into the MNIST data set [27, 24]. In the training set, ten distinct colors are assigned to each class. More precisely, for a given training image, a color is sampled from the isotropic normal distribution with the corresponding class mean color, and a variance parameter ^{2}. For a given test image, a mean color is randomly chosen from one of the ten mean colors, without considering the test label, and a color is sampled from the corresponding normal distribution (with variance ^{2}). Seven transformations of the data set are designed with this protocol, with seven values of ^{2} equally spaced between 0.02 and 0.05. A lower value of ^{2} implies a higher color bias in the training set, making the classification task on the testing set more difficult, since the model can base its predictions on colors rather than shape. The sensitive feature, color, is encoded as a vector with 3 continuous coordinates. For each algorithm and for each data set, we obtain the best hyperparameters by grid search in five-fold cross validation.

	Color variance													
Training	$\sigma = 0.020$		$\sigma= 0.025$		$\sigma = 0.030$		$\sigma = 0.035$		$\sigma = 0.040$		$\sigma = 0.045$		$\sigma = 0.050$	
ERM ($\lambda = 0.0$)	0.476	0.005	0.542	0.004	0.664	0.001	0.720	0.010	0.785	0.003	0.838	0.002	0.870	0.001
Ragonesi et al. [38]	0.592	0.018	0.678	0.015	0.737	0.028	0.795	0.012	0.814	0.019	0.837	0.004	0.877	0.010
Zhang et al. [48]	0.584	0.034	0.625	0.033	0.709	0.027	0.733	0.020	0.807	0.013	0.803	0.027	0.831	0.027
Kim et al. [25]	0.645	0.015	0.720	0.014	0.787	0.018	0.827	0.012	0.869	0.023	0.882	0.019	0.900	0.012
Grari et al. [19]	0.571	0.014	0.655	0.022	0.721	0.030	0.779	0.011	0.823	0.013	0.833	0.026	0.879	0.010
Ours	**0.730**	0.008	**0.762**	0.021	**0.808**	0.011	**0.838**	0.010	**0.878**	0.011	**0.883**	0.012	**0.910**	0.007

Table 1: MNIST with continuous color intensity

Results, in terms of accuracy, can be found in Table 1. Notice, the state-of-the-art obtains different results than reported ones because we consider a continuous sensitive feature and not a 24-bit binary encoding. Our adversarial algorithm achieves the best accuracy on the test set for the seven scenarios. The most important gap is for the smallest sigma where the generalisation is the most difficult. The larger number of degrees of freedom carried by the two functions f and g made it possible to capture more unbiased information than the other algorithms on the multidimensional variables Z and S.

6.3 Real-world Experiments

Our experiments on real-world data are performed on five data sets. In three data sets, the sensitive and the outcome true value are both continuous: the US Census data set [43], the Motor data set [42] and the Crime data set [15]. On two other data sets, the target is binary and the sensitive features are continuous: The COMPAS data set [3] and the Default data set [45]. For all data sets, we repeat five experiments by randomly sampling two subsets, 80% for the training set and 20% for the test set. Finally, we report the average of the mean squared error (MSE), the accuracy (ACC) and the mean of the fairness metrics HGR_NN [19], HGR_KDE [33], HGR_RDC [29] and MINE [6] on the test set. Since none of these fairness measures are fully reliable (they are only estimations which are used by the compared models), we also use the *FairQuant* metric [19], based on the quantization of the test samples in 50 quantiles w.r.t. to the sensitive attribute. The metric corresponds to the mean absolute difference between the global average prediction and the mean prediction of each quantile.

As a baseline, we use a classic, "unfair" deep neural network, Standard NN. We compare our approach with state-of-the-art algorithms. We also compare the Fair MINE NN[19] algorithm where fairness is achieved with the MINE estimation of the mutual information as a penalization in prediction retreatment (lower right in Figure 1 in appendix).

For all the different fair representation algorithms, we assign the latent space with only one hidden layer with 64 units. Mean normalization was applied to all the outcome true values. Results of our experiments can be found in Table 2. For all of them, we attempted to obtain comparable results by giving similar accuracy to all models, via the hyperparameter (different for each model). For each algorithm and for each data set, we obtain the best hyperparameters by grid search in five-fold cross validation (specific to each of them). Notice that, as explained in Section 5, several optimization iterations are performed for the adversarial HGR neural estimation at each global backpropagation iteration (e.g., 50 iterations of HGR estimation at each step for the Compas dataset). For comparable results, we also optimize multiple iterations on the different adversarial state-of-the-art algorithms, we find the best number of adversarial backpropagation iterations by grid search between 1 to 300 by step of 25.

As expected, the baseline, Standard NN, is the best predictor but also the most biased one. It achieves the lowest prediction errors and ranks amongst the highest and thus worst values for all fairness measures for all data sets and

		MSE		HGR_NN		HGR_KDE		HGR_RDC		MINE		FairQuant	
US Census	Standard NN	0.274	0.003	0.212	0.094	0.181	0.00	0.217	0.004	0.023	0.018	0.059	0.00
	Grari et al. [19]	0.526	0.042	0.057	0.011	0.046	0.030	0.042	0.038	**0.001**	0.001	0.008	0.015
	Mary et al. [33]	0.541	0.015	0.075	0.013	0.061	0.006	0.078	0.013	0.002	0.001	0.019	0.004
	Fair MINE NN	0.537	0.046	0.058	0.042	0.048	0.029	0.045	0.037	**0.001**	0.001	0.012	0.016
	Adel et al. [1]	0.552	0.032	0.100	0.028	0.138	0.042	0.146	0.031	0.003	0.003	0.035	0.011
	Zhang et al. [48]	0.727	0.264	0.097	0.038	0.135	0.036	0.165	0.028	0.009	0.005	0.022	0.019
	Madras et al. [31]	0.525	0.033	0.129	0.010	0.158	0.009	0.173	0.012	0.007	0.007	0.041	0.003
	Sadeghi et al. [40]	0.526	0.006	0.077	0.031	0.136	0.001	0.146	0.001	0.008	0.003	0.035	0.000
	Ours	**0.523**	0.035	**0.054**	0.015	**0.044**	0.032	**0.041**	0.031	**0.001**	0.001	**0.007**	0.002
Motor	Standard NN	0.945	0.011	0.201	0.094	0.175	0.0	0.200	0.034	0.188	0.005	0.008	0.011
	Grari et al. [19]	0.971	0.004	0.072	0.029	0.058	0.052	**0.066**	0.009	**0.000**	0.000	0.006	0.02
	Mary et al. [33]	0.979	0.119	0.077	0.023	0.059	0.014	0.067	0.028	0.001	0.001	0.006	0.002
	Fair MINE NN	0.982	0.003	0.078	0.013	0.068	0.004	0.069	0.009	**0.000**	0.000	**0.004**	0.001
	Adel et al. [1]	0.979	0.003	0.101	0.04	0.09	0.03	0.101	0.04	0.002	0.002	0.009	0.004
	Zhang et al. [48]	0.998	0.004	0.076	0.034	0.091	0.024	0.129	0.08	0.001	0.001	**0.004**	0.001
	Madras et al. [31]	0.978	0.004	0.096	0.035	0.083	0.020	0.099	0.030	0.004	0.002	0.008	0.001
	Sadeghi et al. [40]	0.975	0.017	0.102	0.020	0.115	0.027	0.129	0.039	0.001	0.001	0.001	0.001
	Ours	**0.962**	0.002	**0.070**	0.011	**0.055**	0.005	0.067	0.006	**0.000**	0.000	**0.004**	0.001
Crime	Standard NN	0.384	0.012	0.732	0.013	0.525	0.013	0.731	0.009	0.315	0.021	0.353	0.006
	Grari et al. [19]	0.781	0.016	0.356	0.063	0.097	0.022	**0.171**	0.03	**0.009**	0.008	**0.039**	0.008
	Mary et al. [33]	0.778	0.103	0.371	0.116	0.115	0.046	0.177	0.054	0.024	0.015	0.064	0.023
	Fair MINE NN	0.782	0.034	0.395	0.097	0.110	0.022	0.201	0.021	0.032	0.029	0.136	0.012
	Adel et al. [1]	0.836	0.005	0.384	0.037	0.170	0.027	0.371	0.035	0.058	0.027	0.057	0.007
	Zhang et al. [48]	0.787	0.134	0.377	0.085	0.153	0.056	0.313	0.087	0.037	0.022	0.063	0.046
	Madras et al. [31]	**0.725**	0.023	**0.312**	0.022	0.290	0.027	0.175	0.016	0.036	0.013	0.103	0.015
	Sadeghi et al. [40]	0.782	0.002	0.474	0.006	0.123	0.000	0.315	0,009	0.098	0.035	0.062	0.001
	Ours	0.783	0.031	0.369	0.074	**0.087**	0.031	0.173	0.044	0.011	0.006	0.043	0.012

		ACC		HGR_NN		HGR_KDE		HGR_RDC		MINE		FairQuant	
COMPAS	Standard NN	68.7%	0.243	0.363	0.005	0.326	0.003	0.325	0.008	0.046	0.028	0.140	0.001
	Grari et al. [19]	59.7%	2.943	0.147	0.000	0.121	0.002	0.101	0.007	0.004	0.001	0.018	0.018
	Fair MINE NN	54.4%	7.921	0.134	0.145	0.123	0.111	0.141	0.098	0.014	0.023	0.038	0.050
	Adel et al. [1]	55.4%	0.603	0.118	0.022	0.091	0.012	0.097	0.034	0.006	0.007	0.013	0.016
	Zhang et al. [48]	51.0%	3.550	0.116	0.000	0.081	0.003	0.086	0.010	0.002	0.003	0.010	0.005
	Madras et al. [31]	54.9%	2.221	0.175	0.000	0.116	0.015	0.107	0.026	0.005	0.003	0.011	0.020
	Sadeghi et al. [40]	54.3%	0.024	0.194	0.052	0.237	0.040	0.264	0.054	0.003	0.003	**0.003**	0.003
	Ours	**60.2**%	3.076	**0.063**	0.024	**0.068**	0.018	**0.067**	0.014	**0.001**	0.002	0.011	0.018
Default	Standard NN	82.1%	0.172	0.112	0.013	0.067	0.010	0.089	0.014	0.002	0.001	0.015	0.002
	Grari et al. [19]	79.9%	2.100	0.082	0.015	0.075	0.019	0.072	0.010	0.001	0.001	0.007	0.007
	Adel et al. [1]	79.2%	1.207	0.054	0.025	0.048	0.015	0.064	0.009	0.001	0.001	0.005	0.002
	Fair MINE NN	80.1%	2.184	0.093	0.020	0.057	0.002	0.066	0.012	0.001	0.001	0.008	0.001
	Zhang et al. [48]	77.9%	9.822	0.052	0.017	**0.044**	0.013	0.056	0.004	**0.000**	0.000	0.004	0.000
	Madras et al. [31]	78.3%	0.605	0.064	0.025	0.052	0.018	0.061	0.012	0.001	0.001	**0.003**	0.005
	Sadeghi et al. [40]	79.7%	0.236	0.074	0.019	0.062	0.013	0.098	0.041	0.002	0.002	**0.003**	0.002
	Ours	**80.8**%	0.286	**0.041**	0.008	**0.044**	0.006	**0.047**	0.002	0.001	0.002	0.005	0.001

Table 2: Experimental results - Best performance among fair algorithms in bold.

tasks. While being better in terms of accuracy, our fair representation algorithm achieves on four data sets (except on the Crime data set) the best level of fairness assessed by HGR estimation, MINE and FairQuant. On the Crime data set, the approach by Madras2018 [32] gets slightly better results on MSE and HGR estimation but not on the others metrics. Note, Adel [1] with the fair adversarial representation obtains (except on the Crime data set) better results than Zhang [48] which corresponds to the simple adversarial architecture.

What is the impact of mitigation weight ? In Figure 4, we plot the performance of different scenarios by displaying the HGR against the Accuracy with different values of the hyperparameter .

This plot was obtained on the COMPAS data set with 4 algorithms: ours, Adel et al. [1], Grari et al. [19] and Zhang et al. [48]. The different curves is obtained by Nadaraya-Watson kernel regression [8] between the Accuracy of the model and the HGR. Varying the hyperparameter allows to control the fairness/accuracy trade-off. Here, we clearly observe for all algorithms that the Accuracy, or predictive performance, decreases when fairness increases.

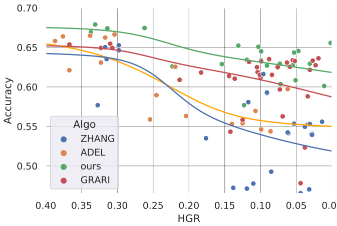


Fig. 4: Impact of hyperparameter (COMPAS data set)

Higher values of produce fairer predictions w.r.t the HGR, while near 0 values of the hyperparameter result in the optimization of the predictor loss with no fairness consideration (dots in the upper left corner of the graph). We note that, for all levels of predictive performance, our method outperforms the state of the art algorithms in terms of HGR.

Where should we apply mitigation in neural architectures ? In order to answer this question, and also further analyze the benefits of mitigation in neural representations compared to prediction retreatments as done in [19], we propose to consider various architectures of encoders h and predictors , with adversarial HGR mitigation being applied on the output of the encoder as depicted in fig.1. in appendix. To get comparable results between settings, we consider a constant full architecture (encoder + predictor), composed of 5 layers with 4 hidden layers with 32 units each.

In figure 5, we compare on the COMPAS dataset 5 different settings where mitigation is applied on a different layer of this full architecture: *LayerX* corresponds to a setting where mitigation is applied on the output of layer X (encoder of X layers, predictor of 5-X layers). *Layer5* thus corresponds to the prediciton retreatment approach proposed in [19] (no predictor function, the encoder function h directly outputs the prediction). *Layer3* is the standard setting used for our approach in the remaining of this paper.

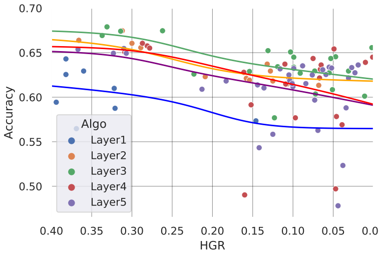


Fig. 5: Impact of hyperparameter (COMPAS data set) for various encoders h and predictors .

As in Figure 4, plotted results correspond to fairness-accuracy trade-offs obtained with different values of . We notice that applying mitigation too early in the architecture (Layer1) leads to very poor results. This can be explained by the fact that for this simple encoding setting, the encoder expressiveness is to weak to effectively remove non-linear dependencies w.r.t. the sensitive attribute, without removing too much useful information for prediction. At the contrary, when mitigation is applied late in the architecture (*Layer4* and *Layer5*) we observe generalization limits of the approach. While results on the training set are similar to those of *Layer3*, these settings lead to predictions at test time that are more dependent on the sensitive attribute. Due to L-Lipschitzness of neural network architectures, we know that $HGR(Z;S) \quad HGR(\ (Z);S)$. Acting on Z leads to remove bias from Z even for components ignored by the predictor in train. However, we argue that this allows to gain in stability at test time, when such components can be activated for new inputs, compared to late approaches, such as $Layer4$ or $Layer5$, which induce a greater variance of sensitive dependence of the output $\hat{Y}$. Mitigation at intermediate levels, such as $Layer3$, appears to correspond to the best trade-off expressiveness/generalization.

7 Conclusion

We present a new adversarial learning approach to produce fair representations with a continuous sensitive attribute. We leverage the HGR measure, which is efficient in capturing non-linear dependencies, and propose to minimize a neural estimation of the HGR between the latent representation and the sensitive attributes. This method proved to be very efficient for different fairness metrics on various artificial and real-world data sets. For further investigation, we will apply this architecture for information bottleneck purposes (e.g. for data privacy), that might be improved with an HGR_NN penalization as suggested in [4].

References

1. Adel, T., Valera, I., Ghahramani, Z., Weller, A.: One-network adversarial fairness. In: AAAI'19. vol. 33, pp. 2412–2420 (2019)
2. Alvi, M., Zisserman, A., Nellåker, C.: Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
3. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias. ProPublica, May 23, 2016 (2016)
4. Asoodeh, S., Alajaji, F., Linder, T.: On maximal correlation, mutual information and data privacy. In: 2015 IEEE 14th Canadian Workshop on Information Theory (CWIT). pp. 27–31. IEEE (2015)
5. Baharlouei, S., Nouiehed, M., Beirami, A., Razaviyayn, M.: Rényi fair inference. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
6. Belghazi, M.I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, R.D.: Mine: Mutual information neural estimation (2018)
7. Bellamy, R.K., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., et al.: Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. arXiv preprint arXiv:1810.01943 (2018)
8. Bierens, H.J.: The nadaraya-watson kernel regression function estimator (1988)
9. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 conference on empirical methods in natural language processing. pp. 120–128 (2006)
10. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: NIPS (2016)

11. Calmon, F.P., Wei, D., Vinzamuri, B., Ramamurthy, K.N., Varshney, K.R.: Optimized pre-processing for discrimination prevention. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 3995–4004 (2017)
12. Celis, L.E., Huang, L., Keswani, V., Vishnoi, N.K.: Classification with fairness constraints: A meta-algorithm with provable guarantees. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. pp. 319–328 (2019)
13. Chen, J., Kallus, N., Mao, X., Svacha, G., Udell, M.: Fairness under unawareness: Assessing disparity when protected class is unobserved. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. pp. 339–348 (2019)
14. Daume III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of artificial Intelligence research **26**, 101–126 (2006)
15. Dua, D., Graff, C.: UCI ml repository. http://archive.ics.uci.edu/ml (2017)
16. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
17. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
18. Geer, S.A., van de Geer, S.: Empirical Processes in M-estimation, vol. 6. Cambridge university press (2000)
19. Grari, V., Lamprier, S., Detyniecki, M.: Fairness-aware neural rényi minimization for continuous features. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. pp. 2262–2268. ijcai.org (2020), https://doi.org/10.24963/ijcai.2020/313
20. Hardoon, D.R., Shawe-Taylor, J.: Convergence analysis of kernel canonical correlation analysis: theory and practice. Machine learning **74**(1), 23–38 (2009)
21. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Advances in neural information processing systems. pp. 3315–3323 (2016)
22. Hornik, K., Stinchcombe, M., White, H., et al.: Multilayer feedforward networks are universal approximators. Neural networks **2**(5), 359–366 (1989)
23. Kamiran, F., Calders, T.: Data preprocessing techniques for classification without discrimination. Knowledge and Information Systems **33**(1), 1–33 (2012)
24. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Colored mnist dataset. https://github.com/feidfoe/learning-not-to-learn/tree/master/dataset/colored_mnist (2019)
25. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: Training deep neural networks with biased data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9012–9020 (2019)
26. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014), http://arxiv.org/abs/1312.6114
27. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database (2010)
28. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE international conference on computer vision. pp. 5542–5550 (2017)
29. Lopez-Paz, D., Hennig, P., Schölkopf, B.: The randomized dependence coefficient. In: Advances in neural information processing systems. pp. 1–9 (2013)
30. Louppe, G., Kagan, M., Cranmer, K.: Learning to pivot with adversarial networks. In: Advances in neural information processing systems. pp. 981–990 (2017)

31. Madras, D., Creager, E., Pitassi, T., Zemel, R.: Learning adversarially fair and transferable representations. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th ICML'18, pp. 3384–3393. (2018),
32. Madras, D., Creager, E., Pitassi, T., Zemel, R.: Fairness through causal awareness: Learning causal latent-variable models for biased data. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. pp. 349–358 (2019)
33. Mary, J., Calauzènes, C., Karoui, N.E.: Fairness-aware learning for continuous attributes and treatments. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th ICML'19, vol. 97, pp. 4382–4391. (2019), http://proceedings.mlr.press/v97/mary19a.html
34. Moyer, D., Gao, S., Brekelmans, R., Galstyan, A., Ver Steeg, G.: Invariant representations without adversarial training. In: Advances in Neural Information Processing Systems. pp. 9084–9093 (2018)
35. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML'13. pp. 10–18 (2013)
36. Pedreshi, D., Ruggieri, S., Turini, F.: Discrimination-aware data mining. In: KDD'08. p. 560 (2008). https://doi.org/10.1145/1401890.1401959, http://dl.acm.org/citation.cfm?doid=1401890.1401959
37. Póczos, B., Ghahramani, Z., Schneider, J.: Copula-based kernel dependency measures. In: Proceedings of the 29th ICML'12. pp. 1635–1642. (2012)
38. Ragonesi, R., Volpi, R., Cavazza, J., Murino, V.: Learning unbiased representations via mutual information backpropagation. arXiv preprint arXiv:2003.06430 (2020)
39. Rényi, A.: On measures of dependence. Acta mathematica hungarica **10**(3-4), 441–451 (1959)
40. Sadeghi, B., Yu, R., Boddeti, V.: On the global optima of kernelized adversarial representation learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7971–7979 (2019)
41. Székely, G.J., Rizzo, M.L., et al.: Brownian distance covariance. The annals of applied statistics **3**(4), 1236–1265 (2009)
42. The Institute of Actuaries of France: Pricing game 2015. https://freakonometrics.hypotheses.org/20191, online; accessed 14 August 2019
43. US Census Bureau: Us census demographic data. https://data.census.gov/cedsci/, online; accessed 03 April 2019
44. Witsenhausen, H.S.: On sequences of pairs of dependent random variables. SIAM Journal on Applied Mathematics **28**(1), 100–113 (1975)
45. Yeh, I.C., Lien, C.h.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst. Appl. **36**(2), 2473–2480 (Mar 2009). https://doi.org/10.1016/j.eswa.2007.12.020
46. Zafar, M.B., Valera, I., Rogriguez, M.G., Gummadi, K.P.: Fairness Constraints: Mechanisms for Fair Classification. In: AISTATS'17. pp. 962–970. Fort Lauderdale, FL, USA (20–22 Apr 2017)
47. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: ICML'13. pp. 325–333 (2013)
48. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: AAAI'18. pp. 335–340 (2018)