Differentiable Optimal Adversaries for Learning Fair Representations

Aaron Ferber¹, Umang Gupta², Greg Ver Steeg² Bistra Dilkina¹,

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.

Problem Formulation

We consider that we are given data with features, target labels, and sensitive labels \( \{(x(i), t(i), s(i))\}_{i=1}^n \) 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 \( \theta_e \), and target classifier \( \theta_t \) such that the feature extractor with weights \( W \), trained against our embedding \( \theta_e \), has poor performance. We can consider that the sensitive adversary is a linear logistic function of the embedding as in [Roy and Bodde, 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 Bodde, 2019], where the adversary minimizes a loss \( \hat{V}_a(\theta_e, W) \), and the target classifier and embedding minimize their own
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 \( \alpha \) is a tradeoff parameter that determines the weight on the adversarial penalty \( V_p \). This setting is represented as the bilevel optimization problem:

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

subject to \( W^* = \arg \min_W V_t(\theta_e, W) \) \( \tag{1b} \)

Here Equation 1a represents the overall loss, a linear combination of the target classification performance and the sensitivity 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 \( V_t(\theta_e, \theta_t) = \ell_t(t, \hat{t}(z(x; \theta_e); \theta_t)) \), and adversary classifier loss \( V_a(\theta_e, W) = \ell_a(s, \hat{s}(z(x; \theta_e); W)) \). 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 \( \hat{s}(z(x; \theta_e); W) = \sigma(W^T z(x; \theta_e)) \), the softmax regression loss is \( V_a(\theta_e, W) = \ell_a(s, \hat{s}(z(x; \theta_e); W)) = -\sum_{i=1}^n s^{(i)} \sigma(W^T z^{(i)}(x; \theta_e)) + \|W\|_2^2 \). 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 \( \hat{s}(z(x; \theta_e), W^*) \). 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. \( CE(p, q) = -\sum_{i \leq 1} p \log_2 q \). 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 \( \theta_e, \theta_t \).

Examining the pipeline in algorithm 1 and visualized in Figure 1, we can now begin to see that what is easily differentiable in parameters \( \theta_e, \theta_t \). 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 \( W^T z \). 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 $z \leftarrow z(x; \theta_z)$
2. Predict targets $\hat{t} \leftarrow \hat{t}(z; \theta_t)$
3. Compute $V_i \leftarrow L_i(t, \hat{t})$
4. Optimize Logistic Regression
   $$W^* \leftarrow \arg \min_W - \sum_{i=1}^{n} s^{(i)} T \sigma(W^T z) + ||W||^2_2$$
5. Predict sensitive $\hat{s} \leftarrow \sigma(W^* T z)$
6. Compute $V_p \leftarrow L_p(s, \hat{s})$
7. Return $V_i + \alpha V_p$

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

Differentiating Through Adversarial 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 $W^* \in \mathbb{R}^{d_x \times c_s}$
with respect to the input features $z$. Here the logistic regres-
sion makes predictions for $c_s$ classes from $d_x$ features. Given
that the objective of the logistic regression is convex in its
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

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

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

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

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

$$\sum_{i=1}^{n} \sum_{c=1}^{c_s} \left( \delta_{c,k} - \hat{s}_c(i) s_k^{(i)} \right) \left( dW^T c z^{(i)} + W^T c d z^{(i)} \right) z^{(i)} + (\hat{s}_k(i) - s_k(i)) d a^{(i)} = 0,$$

where $\delta$ is the Kronecker delta.

Experiments

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

Methods

MLP is a fairness-unaware neural network classifier to mini-
mize 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 ad-
versarial loss.

CE-OptARL, Ent-OptARL are the corresponding vari-
ants 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 opti-
mizes 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 con-
tains 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 $\Delta_{DP}$ [Dwork et al., 2011] measures the difference
in selection rates between sensitive groups and is defined for
targets predictions $t$ 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 ($\Delta_{DP}$). Results spanning this tradeoff
are collected by varying the adversarial penalty coefficient $\alpha$
between 0.1 and 1000 by factors of 10, for all methods but
MLP. Each method and parameter setting is run with 5 ran-
dom 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, demonstrat-
ing that differentiating through the adversary’s optimization
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 ($\Delta_D P$). These cutoffs are selected for each dataset to span the distribution in the results. Metrics are obtained by varying the adversarial penalty coefficient $\alpha$ between 0.1 and 1000 by factors of 10.

**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


