Overview of the Fairness Methods

1 Learning Fair Representations

This algorithm learns fair representations to ensure fairness. X denotes the entire data set of individuals. S is a binary random variable representing whether or not a given individual is a member of the protected set. Z is a multinomial random variable where each of the K values represents one of the intermediate set of prototypes. Each prototype is associated with a vector v_k in the same space as the individuals x. Define a distance measure d, e.g., $d(x_n, v_k) = \|x_n - v_k\|_2$.

A natural probabilistic mapping from X to Z via the softmax:

$$P(Z = k \mid x) = exp(-d(x, v_k)) / \sum_{j=1}^{K} exp(-d(x, v_j)) \tag{1}$$

Let $M_{nk} = P(Z = k \mid x_n)$. Define

$$M_k^+ = \frac{1}{\left|X_0^+\right|} \sum_{n \in X_0^+} M_{nk} \tag{2}$$

and M_k^- is defined similarly. Let $L_z = \sum_{k=1}^{K} \left|M_k^+ - M_k^-\right|$. This term ensures statistical parity.

Let $\hat{x}_n = \sum_{k=1}^{K} M_{nk} v_k$ be the reconstructions of x_n from Z. Let $L_x = \sum_{n=1}^{N} (x_n - \hat{x}_n)^2$, which constrains the mapping to Z to be a good description of X.

Let $L_y = \sum_{n=1}^{N} -y_n log \hat{y}_n - (1 - y_n) log(1 - \hat{y}_n)$ where $\hat{y}_n = \sum_{k=1}^{K} M_{nk} w_k$ is the prediction for y_n. The w_k values are constrained between 0 and 1.

The learning system minimizes the objective:

$$L = A_z \cdot L_z + A_x \cdot L_x + A_y \cdot L_y \tag{3}$$

where A_x, A_y, A_z are hyperparameters governing the tradeoff between the system desiderata.

In order to allow different input features to have different levels of impact, define

$$d(x_n, v_k, \alpha) = \sum_{i=1}^{D} \alpha_i (x_{ni} - v_{ki})^2 \tag{4}$$

and this model can be extended by using different parameter vectors α^+ and α^- for the protected and unprotected groups respectively. These parameters together with $\{v_k\}_{k=1}^K, w$ are optimized.

2 Fairness Constraints: Mechanisms for Fair Classification

This method considers the signed distance from the users' feature vectors to the decision boundary $\{d_\theta(x_i)\}_{i=1}^N$, and compute

$$Cov(z, d_\theta(x)) \approx \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})d_\theta(x_i) \tag{5}$$

where z is the protected feature. This is a convex function with respect to the decision boundary parameters θ.

2.1 Maximizing accuracy under fairness constraints

Let $L(\theta)$ be the loss function.

$$\begin{aligned} \min \quad & L(\theta) \\ \text{s.t.} \quad & \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})d_\theta(x_i) \leq c \\ & \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})d_\theta(x_i) \geq -c \end{aligned} \tag{6}$$

where c trades off fairness and accuracy.

2.2 Maximizing fairness under accuracy constraints

$$\begin{aligned} \min \quad & \left| \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})d_\theta(x_i) \right| \\ \text{s.t.} \quad & L(\theta) \leq (1+\gamma)L(\theta^*) \end{aligned} \tag{7}$$

where $L(\theta^*)$ denotes the optimal loss over the training set provided by the unconstrained classifier and $\gamma \geq 0$ specifies the maximum additional loss with respect to the loss provided by the unconstrained classifier.

3 Fairness Beyond Disparate Treatment and Disparate Impact: Learning Classification without Disparate Mistreatment

This method considers

$$Cov(z, g_\theta(y,x)) \approx \frac{1}{N} \sum_{(x,y,z)\in\mathcal{D}} (z-\bar{z}) g_\theta(y,x) \tag{8}$$

where g_θ can be defined as

$$\begin{aligned}
g_\theta(y,x) &= min(0, yd_\theta(x)) \\
g_\theta(y,x) &= min(0, \frac{1-y}{2} yd_\theta(x)) \\
g_\theta(y,x) &= min(0, \frac{1+y}{2} yd_\theta(x))
\end{aligned}$$

However, since the problem

$$\begin{aligned}
\min \quad & L(\theta) \\
\text{s.t.} \quad & \frac{1}{N} \sum_{(x,y,z)\in\mathcal{D}} (z-\bar{z}) g_\theta(y,x) \leq c \\
& \frac{1}{N} \sum_{(x,y,z)\in\mathcal{D}} (z-\bar{z}) g_\theta(y,x) \geq -c
\end{aligned} \tag{9}$$

is nonconvex, the constraints are converted into a Disciplined Convex Concave Program which can be solved efficiently.

$$\begin{aligned}
\min \quad & L(\theta) \\
\text{s.t.} \quad & \frac{-N_1}{N} \sum_{(x,y)\in\mathcal{D}_0} g_\theta(y,x) + \frac{N_0}{N} \sum_{(x,y)\in\mathcal{D}_\infty} g_\theta(y,x) \leq c \\
& \frac{-N_1}{N} \sum_{(x,y)\in\mathcal{D}_0} g_\theta(y,x) + \frac{N_0}{N} \sum_{(x,y)\in\mathcal{D}_\infty} g_\theta(y,x) \geq -c
\end{aligned} \tag{10}$$

where $\mathcal{D}_0$ and $\mathcal{D}_1$ are the subsets of the training dataset $\mathcal{D}$ taking values $z = 0$ and $z = 1$, respectively. $N_0 = |D_0|$ and $N_1 = |D_1|$.

4 Fairness-aware Classifier with Prejudice Remover Regularizer

This method considers the objective function

$$-L(\mathcal{D};\theta) + \eta R(\mathcal{D},\theta) + \frac{\lambda}{2}\|\theta\|_2^2 \tag{11}$$

where λ and η are positive regularization parameters, $R(\mathcal{D}, \theta)$ is the prejudice index and

$$L(\mathcal{D}; \theta) = \sum_{(y_i, x_i, s_i) \in \mathcal{D}} log M(y_i \mid x_i, s_i; \theta) \tag{12}$$

The prejudice index is defined as

$$PI = \sum_{Y,S} \hat{P}(Y, S) log \frac{\hat{P}(Y, S)}{\hat{P}(S)\hat{P}(Y)} \tag{13}$$

which can be written as

$$\frac{1}{|\mathcal{D}|} \sum_{(x_i, s_i) \in \mathcal{D}} \sum_{y \in \{0,1\}} M(y \mid x_i, s_i; \theta) log \frac{\hat{P}(y \mid s_i)}{\hat{P}(y)} \tag{14}$$

where

$$\hat{P}(y \mid s) \approx \frac{\sum_{(x_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s} M(y \mid x_i, s; \theta)}{|\{(x_i, s_i) \in \mathcal{D} \text{ s.t. } s_i = s\}|} \tag{15}$$

$$\hat{P}(y) \approx \frac{\sum_{(x_i, s_i) \in \mathcal{D}} M(y \mid x_i, s_i; \theta)}{|\mathcal{D}|} \tag{16}$$

5 Handling Conditional Discrimination

This method considers

$$D_{all} = D_{expl} + D_{bad} \tag{17}$$

where $D_{all} = P(y = + \mid s = m) - P(y = + \mid s = f)$. D_{expl} is the explainable part of the discrimination. s is the protected variable.

Let

$$P^*(+ \mid e_i) = \frac{P(+ \mid e_i, m) + P(+ \mid e_i, f)}{2} \tag{18}$$

Then

$$\begin{aligned} D_{expl} &= \sum_{i=1}^{k} P(e_i \mid m) P^*(+ \mid e_i) - \sum_{i=1}^{k} P(e_i \mid f) P^*(+ \mid e_i) \\ &= \sum_{i=1}^{k} (P(e_i \mid m) - P(e_i \mid f)) P^*(+ \mid e_i) \end{aligned} \tag{19}$$

and

$$D_{bad} = P(+ \mid m) - P(+ \mid f) - \sum_{i=1}^{k} (P(e_i \mid m) - P(e_i \mid f)) P^*(+ \mid e_i) \tag{20}$$

To make the classifiers free from bad discrimination, the method modifies the original labels of the training data. It achieves

$$P'(+ \mid e_i, f) = P'(+ \mid e_i, m) = P'^{*}(+ \mid e_i) \tag{21}$$

where P' denotes the probability in the modified data. It proposes two possible techniques called local massaging and local preferential sampling.

Algorithm 1: Local massaging

input : dataset $(\mathbf{X}, \mathbf{s}, \mathbf{e}, \mathbf{y})$
output: modified labels $\hat{\mathbf{y}}$

`PARTITION` $(\mathbf{X}, \mathbf{e})$ (Algorithm 3);
for *each partition* $X^{(i)}$ **do**
- learn a ranker $\mathcal{H}_i : X^{(i)} \to y^{(i)}$;
- rank males using $\mathcal{H}_i$;
- `relabel DELTA` (male) males that are the closest to the decision boundary from $+$ to $-$ (Algorithm 4);
- rank females using $\mathcal{H}_i$;
- `relabel DELTA` (female) females that are the closest to the decision boundary from $-$ to $+$

end

Algorithm 2: Local preferential sampling

input : dataset $(\mathbf{X}, \mathbf{s}, \mathbf{e}, \mathbf{y})$
output: resampled dataset (a list of instances)

`PARTITION` $(\mathbf{X}, \mathbf{e})$ (see Algorithm 3);
for *each partition* $X^{(i)}$ **do**
- learn a ranker $\mathcal{H}_i : X^{(i)} \to y^{(i)}$;
- `rank` males using $\mathcal{H}_i$;
- `delete` $\frac{1}{2}$`DELTA` (male) (see Algorithm 4) males $+$ that are the closest to the decision boundary;
- `duplicate` $\frac{1}{2}$`DELTA` (male) males $-$ that are the closest to the decision boundary;
- `rank` females using $\mathcal{H}_i$;
- `delete` $\frac{1}{2}$`DELTA` (female) females $-$ that are the closest to the decision boundary;
- `duplicate` $\frac{1}{2}$`DELTA` (female) females $+$ that are the closest to the decision boundary;

end

6 Information Theoretic Measures for Fairness-aware Feature selection

This method proposes the accuracy measure for a subset of features $X_S \subseteq X^n$, denoted by $v^{Acc}(X_S)$.

$$\begin{aligned} v^{Acc}(X_S) &= I(Y; X_S \mid \{A, X_{S^c}\}) \\ &= UI(Y; X_S \backslash \{A, X_{S^c}\}) + CI(Y; X_S, \{A, X_{S^c}\}) \end{aligned} \tag{22}$$

where $UI(T; R_1 \backslash R_2)$ denotes the unique information of R_1 with respect to T and $CI(T; R_1, R_2)$ denotes the information content that can be obtained only if both R_1 and R_2 are available.

For a subset of features $X_S \subseteq X^n$, the discrimination coefficient is defined as

$$v^D(X_S) = SI(Y; X_S, A) \times I(X_S; A) \times I(X_S; A \mid Y) \tag{23}$$

Given a characteristic function $v(\cdot) : \mathcal{P}([n]) \to \mathbb{R}$, the Shapley value function $\phi_{(\cdot)} : [n] \to \mathbb{R}$ is defined as:

$$\phi_i = \sum_{T \subseteq [n] \backslash i} \frac{|T|!(n - |T| - 1)!}{n!} (v(T \cup \{i\}) - v(T)),\ \forall i \in [n] \tag{24}$$

Given the characteristic functions $v^{Acc}(\cdot)$ and $v^D(\cdot)$, the corresponding Shapley value functions are denoted by $\phi^{Acc}_{(\cdot)}$ and $\phi^D_{(\cdot)}$. They are referred to as marginal accuracy coefficient and marginal discrimination coefficient. They can be used to define a score for each feature. Let $\mathcal{F}_i = \phi^{Acc}_i - \alpha \phi^D_i$ where α is a positive hyperparameter which trades off between accuracy and discrimination. $\mathcal{F}_i$ can be used for feature selection.