Differentiable Optimal Adversaries for Learning Fair Representations

Aaron Ferber[1], **Umang Gupta**[2], **Greg Ver Steeg**[2] **Bistra Dilkina**[1],
[1] University of Southern California
[2] USC Information Sciences Institute
{aferber, umanggup, dilkina}@usc.edu, {gregv}@isi.edu

Abstract

Fair representation learning is an important task in many real-world domains, with the goal of finding a performant model that obeys fairness requirements. We present an adversarial representation learning algorithm that learns an informative representation while not exposing sensitive features. Our goal is to train an embedding such that it has good performance on a target task while not exposing sensitive information as measured by the performance of an optimally trained adversary. Our approach directly trains the embedding with these dual objectives in mind by implicitly differentiating through the optimal adversary's training procedure. To this end, we derive implicit gradients of the optimal logistic regression parameters with respect to the input training embeddings, and use the fully-trained logistic regression as an adversary. As a result, we are able to train a model without alternating min max optimization, leading to better training stability and improved performance. Given the flexibility of our module for differentiable programming, we evaluate the impact of using implicit gradients in two adversarial fairness-centric formulations. We present quantitative results on the trade-offs of target and fairness tasks in several real-world domains.

Introduction

Deep learning models learn expressive data representations which make them applicable in many settings such as healthcare, criminal justice, or financial support. However, when used in automatic processes, practitioners often want to ensure that the model is performing fairly, with a variety of approaches enforcing different forms of fairness [Mehrabi *et al.*, 2019]. One way to approach fairness is to ensure the learned latent representation doesn't encode any sensitive information such as race or gender [Zemel *et al.*, 2013]. Several recent works learn fair representations through adversarial representation learning (ARL). In ARL approaches, an embedding model is trained such that a classifier has good performance on a target task, while also ensuring that an optimally trained adversary has poor performance extracting the sensitive information. Many of the ARL approaches use a multi-agent approach, alternating between training the embedding and adversary [Xie *et al.*, 2017; Roy and Boddeti, 2019]. However, these alternating ARL approaches disregard how changes in the embedding impact the corresponding new optimized adversary. As a result, they can suffer from training instability and suboptimality.

We propose an approach that directly trains the embedding by treating the optimal adversary as a differentiable function of the latent representation. We incorporate the adversarial loss in the training, by considering **adversary's model parameters as an implicit differentiable function of the embedding**. We derive gradients for the optimal logistic regression solution with respect to the input embedding, thus enabling backpropagation from the adversary loss and the application of the optimal adversary model to the embedding, through the optimality conditions of the adversary, back to the model parameters.

Our contributions are: 1) develop an end-to-end adversarial learning methodology that does not alternate between the target and sensitive attribute tasks, but instead optimizes both jointly; 2) derive how to incorporate optimal logistic regression as a differentiable layer in predictive models, which is interesting its own right; 3) show that our approaches often provide better tradeoffs between target and sensitive accuracy (as well as demographic parity) on diverse set of domains.

Problem Formulation

We consider that we are given data with features, target labels, and sensitive labels $\left\{(x^{(i)}, t^{(i)}, s^{(i)})\right\}_{i=1}^{n}$ with $x^{(i)} \in \mathbb{R}^{d_f}$ being $d_f -$ dimensional feature vectors, and target labels $t^{(i)} \in \mathbb{R}^{d_t}$ and $s^{(i)} \in 2^{c_s}$ being one-hot sensitive labels among c_s sensitive classes.

The goal is to find a classifier parameterized by embedding parameters θ_e, and target classifier θ_t such that the feature extractor with weights W, trained against our embedding θ_e, has poor performance. We can consider that the sensitive adversary is a linear logistic function of the embedding as in [Roy and Boddeti, 2019]. We consider the embedding function $z(x^{(i)}; \theta_e) \in \mathbb{R}^{d_e}$ to return a representation of an example in the latent space of dimensionality d_e.

We consider the 3-player game proposed in [Roy and Boddeti, 2019], where the adversary minimizes a loss $V_a(\theta_e, W)$, and the target classifier and embedding minimize their own

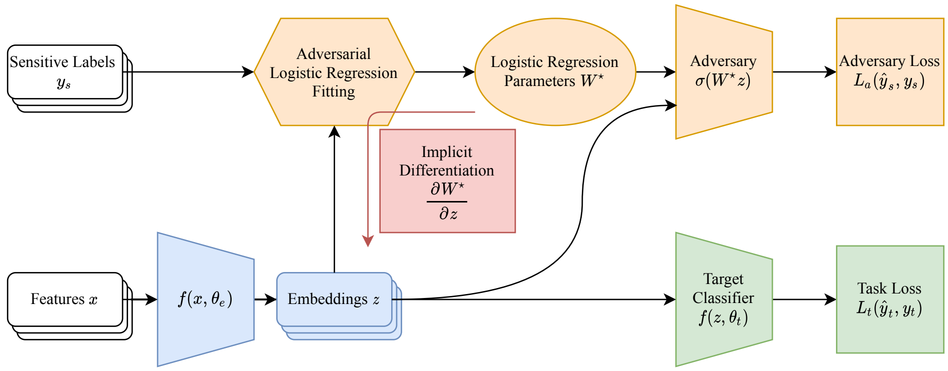

Figure 1: Fair representation learning model computation diagram.

loss, linearly weighting a penalty from the performance of the adversary $V_p(\theta_e, W)$ and the predictive performance on the target data $V_t(\theta_e, \theta_t)$. The adversarial penalty coefficient α is a tradeoff parameter that determines the weight on the adversarial penalty V_p. This setting is represented as the bi-level optimization problem:

$$\min_{\theta_e, \theta_t, W^*} \quad V_t(\theta_e, \theta_t) + \alpha V_p(\theta_e, W^*) \tag{1a}$$

$$\text{s.t.} \quad W^* = \arg\min_W V_a(\theta_e, W) \tag{1b}$$

Here Equation 1a represents the overall loss, a linear combination of the target classification performance and the sensitive penalty. Similarly, Equation 1b ensures the adversarial weights W^* optimize the adversary's objective V_a.

Considering that our setting consists of supervised learning tasks, we consider the target and adversary classifiers output predictions for targets $\hat{t}(z(x;\theta_e);\theta_t)$ and sensitive labels $\hat{s}(z(x;\theta_e);W)$ respectively. We define the target and adversary objective functions using standard supervised losses, with target classifier loss $V_t(\theta_e, \theta_t) = L_t(t, \hat{t}(z(x;\theta_e);\theta_t))$, and adversary classifier loss $V_a(\theta_e, W) = L_a(s, \hat{s}(z(x;\theta_e);W))$. We now define the target and adversary loss functions as well as the adversarial penalty to fully specify our problem.

Target loss function: V_t

This loss function represents the performance of the classifier on the target class. It is a supervised loss $V_t(\theta_e, \theta_t) = L_t(t, \hat{t}(z(x;\theta_e);\theta_t))$ with L_t being a differentiable supervised loss function such as cross-entropy loss.

Adversary loss function: V_a

We consider the adversary to be solving a logistic regression problem, so our loss function on the adversary's weights W is considered to be the logistic loss with L2 penalty. Given the one-hot encoded sensitive targets s, and softmax predictions $\hat{s}(z(x;\theta_e);W)) = \sigma(W^T z^{(i)}(x;\theta_e))$, the softmax regression loss is $V_a(\theta_e, W) = L_a(s, \hat{s}(z(x;\theta_e);W)) = -\sum_{i=1}^{n} s^{(i)T}\sigma(W^T z^{(i)}(x;\theta_e)) + \|W\|_2^2$. Although the functions here are known to be differentiable, our approach will take gradients of the optimal weights W^* with respect to the input embeddings $z(x;\theta_e)$ to perform backpropagation.

Adversarial penalty: V_p

Lastly, given our flexible formulation, we can consider both formulations of adversarial representation learning (ARL) presented in [Roy and Boddeti, 2019], one penalizing the embedding based on the entropy of the optimal adversary (referred to as MaxEnt-ARL), and another based on adversary's classification performance (referred to as ML-ARL).

Optimizing the entropy considers that we want to maximize the entropy of the sensitive classifier's predictions. For simplicity, we can consider minimizing the cross-entropy between the uniform distribution and the predictions $\hat{s}(x;\theta_e, W^*)$. Thus we can formulate entropy maximization as minimizing $V_p(\theta_e, W^*) = L_p(s, \hat{s}(z(x;\theta_e);W^*)) = CE(1/c_s, \hat{s}(z(x;\theta_e);W^*))$, with $CE(p,q)$ being the cross entropy between p and q i.e. $CE(p,q) = -\sum_{i=1}^{c_s} p \log_2 q$. Note that in this setting, the adversary penalty disregards the sensitive labels, but the sensitive labels will still be used in the training of the adversary.

To encode ML-ARL in our formulation, we can consider the adversary penalty V_p to be the negative of the classification performance of the worst-case adversary. In this case we would have $V_p(\theta_e, W^*) = L_p(s, \hat{s}(z(x;\theta_e);W^*)) = -CE(s, \hat{s}(z(x;\theta_e);W^*))$, or the negative of the cross entropy between the sensitive labels and the adversary's predictions of the sensitive labels.

Evaluating the objective: Equation 1a

Given this problem formulation, we can clearly evaluate the objective function we are trying to minimize given embedding and target parameters θ_e, θ_t.

Examining the pipeline in algorithm 1 and visualized in Figure 1, we can now begin to see that what is easily differentiable in parameters θ_e, θ_t. Clearly step 1, 2, and 3 are known differentiable functions of the weights so standard libraries will handle backpropagation. Furthermore, step 5 is clearly a differentiable function of both the embedding and the optimal logistic layer so a standard autograd library will chain together gradients from softmax and product rule for differentiating $W^{*T}\mathbf{z}$. In step 6, the adversarial penalty loss is a standard cross entropy loss on the predictions. Lastly, the returned loss is a simple linear combination. Therefore, the only component that does not yet have readily-available gradient computation is step 4.

Algorithm 1: Compute objective function

1 **Embed** $\mathbf{z} \leftarrow z(x; \theta_e)$
2 **Predict targets** $\hat{\mathbf{t}} \leftarrow \hat{t}(\mathbf{z}; \theta_t)$
3 **Compute** $V_t \leftarrow L_t(t, \hat{\mathbf{t}})$
4 **Optimize Logistic Regression**
$$W^* \leftarrow \arg\min_W - \sum_{i=1}^{n} s^{(i)T} \sigma(W^T \mathbf{z}) + \|W\|_2^2$$
5 **Predict sensitive** $\hat{\mathbf{s}} \leftarrow \sigma(W^{*T} \mathbf{z})$
6 **Compute** $V_p \leftarrow L_p(s, \hat{\mathbf{s}})$
7 **Return** $V_t + \alpha V_p$

Our approach derives gradients of the optimal solution to the logistic regression problem W^* with respect to the input feature embeddings $\mathbf{z}$ so that we can backpropagate from the loss function, through the logistic regression training, to the original embedding for training.

Differentiating Through Adversary Optimization

Given that the rest of the pipeline is specified for both forward and backward passes, here we investigate gradients for the remaining step: step 4 in algorithm 1. We derive gradients of the optimal logistic regression parameters $W^* \in \mathbb{R}^{d_e \times c_s}$ with respect to the input features $\mathbf{z}$. Here the logistic regression makes predictions for c_s classes from d_e features. Given that the objective of the logistic regression is convex in it's weights [Boyd and Vandenberghe, 2004], we know that the optimal solution is defined as the solution where the gradient of the objective function is 0. Thus we know that W^* must satisfy the constraints

$$0 = \nabla_W \bigg|_{W^*} \left(- \sum_{i=1}^{n} s_i^T \sigma\left(W^T \mathbf{z}\right) + \|W\|_2^2 \right).$$

We write the gradients of the logistic regression objective with respect to the model parameters evaluated at optimality:

$$\nabla_W \bigg|_{W^*} \left(- \sum_{i=1}^{n} s_i^T \sigma\left(W^T \mathbf{z}\right) + \|W\|_2^2 \right)$$
$$= \sum_{i=1}^{n} \left(\sigma\left(W^{*T} \mathbf{z}\right) - s_i^T \right) \mathbf{z} + 2W^*$$

Here we can see that the trained parameters W^* are an implicitly defined function of the embedding $\mathbf{z}$, namely those which ensure the gradients are 0. Thus, to find gradients of the optimal parameters W^* with respect to a single embedding $\mathbf{z}^{\mathbf{i_0}}$ of example i_0, we can relate changes in W^* to changes in $\mathbf{z}^{\mathbf{i_0}}$ as those satisfying a set of equations. Specifically, we have that for each sensitive class $k \in [c_s]$,

$$\sum_{i=1}^{n} \left[\sum_{c=1}^{c_s} (\delta_{c,k} - \hat{s}_c^{(i)}) s_k^{(i)} \left(dW_c^{*T} \mathbf{z}^{(i)} + W_c^{*T} d\mathbf{z}^{(i)} \right) \right] \mathbf{z}^{(i)} +$$
$$+ (\hat{s}_k^{(i_0)} - s_k^{(i_0)}) d\mathbf{z}^{(i_0)} = 0,$$

where δ is the Kronecker delta.

Experiments

We train all methods with early stopping based on the validation loss of the encoder. We selected model hyperparameters and architectures for the embedding model and target classifier from [Roy and Boddeti, 2019].

Methods

MLP is a fairness-unaware neural network classifier to minimize a target loss without regard for the sensitive classifier.

CE-ARL [Xie *et al.*, 2017], Ent-ARL [Roy and Boddeti, 2019] are standard alternating approaches. CE-ARL imposes an adversarial penalty on the embedding of the negative cross entropy loss. EntARL uses the prediction entropy as the adversaryial loss.

CE-OptARL, Ent-OptARL are the corresponding variants of our method which penalize our embedding using the negative of the adversary's cross-entropy and the adversary's output entropy respectively. This method follows the same mathematical program as [Xie *et al.*, 2017] but fully optimizes the adversary model instead of iteratively training the embedding and the adversary.

Datasets

COMPAS [Angwin *et al.*, 2016] has defendant data where we aim to predict whether the person will recidivate within 2 years, being sensitive to race. **Heritage Health** data contains features about 60,000 patients from insurance claims and physician records. As in [Madras *et al.*, 2018; Song *et al.*, 2018], we consider the target task of predicting whether the Charlson Index is nonzero being sensitive to age group (9 age groups total). **Adult** is a UCI dataset [Frank and Asuncion, 2010] of 40,000 adults where the task is to predict whether the income is above \$50,000, while being sensitive to gender. **German** is another UCI dataset [Frank and Asuncion, 2010] of 1,000 people where the task is to predict low or high credit score while being sensitive to gender.

Evaluation

Sensitive Accuracy evaluates the sensitive information in an embedding. We train a logistic regression classifier to predict the sensitive features from the embeddings of the training set and evaluate the test accuracy of that fully-trained model.

Demographic Parity Difference. The demographic parity difference Δ_{DP} [Dwork *et al.*, 2011] measures the difference in selection rates between sensitive groups and is defined for targets predictions $\hat{t}$ and sensitive labels s as

$$\Delta_{DP} = |P(\hat{t} = 1 | s = 1) - P(\hat{t} = 1 | s = 0)|.$$

Results. Table 1 reports best target accuracy achieved by each method at different cutoffs of sensitive accuracy and demographic parity (Δ_{DP}). Results spanning this tradeoff are collected by varying the adversarial penalty coefficient α between 0.1 and 1000 by factors of 10, for all methods but MLP. Each method and parameter setting is run with 5 random seeds. We observe that our approaches, CE-OptARL and Ent-OptARL, outperform their respective standard ARL counterparts. The OptARL approaches provide better target accuracy at the given sensitive accuracy cutoffs, demonstrating that differentiating through the adversary's optimization

COMPAS	sens acc < 0.98	sens acc < 0.99	sens acc < 1.00	$\Delta_{DP} < 0.10$	$\Delta_{DP} < 0.15$	$\Delta_{DP} < 0.20$
MLP	-	-	0.6961	-	0.6945	0.6961
CE-ARL	-	0.5429	0.6848	0.6005	0.6572	0.6848
CE-OptARL (ours)	**0.701**	**0.701**	**0.701**	0.6969	**0.701**	**0.701**
Ent-ARL	-	0.6921	0.6921	0.6669	0.6872	0.6921
Ent-OptARL (ours)	**0.701**	**0.701**	**0.701**	**0.7002**	0.7002	0.7002
Health	sens acc < 0.30	sens acc < 0.32	sens acc < 0.34	$\Delta_{DP} < 0.40$	$\Delta_{DP} < 0.60$	$\Delta_{DP} < 0.80$
MLP	-	0.8177	0.8192	-	0.8192	0.8192
CE-ARL	-	0.8176	0.8176	**0.708**	0.8176	0.8176
CE-OptARL (ours)	0.8165	0.8178	0.8178	-	0.8178	0.8178
Ent-ARL	0.7492	0.8184	0.8194	0.7066	0.8194	0.8194
Ent-OptARL (ours)	**0.8203**	**0.8203**	**0.8203**	0.6883	**0.8203**	**0.8203**
Adult	sens acc < 0.68	sens acc < 0.69	sens acc < 0.70	$\Delta_{DP} < 0.10$	$\Delta_{DP} < 0.15$	$\Delta_{DP} < 0.20$
MLP	0.8216	0.8242	0.8242	-	0.8242	0.8242
CE-ARL	0.8163	0.8163	0.8163	0.814	0.8163	0.8163
CE-OptARL (ours)	**0.8248**	0.8248	0.8248	**0.8167**	0.8248	0.8248
Ent-ARL	0.8186	0.821	0.821	0.8153	0.821	0.821
Ent-OptARL (ours)	0.8192	**0.827**	**0.827**	0.8013	**0.827**	**0.827**
German	sens acc < 0.90	sens acc < 0.95	sens acc < 1.00	$\Delta_{DP} < 0.02$	$\Delta_{DP} < 0.03$	$\Delta_{DP} < 0.04$
MLP	0.6933	0.73	0.73	0.6933	0.6933	0.6933
CE-ARL	0.6967	0.71	0.71	0.6967	0.6967	0.6967
CE-OptARL (ours)	0.72	0.72	0.72	0.7	0.72	0.72
Ent-ARL	0.7067	0.7067	0.7067	0.69	0.7	0.7067
Ent-OptARL (ours)	**0.7333**	**0.7333**	**0.7333**	**0.7267**	**0.7267**	**0.7333**

Table 1: Target accuracy at fairness cutoffs: We present test results for maximum target accuracy at given cutoffs on the accuracy of a fully-trained adversary (sens acc), as well as on the demographic parity (Δ_{DP}). These cutoffs are selected for each dataset to span the distribution in the results. Metrics are obtained by varying the adversarial penalty coefficient α between 0.1 and 1000 by factors of 10.

procedure is able to improve the desired effect of adversarial representation learning. In addition, we observe that our methods provide better target accuracy at most Δ_{DP} cutoffs, with the exception of the Adult and Health datasets only at the lowest Δ_{DP} threshold.

Related Work

In [Zemel *et al.*, 2013], the authors optimize clusters of individuals to generate discrete and fair representations. [Calmon *et al.*, 2017] optimize a random data transformation preserving utility for downstream tasks but obfuscating sensitive attributes. Approaches with alternating training such as [Roy and Boddeti, 2019; **?**] iteratively train an embedding along with an adversary by optimizing the models with respective parameters and objectives. These approaches generally formulate the objective of the embedding using an optimal adversary; however, the optimiziation procedures don't differentiate through the adversary's optimization procedure, and instead treat the adversary's parameters as constants during backpropagation to the embedding model. Previous work has considered a similar differentiable optimization approach for meta-learning, proposing a differentiable svm optimization algorithm [Lee *et al.*, 2017], closed-form ridge-regression formulation, or iterative logistic regression solver [Bertinetto *et al.*, 2019] as a last-layer fine tuning methodology. In our work, we consider adversarial representation learning, and directly differentiate through the optimality condition of logistic regression rather than the unrolled solver iterates.

Discussion

We improve adversarial representation learning approaches by implicitly defining the fully-trained adversary as a differentiable function of the embedding, allowing us to directly train the representation with gradient information from the adversary's optimality conditions. In particular, we provide a novel methodology for computing gradients of the optimal logistic regression adversary with respect to the input embeddings. This approach can be viewed in several lights. One interpretation is that we fully backpropagate the global loss (the penalty of the adversary and the target performance) through the adversary optimization to the embedding model's parameters. Another facet is that we train the embedding with explicit information about how the fully-trained adversary will change due to changes in the embedding. Lastly, we can view the overall optimization procedure as optimizing the embedding for the loss it observes at equilibrium in the 3-player game formulation suggested in [Roy and Boddeti, 2019].

The evaluation using four different datasets, spanning criminal risk assessment, healthcare, and finance, showed that our optimal adversary approach improves the performance of both adversarial representation learning baselines. In particular, we showed we are able to (almost always) provide better target accuracy at different thresholds on fairness in terms of both sensitive accuracy and demographic parity.

Since our contribution enables logistic regression fitting as a differentiable layer in any end-to-end learning, we hope in future work to evaluate other relevant settings.

References

[Angwin *et al.*, 2016] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23:2016, 2016.

[Bertinetto *et al.*, 2019] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

[Boyd and Vandenberghe, 2004] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Calmon *et al.*, 2017] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3992–4001. Curran Associates, Inc., 2017.

[Dwork *et al.*, 2011] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness Through Awareness, apr 2011.

[Frank and Asuncion, 2010] Andrew Frank and Arthur Asuncion. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of information and computer science*, 213(11), 2010.

[Lee *et al.*, 2017] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017.

[Madras *et al.*, 2018] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393, 2018.

[Mehrabi *et al.*, 2019] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[Roy and Boddeti, 2019] Proteek Chandan Roy and Vishnu Naresh Boddeti. Mitigating information leakage in image representations: A maximum entropy approach. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Song *et al.*, 2018] Jiaming Song, , Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. *arXiv preprint arXiv:1812.04218*, 2018.

[Xie *et al.*, 2017] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, pages 585–596, 2017.

[Zemel *et al.*, 2013] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.