No shortcuts: Purging Spurious Correlations with Invertible Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Removing algorithmic bias to create 'fair' learning models has become a pressing topic in recent years. It has been shown that despite our best efforts to remove bias, under some conditions, such as dataset shift, 'fair' models actually exacerbate the problem and introduce more bias into our models. We consider the case where the training set is a severely biased sub-population of a dataset but unbiased unlabeled data is available. We develop a semi-supervised approach uses invertible neural networks to combat the problem. We leverage the invertibility of the network for exact density estimation in order keep as much non-sensitive information as possible. We demonstrate the effectiveness of our approach on a colorized MNIST dataset and datasets with tabular data.

1 Introduction

It is often difficult to control the relationships a machine learning (ML) system finds. Recent work [1] has shown that even state-of-the-art ML systems strongly rely on textures and not shapes when, e.g., classifying images of animals. This is not merely a theoretical problem, because it limits the generalisation of these systems in practice. If a human has never seen a black dog, they can still recognise it as a dog but an ML system may fail to do so.

The cause of the problem is that the ML system relies on *spurious correlations* that may exist in the training set, but not in the real world. What we would want the network to learn are the true relations that are invariant to these spurious features. However, if the training set contains spurious correlations, then an ML system cannot learn the true relations just from that dataset. We either need to supply an inductive bias (as recently investigated by Locatello et al. [2]) or give additional information.

A concrete dataset that allows us to investigate this problem in detail is the coloured MNIST dataset. In coloured MNIST, either the background or the digits are coloured and the relation between digit class and colour differs between training and test set. In the training set, there is a 1-to-1 correspondence between digit class and colour, but in the test set this correspondence does not exist. This can be understood as an extreme form of sampling bias where the training set only contains those samples that have very specific combinations of digit class and colour. The task with this dataset is to predict the correct digits in the test set. However, an ML system trained on the training set, learns the spurious correlation between colour and training label, because it is easier than learning to recognise the shape. In other words, the classifier is taking a *shortcut* that it shouldn't take. An ideal classifier would learn to recognise digits and be *invariant* to colour changes.

The concept of *invariance* is also important in the closely related field of algorithmic fairness. There, the goal is usually to make predictions that are invariant to *sensitive attributes* like gender and race. Furthermore, sampling bias has also been considered to be a problem of fairness. For example, Kallus and Zhou [3] found that the Stop, Question and Frisk dataset that was collected in New York City had very different demographics that New York City. This has been termed 'residual unfairness'[3]; even a classifier explicitly conditioned for fairness can yield grossly unfair predictions within the compass of the broader population, should the underlying training data be only an unrepresentative subset of it.

Our solution to the coloured MNIST problem is also applicable to these fairness problems. The approach is based on the idea of producing a representations of the inputs that is invariant to the spurious correlations. To this end, we assume the existence of unlabelled data which is ‘fair’ having minimal spurious correlation. We argue that this is a realistic assumption: while labelled data is relatively limited, given the resources needed to produce it, unlabelled data is abundant (census data, electoral rolls). The invariant representation is learned from this data. As we do not have labels for this data and cannot know which parts of the input are the most important for predicting the labels, we are using invertible neural networks to ensure that no information is lost that could be relevant for classification.

2 Background

2.1 Coloured MNIST

Coloured MNIST is a dataset with very strong spurious correlations. A standard classifier trained on the training set will not generalize to the test set. We present some previous works on this topic.

Adversarial approaches. This same problem of learning from biased data was tackled by [4] who a regularization loss that aims ot penalise mutual information between the feature embedding the spurious variable and thereby enforce their independence. This is realised with an adversarial training process, borrowing the gradient reversal technique showcased in Ganin et al. [5]. The authors construct the coloured MNIST dataset in two steps. First, 10 distinct colours are assigned to each digit uniquely; these colours are parameterize the means of 10 corresponding Normal distributions from which color samples are drawn. The standard deviation σ of the Normal distribution controls how close the sampled colours are to the mean colours. During training the colors are sampled abiding by this one-to-one colour mapping; at test time there is no such designation and colours are sampled randomly and unrestrictedly from the complete palette. As such a classifier that lazily minimises its loss by treating the pixel values as a lookup table falls flats at inference owing to a shift in the distribution of the spurious variables away that of the target.

For their training strategy, they train a neural network to predict the digit class y and have another network take one of the intermediate layers as input to predict the colour from it. The first network tries to prevent the adversary from making correct predictions. Thus, it has to discard the colour information.

For this approach to work, the adversary needs to distinguish between the digit class and the colour. To do this, the adversary gets to see the actual colour and not just the mean. As the sampled colour varies according to the Normal distribution, the actual colour and the digit class are not as strongly correlated as the mean colour and the digit class, which allows the network to disentangle the two. This works better, the larger σ is. A limitation is that this approach does not work at all when $\sigma = 0$. We address this limitation in our work.

Unsupervised approaches. There is a large literature on unsupervised disentangled representations; we only highlight one of the more recent ones. Locatello et al. [2] provide a theorem which states that the unsupervised learning of disentangled representations is impossible without inductive biases on both the data set and the models. Thus, such methods can usually only be used for one task or only for one kind of data.

2.2 Literature on Fair Representations

As mentioned in the introduction, the goal of producing invariant representations is similar to that of producing *fair* representations. In fairness problems, there is usually a *sensitive attribute* s (for example, gender or race), that should not be used to make decisions. A fair representation z is then one for which $Z \perp S$ holds and which is predictive of the class label y. The methods are often based on variational autoencoders (VAEs) [6–8].

The achieved fairness can be measured with one of several fairness metrics. These are however usually defined with respect to predictions and not representations. The two most important ones are Demographic Parity and Equality of Opportunity. It is not entirely clear which metric should be used to judge invariant (or fair) representations, but usually Demographic Parity is used [6–8]. Demographic Parity demands $\hat{Y} \perp S$ where $\hat{Y}$ refers to the predictions of the classifier.

A central aspect of all fairness methods is the accuracy-fairness trade-off. As mentioned, the fair representation should be invariant to s ($\rightarrow$ fairness) but still be predictive of y ($\rightarrow$ accuracy). These desiderata cannot, in general, be satisfied simultaneously if s and y are correlated (Oliver, do we have a reference for this?). We explore this trade-off with our method as well.

The methods for fair representations can in theory be used in an unsupervised fashion (without knowledge of y), but rarely are in practice. As Louizos et al. [6] state, the reason is that when s is removed from the representation, the representation can become *degenerate* with respect to y. For this reason, y usually is supplied during training and the representation is encouraged to be predictive of it. Our approach avoids this problem.

2.3 Invertible Neural Networks

Invertible neural networks are a class of neural network architectures that are characterized by three properties:

the following list has been copied from another paper

(i) The mapping from inputs to outputs is bijective

(ii) both forward and inverse mapping are efficiently computable

(iii) both mappings have a tractable Jacobian, which allows explicit computation of posterior (not posterior) probabilities.

Such flow-based models cachieve *exact* maximum likliehood estimation [9], warping a known base density with a series of invertible transformations by computing the resulting, highly-model, but still normalised, density, by leveraging the change of variable theorem.

Flow-GAN [10] combines the *exact* log-likelihood estimation of the invertible network with the adversarial training of a GAN.

$$\log P(z) = \log P(x) - \sum \log \left| \det \left(\frac{\mathrm{d}h_i}{\mathrm{d}h_{i-1}} \right) \right| \tag{1}$$

Invertible Networks are restricted to using transformations that are invertible and for which the determinant of the Jacobian can be tractably computed, most often by choosing transformations that ensure its lower or upper triangularity [11][12]. This rules out the use of most conventional Neural Network layers, but the set of known practical invertible transformations has grown steadily over recent years. Dinh et al. [13] proposed the use of invertible batch normalization while Kingma and Dhariwal [12] introduced Actnorm, that performs an affine (scale and shift) transformation akin to batch normalisation, and invertible 1x1 convolutions which generalise the permutation operation proposed in Dinh et al. [11] Finally, Dinh et al. [11] proposed Coupling Layers which split the input into two parts, apply a non-invertible transformation to one of the parts and then recombine them such that the whole operation is invertible.

That is, input vector $\boldsymbol{u}$ is split into two evenly sized vectors: $\boldsymbol{u} = [\boldsymbol{u}_1, \boldsymbol{u}_1]$. The output of the Coupling Layer is then a concatenation of vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, where $\boldsymbol{v}_1 = \boldsymbol{u}_1 + f(\boldsymbol{u}_2)$ and $\boldsymbol{v}_2 = \boldsymbol{u}_2$; and f is a non-invertible function.

2.3.1 Mutual Information

The cascade of homeomorphic layers allow us to preserve the mutual information between input and hidden representation

From InfoGAN “If A and B are related by a deterministic, invertible function, then maximal mutual information is attained”.

3 Using Invertible Networks to Create Fair Representations

3.1 General idea

For the task that we are considering, we assume that we have inputs $\boldsymbol{x} \in \mathcal{X}$ and corresponding labels $y_{\boldsymbol{x}} \in \mathcal{Y}$. Furthermore, there is a nuisance label $s_{\boldsymbol{x}} \in \mathcal{S}$ associated with each input $\boldsymbol{x}$ which we do *not* want to predict. Let X, S and Y be random variables that take on the values $\boldsymbol{x}$, s and y, respectively.

Both y and s are predictive of $\boldsymbol{x}$. So, $\mathcal{I}(X;Y), \mathcal{I}(X;S) > 0$, where $\mathcal{I}(\cdot;\cdot)$ is the mutual information. Note, however, that the conditional entropy is non-zero: $H(S|X) \neq 0$, i.e., S is not completely determined by X.

The difficulty emerges in the construction of the fully-supervised training dataset in which correspondence between S and Y is exaggerated compared to the test set. While demonstrably contrived, analogous scenarios arise naturally in a number of settings in which we only have access to the labels of a biasedly sampled subpopulation; since the nuisance variable is undesirably indicative of the class label, learning a model naïvely using this dataset would incur similar bias when deployed to a superpopulation devoid of it. Failure to account for this distributional shift can have dire consequences, especially when data is sparse; if our training and test distributions are not sufficiently well-matched, then the problem is not ignorable and the effects of the pathological variable need to be weeded out in order for us to achieve good generalization. Let (X^{tr}, S^{tr}, Y^{tr}) then be the random variables sampled for the training set and (X^{te}, S^{te}, Y^{te}) be the random variables for the test set. The training and test sets thus induce the following inequality of on the mutual information:

$$\mathcal{I}(S^{tr};Y^{tr}) \gg \mathcal{I}(S^{te};Y^{te}) \approx 0\,. \tag{2}$$

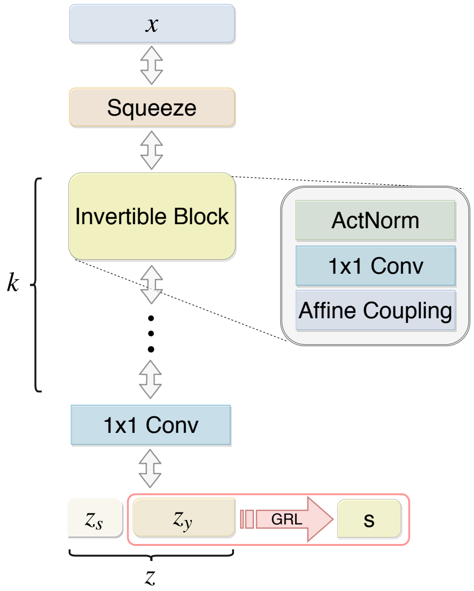


Figure 1: Architecture of our invertible network. We used the same normalising flow steps used in [12], which are repeated to depth k, with a final 1x1 Convolution to engender further mixing of the output dimensions. The network produces a bipartite latent encoding, z, with partitions distinguished by the independence enforced by means of a Gradient Reversal Layer (GRL) [5] between z_y and the nuisance variable, while z_s is unconstrained in the information it may contain.

We leverage recent advancements in flow-based modelling in the form of an invertible network f, which maps the inputs $\boldsymbol{x}$ to a representation $\boldsymbol{z}$: $f(\boldsymbol{x}) = \boldsymbol{z}$. $\boldsymbol{z}$ is a vector in which each element follows an isotropic Gaussian, $\mathcal{N}(z;0,\mathbb{I})$. We interpret the vector $\boldsymbol{z}$ as being the concatenation of two smaller vectors: $\boldsymbol{z} = [\boldsymbol{z}_s, \boldsymbol{z}_y]$. (The choice of indices will soon be clarified.) The lengths of $\boldsymbol{z}_s$ and $\boldsymbol{z}_y$ are a free parameter. As f is invertible, $\boldsymbol{x}$ can be recovered in the following manner:

$$\boldsymbol{x} = f^{-1}([\boldsymbol{z}_s, \boldsymbol{z}_y]) \tag{3}$$

We call the corresponding random variables Z_s and Z_y.

Our goal then is to make $\boldsymbol{z}_y$ not predictive of s:

$$I(Z_y;S) \overset{!}{=} 0 \tag{4}$$

z_s is not needed for our purposes but as we use an invertible network, the output dimension has to be equal to the input dimension. So we cannot just output z_y, but have to output z_s as well. In an analogy from thermodynamics, z_s can be thought of as the place to dump the waste heat from the network.

To accomplish the objective in Eq (4), we introduce an additional regularisation term which pushes the network to minimise the mutual information term. Our complete objective function is then given as:

$$\min_{\theta} \mathbb{E}_{x \sim X}[-\log P_\theta(x)] + \lambda I(Z_y; S) \tag{5}$$

where θ refers to the trainable parameters of the invertible network f. We get $P_\theta(x)$ from Eq (1).

In the fashion of a GAN, we optimise this loss by playing a min max game, in which our invertible network serves as the generative component. The adversary is an auxiliary classifier g, which receives z_y as input and attempts to predict the shortcut label s.We denote the parameters of the adversary as ϕ; for the parameters of the invertible network we use θ as before. Furthermore, let $b(\cdot)$ be the function that maps z to z_y: $b(z) = z_y$. The objective from Eq (5) is then realised as

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi} \mathbb{E}_{x \sim X}[\log P_\theta(x) - \lambda \mathcal{L}_c(g_\phi(b(f_\theta(x))); s)] \ . \tag{6}$$

However, the complication is that we want z_y to still be predictive of y, which precludes us from directly training on the target-labelled dataset (X^{tr}, S^{tr}, Y^{tr}), where y and s are so strongly correlated that removing the information about s also removes the information about y, since the loss offers no distinction in this respect. We therefore need another source of information that allows us to learn how to disentangle s and y. For this, we assume the existence of another dataset that follows a similar distribution to the test set, but for which we do not have access to the class labels. In practice, this is not an unreasonable assumption, as, while rigorously-annotated data is relatively hard to come by, unlabelled data, on the other hand, is a near-inexhaustible resource (e.g. census data, electoral rolls), and we are only restricted only in the sense that the spurious correlations we hope to prune are immanent in the features. y. We call this dataset 'meta dataset' and it consists of X^{me} and S^{me}. It fulfils $\mathcal{I}(S^{me}; Y^{me}) \approx 0$ (or rather, it would, if the class labels Y^{me} were available).

justify the existence of such a dataset

The concrete procedure is then as follows. First, the invertible network f is trained on (X^{me}, S^{me}). Then, the weights of f are frozen and f is used to encode the training set by taking in x and returning z_y. Finally, any classifier can be trained on the produced z_y. As all information about s has been purged from z_y, no spurious correlations between s and y are left. Thus, the classifier cannot take the shortcut of learning s and actually has to learn how to predict y.

How can we be sure that z_y contains enough information about y? This is where the strength of the invertible architecture comes into play. Due to the invertibility of the network, and homeomorphic mapping between layers, no information about the input is discarded. We know that it is always possible to recover x from z because f^{-1} exists and can do just that. So, as long as z_s does not contain the information about y, z_y must contain it. We can influence how much information z_s can hold by changing the size of z_s. A size should be chosen that is enough to contain all information about s, but not any more than that.

we should do experiments to show what effect the size of z_s has

For more advantages of the invertible architecture, see Section 3.2.

3.1.1 Preimages

For the CMNIT dataset, we found less success training on the representations $\boldsymbol{z}_y$ compared to the preimage obtained by performing an inverse pass after zeroing all elements of $\boldsymbol{z}_s$. Eq (3) defines how to obtain $\boldsymbol{x}$. In order to reconstruct only the part that is characterised by $\boldsymbol{z}_y$, we perform null-sampling: all elements of $\boldsymbol{z}_s$ are zeroed out, i.e. to the mean of the prior density imposed on $\boldsymbol{z}$: $\mathcal{N}(z; 0, I)$. Thus, $\boldsymbol{x}_{zy} = f^{-1}([\mathbf{0}, \boldsymbol{z}_y])$.

3.2 Advantages of an invertible encoder

Using an INN to gneerate our encodings z_y carries a number of advantages over, other than circumventing the need for invoking a lower bound on the log-likelihood. The invertibile property of the network guarantees the preservation of all semantically-meaningful information, y, regardless of how it is allocated in readout layer. Secondly, we conjecture that the encodings are more robust to out-of-distribution data. Whereas a normal autoencoder could map a previously seen input and a previously unseen input to the same representation, an invertible network cannot do this without violation of network's the bijective property. This ensures that no relevant information can be lost.

Related to that, invertible networks should not be susceptible to 'posterior collapse' [14].

Apart from these, there is another, more subtle, advantage. When considering fairness problems, it is actually an advantage to not make use of the class labels y when learning the fair representation. This is because the class labels can also be a source of bias [15, 16]. Our approach avoids this problem for the encodings, but the classifier that is trained on these encodings with the dataset labels can still be susceptible to this bias.

3.3 Network architecture

We use a downscaled version of GLOW [12] with an additional invertible 1x1 convolution before the final readout to conduce further mixing of the dimensions, finding this to slightly b The architecture of the network is similar to RealNVP [13], with the addition of the invertible 1×1 convolution from Glow [12] and the invertible batch normaliation layer from [11]

4 Experiments

Experiments on 3 datasets.

UCI Adult Dataset. Data from the 1994 U.S. Census with 12 features such as hours worked per week, age, work class and relationship status. The binary classification task is to predict an individual's earnings per annum. A positive class label denotes an income greater than $50,000$USD, and a negative label denotes an income less than or equal $50,000$USD. We use the binary label 'sex' as the sensitive attribute, where $s = M$ and $s = F$ describe a male and female respectively.

Coloured MNIST. We follow the general procedure outlined in [4] to create an augmented version of the MNIST handwritten digits dataset [17]. The augmentation scheme differs between the datasets we designate as spuriously correlated (task-dataset) and uncorrelated (pure-learning and task datasets) datasets. In both cases the digit samples are binarised and the digits coloured with an RGB value sampled from a univariate Gaussian distribution with mean specified by ten maximally dispersed colours (see **??**) for the specific values of our palette). The manner in which the mean values are prescribed to samples distinguishes the two dataset archetypes. In order to implant spurious correlation, mean values are bijectively mapped to a corresponding digit, such that the class and spurious variable become synonymous. Adjusting the standard deviation of the distribution is tantamount to scaling the bias of the dataset: when the parameter is small, there is minimal overlap between colours and, consequently, tight-coupling between the spurious and target variables, whereas large values have a decoupling effect as the former ceases to become a reliable, but misleading, indicator of the latter.

As such, we construct three variations of the same dataset. The pure-learning dataset and task datasets are constructed with the view to minimise correlations between the spurious and target variables. Our approach is premised on not having the target labels during the first-stage of training as if they were the problem would be trivially soluble. Conversely, such correlations are pronounced in the target-labelled data, such that there is a mismatch between the training and test distributions bridged by the INN. This labelled dataset is imagined to represent a biased subpopulation.

Experimental procedure.

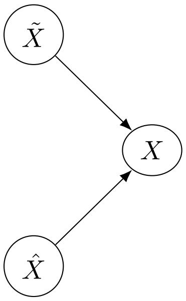
We split the test set into two creating a third dataset, which we cal the meta-set.

1) we train a model to distinguish between the features related to s and those which are not. We refer to this as a *(*pure-learning). In this stage we follow work from domain adaptation and fairness literature and transform the input into a new representation that is partitioned so that one part is domain, or s invariant, and to ensure that no information is lost, the second partition contains all information that is related to s. In this stage the model is trained on the meta-set.

Conceptually we can think of this as following a similar decomposition to Quadrianto et al Discovering Fair Representations in the Data Domain. However, Quadrianto et al. required both $\tilde{X}$ and $\hat{X}$ to be in the same space, seeing this as additive decomposition. We do not require $\tilde{x}$ and $\hat{x}$ to be in the same space, or require them to be the same number of dimensions. Because of this we assume a more complex decomposition.

Table 1: Performance Inv Disc False

	Adult	**CMNIST**
Majority Classifier		10%
Classifier on X		57.64%
Classifier on $X\&S$		20.35%
Classifier on $Z_{\neg S}$		
Classifier on Z_S		
Classifier on $f^{-1}(Z_y, Z^0_{\neg S,N}, Z^0_S, Z^0_{S,N})$		
Classifier on $f^{-1}(Z^0_y, Z^0_{\neg S,N}, Z_S, Z^0_{S,N})$		



We now have a decomposer, a model that takes X and returns two parts that together perfectly reconstruct X. $\tilde{X}$ represents the decomposition that is invariant to S and $\hat{X}$ the decomposition that is correlated with s.

We then feed our training data through the decomposer and obtain the two parts for our training set. We train a new classifier on this representation $f(\tilde{x}) \rightarrow y$.

To judge the performance of the model we train a classifier on the original training set, and judge the performance on the withheld test set.

We then re-train the same model on the reconstruction, but with Z_s set to all 0. Repeat for increasingly random.

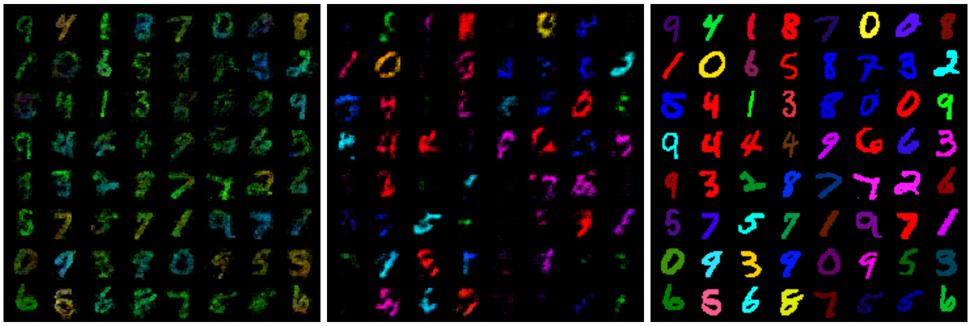


Figure 2: Preimage samples from the meta-train set. Left: Preimages obtained by setting all elements other than those in z_{true} to the mean of their prior distribution (null-sampling), demonstrating the success of our approach in successfully winnowing out spurious colour-information from z_{true}. Middle: Preimages obtained by null-sampling $z_{spurious}$. Right: Original samples.

We really need to show the grayscale images, and the images from task-train where colour = class.

5 Conclusion

blahblahblahblah

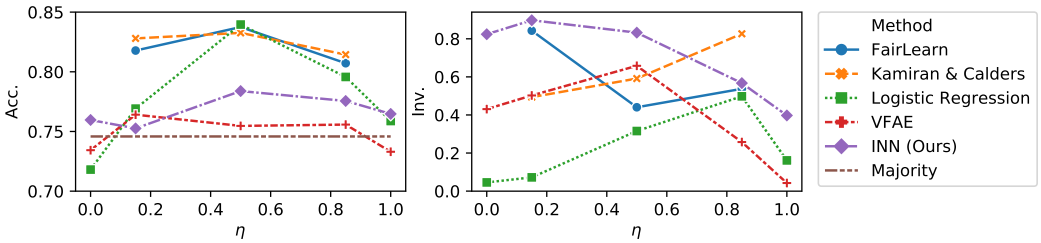


Figure 3: Comparison of our model against a number of baselines on the UCI Adult dataset. The balancing scale is denoted by η. Our model mostly outperforms the closest approach, VFAE across a range of values, removing more information about the sensitive label according to the Inv. measure. Our approach is particularly well suited to extreme cases where the labelled dataset is equivalent to the class label ($\eta = 0$ and $\eta = 1$) which fairness specific classification models (Kamiran&Calders and FairLearn.) cannot solve

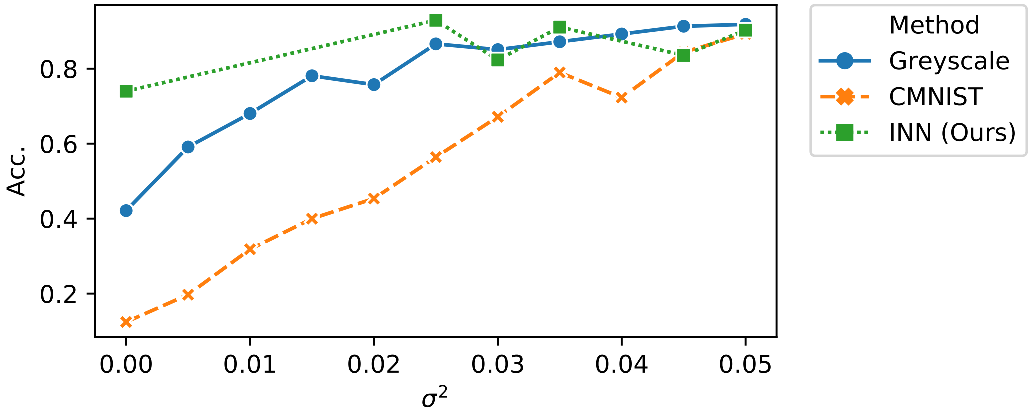


Figure 4: Comparison of our model against a number of baselines on the UCI Adult dataset. The balancing scale is denoted by η. Our model mostly outperforms the closest approach, VFAE across a range of values, removing more information about the sensitive label according to the Inv. measure. Our approach is particularly well suited to extreme cases where the labelled dataset is equivalent to the class label ($\eta = 0$ and $\eta = 1$) which fairness specific classification models (Kamiran&Calders and FairLearn.) cannot solve

References

[1] Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive Invariance Causes Adversarial Vulnerability. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[2] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, 2019.

[3] Nathan Kallus and Angela Zhou. Residual unfairness in fair machine learning from prejudiced data. In *International Conference on Machine Learning (ICML)*, pages 2444–2453, 2018.

[4] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Computer vision and pattern recognition (CVPR)*, 2019.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17:2096–2030, 2016.

[6] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The Variational Fair Autoencoder. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

[7] Harrison Edwards and Amos Storkey. Censoring Representations with an Adversary. In *International Conference on Learning Representations (ICLR)*, 2016.

[8] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2017.

[9] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning (ICML)*, 2015.

[10] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *AAAI Conference on Artificial Intelligence*, 2018.

[11] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representations (ICLR)*, 2014.

[12] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with Invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10236–10245, 2018.

[13] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[14] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30*, pages 6306–6315, 2017.

[15] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. *arXiv preprint arXiv:1901.04966*, 2019.

[16] Thomas Kehrenberg, Zexun Chen, and Novi Quadrianto. Tuning fairness by marginalizing latent target labels. *arXiv preprint arXiv:1810.05598*, 2018.

[17] Yann LeCun. Gradient-based learning applied to document recognition. 1998.