ISYE 8813: Algorithmic Foundations of Ethical ML **Fall 2021**

Lectures 17: Algorithmic Interventions for Statistical Fairness

Lecturer: Juba Ziani

In this lecture, we will see some of the techniques that have been used in the algorithmic fairness literature to guarantee statistical fairness. Note that the list is not exhaustive, and is only intended to give you a sense of what kind of interventions arise in the literature. There is still a lot of research and a lot of open questions in how to obtain fairness in a lot of settings of interest.

1 Pre-processing for demographic parity

One of the techniques we can use to obtain statistical fairness is pre-processing. I.e., we do not touch the machine learning algorithm itself, but act even before then: we try to modify the data so as to remove bias in this data. Here, we will look at one way to do so in the context of demographic parity. Remember the notations from last lecture: we have a set of features X, a sensitive attribute $A \in 0, 1$, and decision or a score R. We want to guarantee that R is statistically independent of A, or, in other words

$$\Pr[R = 1 | A = 0] = \Pr[R = 1 | A = 1].$$

We may also want just to achieve a relaxation of that definition, of the form:

$$\frac{\Pr[R = 1 | A = 1]}{\Pr[R = 1 | A = 0]} \geq \tau,$$

for some threshold $\tau \in [0, 1]$; the higher the threshold, the more fair and stringent the guarantee.

In our pre-processing step, we will try to lose any information in the data X on whether someone belongs to the protected subgroup defined by $A = 1$; then, our decisions will be statistically independent of each individual's group, which is sufficient to achieve demographic parity. This property on its own is easy to achieve: one can decide simply to not use any information at all about an agent when making a decision about him/her; but this is obviously comes at the cost of making our decisions trivial and useless. What we want to do is to lose information about the protected sub-group in the data *while retaining as much information and predictive power as possible.*

Formnally: instead of working with the individuals' feature vector X, we will carefully design some function $f(.)$ that maps the features spaces to a new space. Let $Z \triangleq f(X)$. We will then restrict our attention and perform any machine learning task and decision only using Z. The idea then is to try to design $f(.)$ such that $Z = f(X)$ is correlated with A as little as possible. Then, any algorithm that makes decisions based on Z only will approximately satisfy $R = g(Z) \perp A$ (for some randomized mapping g representing the machine learning model).

Here, we will look at the approach of [ZWS+13], which is one of the first to propose a pre-processing approach for demographic parity. There, they pick $Z \in \{1, \dots, K\}$. This corresponds to a clustering of all possible vectors into K different clusters, each cluster containing simlar feature vectors X on which we want to make similar decisions. For each cluster k, we associate a vector v_k (in the original space of features X): you can think of this v_k as the center of the cluster, which will be used for training the model. They then aim to design the clusters such that 3 properties hold:

1. Statistical independence between Z and A:

$$|\Pr[Z = k | A = 0] - \Pr[Z = k | A = 1]| \text{ is small.}$$

2. The new representation does not lose too much information. I.e., given any training sample x_n,

$$\left(x_n - \sum_{k=1}^{K} \Pr[Z = k | x_n] \, v_k \right)^2 \text{ is small}$$

Note that $\sum_{k=1}^{n} \Pr[Z = k | x_n] \, v_k$ is the expected vector to which x_n is associated, and we are just saying that we want this expected vector or representation to be close to x_n.

3. The predictions are still accurate. I.e., if y_n is the true label for agent i with features x_n, and we assign a label of w_k to cluster i, we want that

$$\left| y_n - \sum_{k} \Pr[Z = k | x_n] \right| \text{ is small.}$$

Simplifying notations, let $M_{nk} = \Pr[Z = k | x_n]$, $\hat{x}_n = \sum_k \Pr[Z = k | x_n] \, v_k$ the new representation for agent n, $\hat{y}_n = \sum_k \Pr[Z = k | x_n] \, w_k$ the predicted label for agent n, and $M_k^A = \frac{1}{|A|} \sum_{n \in A} M_{nk}$ the probability that $Z = k$ in group A. The authors then aim to optimize the following loss function to obtain the new representation Z and the associated labels w_k for $Z = k$:

$$L = \alpha \sum_{k=1}^{K} |M_k^1 - M_k^0| + \beta \sum_{n=1}^{N} (x_n - \hat{x}_n)^2 - \gamma \sum_{n=1}^{N} \left(y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right)$$

The first term corresponds to 1), the second term to 2), the third term to 3). The third-term is just standard logistic regression. α, β, γ are hyper-parameters that can be optimized over for better results.

The authors evaluate their results on three different datasets (German, Adult, and Health). German = german credit dataset with 1000 instances which classify bank account holders in credit classes Good or Bad; each person is described by 20 attributes. They use age as the sensitive attribute. Adult = income dataset with 45,222 instances, 14 features, and the target variable is whether someone's income is larger than 50k dollars. They use gender as the sensitive attribute. The Health dataset contains 147,473 patients, using 139 features; the target variable is the number of days a person spends in the hospital on a given year. The sensitive attribute is once again age. The authors split their data into a training and a validation set, and obtain the following results:

1. In Figure 1, the authors study how well one can predict the sensitive attribute from i) the raw data (in yellow-green) and ii) the representation they learned (in blue). In all three datasets, the figure shows that the learned representation contains less information about the sensitive attribute than the original data. The improvement is particularly impressive for the Adult and Health datasets. In fact, in the health dataset, the accuracy of predicting A from Z is close to 50 percent: i.e., it is not possible to predict the sensitive attribute from the new, fair representation better than random guessing! This means A and Z are almost perfectly statistically independent.

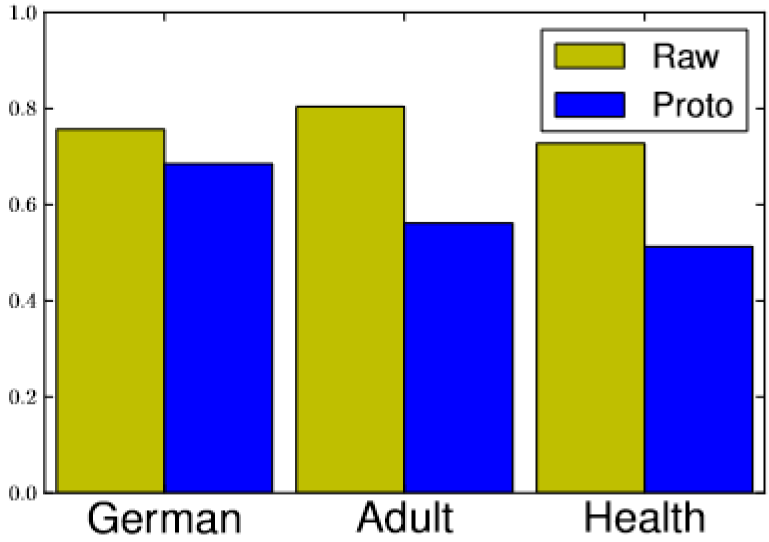

Figure 1: Accuracy of predicting A from X (in green) and Z (in blue)

2. In Figure 2, the authors look at the accuracy and the level of discrimination of their approach, compared to a baseline that do not take fairness into account, and two "naive" baselines for fairness (more information in the paper). They show their method leads to little discrimination, while retaining good accuracy.

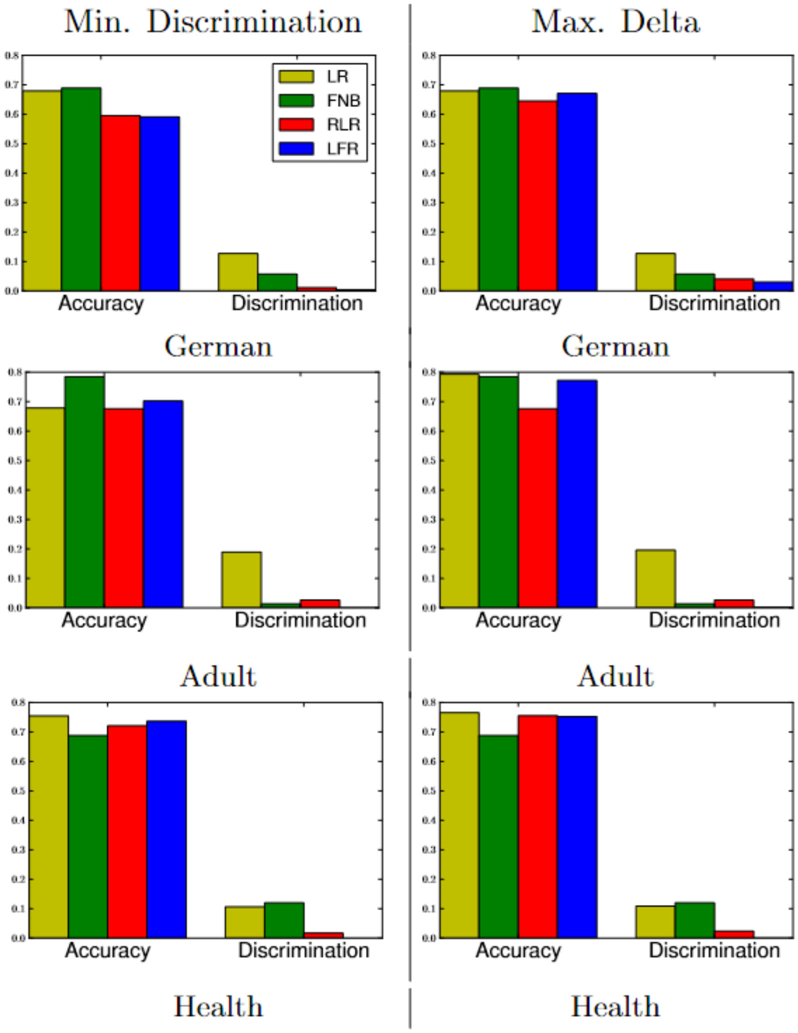

Figure 2: Accuracy and discrimination of the authors approach (in blue) compared to previous baselines

2 Including fairness constraints at training time

Here, a common approach is to explicitly add constraints that our classifier has to satisfy. Imagine the goal is to train a classifier that is parametrized by a vector parameter θ. Generally, the way this is done in many “simple” settings (linear or logistic regression, SVM, etc.) is that we want to find the parameters that minimize a convex (for tractability) loss function. For example, for a linear model and linear regression, we might want to minimize the mean squared error. I.e., given pairs of features and labels (x_i, y_i) for $n \in [N]$, we want to find the parameter $\hat{\theta}$ that minimizes loss function

$$L((x_i, y_i)_{n \in N}; \theta) = \sum_{n=1}^{N} \left(\theta^\top x_i - y_i\right)^2 .$$

To make sure the parameters satisfy a fairness constraint , one can try to explicitly write down such a constraint in the optimization program. Note that demographic parity means that the prediction $r_\theta(x_i) = \theta^\top x_i$ (in the linear regression setting) is independent of group membership A. One way to do so is to require that the empirical covariance matrix between r and A is close to 0. This is the approach taken for example in [ZVRG17]. The true covariance matrix can be written as

$$\begin{aligned}\mathbb{E}\left[(a - \mathbb{E}[a])\left(r_\theta(x_i) - \mathbb{E}\left[r_\theta(x_i)\right]\right)\right] &= \mathbb{E}\left[(a - \mathbb{E}[a]) r_\theta(x_i)\right] - \mathbb{E}\left[(a - \mathbb{E}[a])\right] \mathbb{E}\left[r_\theta(x_i)\right] \\ &= \mathbb{E}\left[(a - \mathbb{E}[a]) r_\theta(x_i)\right] .\end{aligned}$$

An empirical estimator of the covariance is then given by

$$\frac{1}{N} \sum_{n=1}^{N} (a_i - \bar{a}) r_\theta(x_i)$$

where $\bar{a}$ is the empirical mean of the a_i's. Note that with $r_\theta(x_i) = \theta^\top x_i$, we have that the above is linear in θ. We can then write approximate demographic parity as the empirical covariance between A and R being small, i.e. with convex constraints of the form

$$-\epsilon \leq \frac{1}{N} \sum_{n=1}^{N} (a_i - \bar{a}) r_\theta(x_i) \leq \epsilon.$$

Finding the optimal (empirical) classifier is then a convex optimization program in this setting, which makes it tractable.

Remark 1 (On covariances and independence)**.** *Note that two random variables X and Y that are independent must have 0 covariance. However, $Cov(X, Y)$ implies uncorrelation, but not independence in general (unless the variables are jointly Gaussian). In turn, the covariance constraints used in [ZVRG17] are a **relaxation** and a **proxy** for independence here. Approximate independence of R and A implies low covariance $Cov(R, A)$, but the space of feasible solutions of the above optimization problem is strictly bigger and involves solutions that are not independent across A and R.*

Remark 2 (On the convexity of the problem)**.** *Note however that for more general functions $r_\theta(x)$, it may not be the case that the constraints we added are convex. Often, a difficulty with more complex settings is that, while we can write the training procedure under fairness constraints as an optimization program, this program may not be convex or tractable (even if the loss function itself is convex!): fairness constraints can be non-tractable in the parameter θ we are trying to recover. A lot of the work for fairness at training time is to find tractable reformulations or relaxations of such hard problems.*

3 Equality of FPRs/FNRs via Post-Processing

This sub-section is based on the work of [HPS16]. Here, we want to look at post-processing. I.e., we already have a score or model $R = g(X)$ (g is our machine learning model, X are the features, and R is the score assigned by the model, as before), and we want to find a predictor $\tilde{R}$ that satisfies fairness, using only R, but not X. I.e., we want to post-process the results R of our model to guarantee fairness. One of the benefits of such an approach is that it works even if you only have black-box access to the model, and can only see its predictions R.

Here, we look at equality of FPRs and FNRs in a classification setting; i.e., we imagine that $R, \tilde{R}, Y$ are binary, and we want to guarantee

$$\Pr\left[\tilde{R} = 1 | A = 0, Y = 0\right] = \Pr\left[\tilde{R} = 1 | A = 1, Y = 0\right] \qquad \text{(equality of false positive rates)}$$
$$\Pr\left[\tilde{R} = 1 | A = 0, Y = 1\right] = \Pr\left[\tilde{R} = 1 | A = 1, Y = 1\right] \qquad \text{(equality of true positive rates)}$$

For simplicity of notations, let $\gamma_a(R') = \left(\Pr\left[R' = 1 | A = a, Y = 0\right], Pr\left[R' = 1 | A = a, Y = 1\right]\right)$. Here, note that we want $\gamma_0(\tilde{R}) = \gamma_1(\tilde{R})$, where $\tilde{R}$ is post-processed/derived from R. [ZVRG17] shows the following lemma, that makes it easier to search among the set of predictors $\tilde{R}$ that are derived from R and satisfy our fairness constraints:

Lemma 1. *A predictor $\tilde{R}$ is derived from R if and only if for all $a \in \{0, 1\}$, we have that $\gamma_a(\tilde{R}, R)$ is in the convex hull of*

$$P_a(R) = \left\{(0, 0), (1, 1), \gamma_a(R), \gamma_a(1 - R)\right\}.$$

Before proving this lemma, the reason this is nice is the following: being in the convex hull of a set of a few points can simply be expressed as a set of linear constraints over $\gamma_a(\tilde{R})$! We can now find the best derived predictor by searching over the values of $\gamma_0(\tilde{R}) = \gamma_1(\tilde{R})$ which belong in this convex hull $P_a(R)$. I.e., if we have a loss function $l(\tilde{R}, R)$ that tells us how good our predictor is, we can solve the following optimization problem to find the best derived predictor $\tilde{R}$ (best being the one that minimizes the loss) that is fair: it suffices to solve the following linear program:

$$\begin{aligned} \min_{\gamma_a(\tilde{R})} \quad & l(\tilde{R}, R) \\ \text{s.t.} \quad & \gamma_a(\tilde{R}) \in P_a(R) \\ & \gamma_0(\tilde{R}) = \gamma_1(\tilde{R}) \end{aligned}$$

Proof of Lemma 1. Fix a. Note that because we are in a binary setting with R and $\hat{R}$ in $\{0, 1\}$, we can fully define $\tilde{R}$ as a function of R using only the 2 variables of the form

$\Pr\left[\tilde{R}=1|R\in\{0,1\},A=a\right]$. We then have, via conditioning, that

$$\begin{aligned}\Pr\left[\tilde{R}=1|Y=0,A=a\right] &= \Pr\left[\tilde{R}=1|R=1,A=a\right]\Pr\left[R=1|Y=0,A=a\right]\\ &+\Pr\left[\tilde{R}=1|R=0,A=a\right]\Pr\left[R=0|Y=0,A=a\right]\\ &=\Pr\left[\tilde{R}=1|R=1,A=a\right]\Pr\left[R=1|Y=0,A=a\right]\\ &+\Pr\left[\tilde{R}=1|R=0,A=a\right]\Pr\left[1-R=1|Y=0,A=a\right].\end{aligned}$$

We have a similar expression for $Y=1$, which leads to

$$\begin{aligned}\gamma_a(\tilde{R}) &= \Pr\left[\tilde{R}=1|R=1,A=a\right]\gamma_a(R)\\ &+\Pr\left[\tilde{R}=1|R=0,A=a\right]\gamma_a(1-R)\\ &=p\gamma_a(R)+q\gamma_a(1-R).\end{aligned}$$

Note here that (p,q) can be any value in $[0,1]$, and this covers exactly all the possible ways of obtaining $\tilde{R}$ from R. This defines a polytope whose extreme points are reached when $p=0,q=0$, $p=0,q=1$, $p=1,q=0$, $p=1,q=1$; this is exactly the four points defining the convex hull. □

4 A Heuristic for Calibration via Post-Processing: Platt Scaling

Platt scaling is a heuristic to obtain calibration. The idea is to take an uncalibrated score R, and post-process this score via logistic regression. Formally, given an uncalibrated score R, Platt scaling aims to find a new score

$$S=\frac{1}{1+\exp(aR+b)}.$$

To do so, Platt scaling aims to find the best parameters a and b via logistic regression. I.e., we aim to find a and b that minimize the log-loss: i.e., if the target variable is Y, we aim to find the parameters (a,b) that minimize

$$-\mathbb{E}\left[Y\log S+(1-Y)\log(1-S)\right],$$

or in practice the empirical loss

$$-\sum_i y_i\log S_i-\sum_i(1-y_i)\log(1-S_i),$$

where (R_i,y_i) is the uncalibrated score and true label for agent i, and $S_i=\frac{1}{1+\exp(aR_i+b)}$.

The intuition behind Platt scaling is that logistic regression in general yields well-calibrated models. This is often observed empirically, as shown in Figure 3 from Chapter 2 of the [BHN17] book: there we can see that in each group (defined by gender), the deciles of the score closely match the actually probabilities of positive true labels/outcomes. We will also formally prove a weaker version of “logistic regression ⇒ calibration” in HW3.

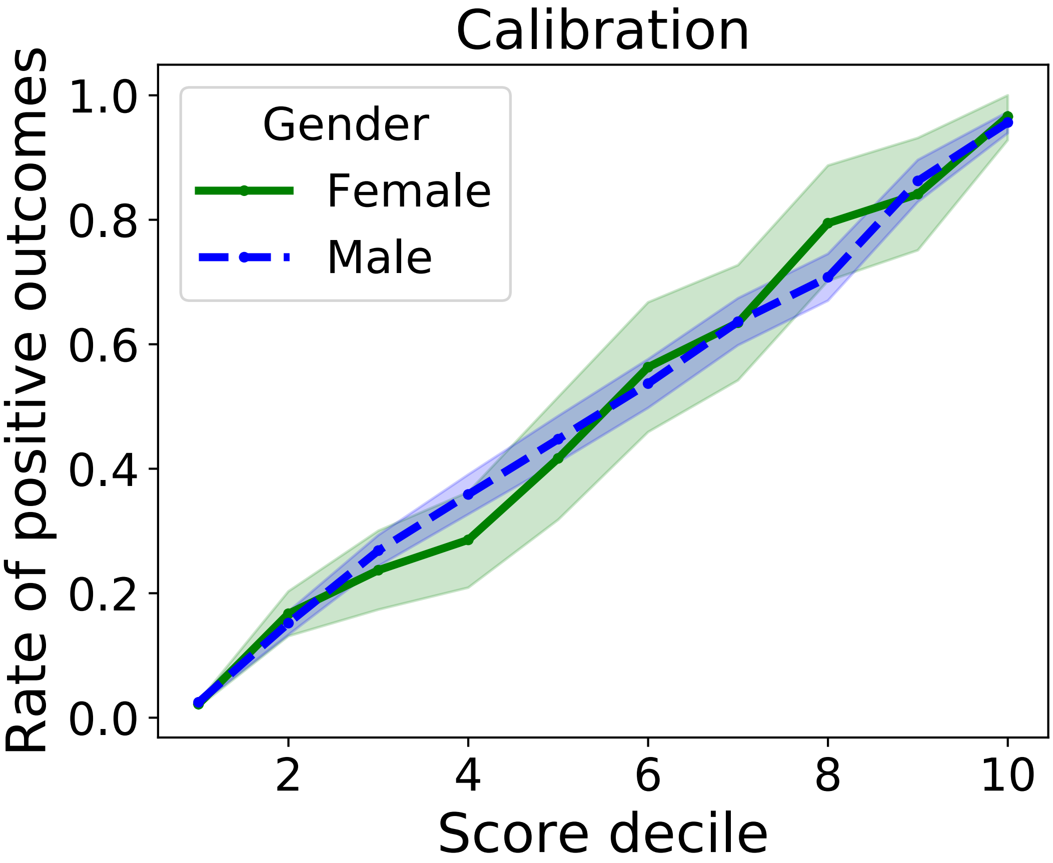

Figure 3: Results of Logistic Regression on the Adult Dataset. Calibration-by-group is obtained as a result of logistic regression, without intervention.

References

[BHN17] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *Nips tutorial*, 1:2017, 2017.

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[ZVRG17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[ZWS$^+$13] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.