August 3, 2021
Changho Suh (chsuh@kaist.ac.kr)

Lecture 2: A fair classifier using mutual information

Outline

This lecture investigates a fair classifier which is inspired by an interesting connection between fairness measures and mutual information (MI) [2]. Specifically what we are going to do are five folded. First we will introduce a problem setting together with associated notations. We will then introduce an optimization framework for a conventional classifier which forms the basis of a fair classifier to be explored. Next we will establish a connection between fairness measures and MI. Building upon the connection, we will then develop an MI-based optimization for a fair classifier. Finally we will translate it into an implementable optimization, thereby coming up with a concrete way to solve the optimization.

Problem setting

Fig. 1 illustrates the architecture of a conventional binary classifier. There are two types of

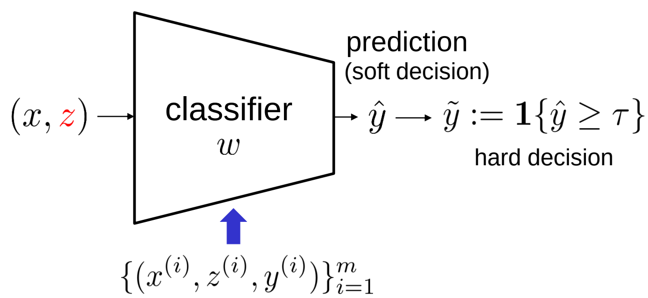

Figure 1: A problem setting of a binary fair classifier. Here $X \in \mathbb{R}^d$ denotes normal (possibly non-sensitive) data, $Z \in \mathcal{Z}$ indicates a sensitive attribute with arbitrary alphabet size, and Y is a binary label. Let $\hat{Y}$ be the prediction output that intends to learn the ground-truth conditional probability $\mathbb{P}(Y = 1|X = x, Z = z)$ and $\tilde{Y}$ be its hard-decision value $\tilde{Y} := \mathbf{1}\{\hat{Y} \geq \tau\}$ where τ is a certain threshold. Here the classifier is parameterized by w.

data for input: (i) normal (possibly non-sensitive) data; (ii) sensitive attributes. We denote the normal data by $X \in \mathbb{R}^d$. In the case of recidivism score prediction, such X may refer to a collection of the number of prior criminal records and a criminal type, e.g., misdemeanour or felony. For sensitive data, we employ a different notation, say Z. In the above example, Z may indicate a race type among black ($Z = 0$) and white ($Z = 1$). In general, the alphabet size of Z is arbitrary. For instance, there are many race types such as Black, White, Asian, Hispanic, to name a few. Also there could be multiple sensitive attributes like gender and religion. In order to reflect such practical scenarios, we consider $Z \in \mathcal{Z}$ with an arbitrary alphabet size that can represent a collection of possibly many sensitive attributes. Let $\hat{Y}$ be the classifier output which aims to represent the ground-truth conditional distribution $\mathbb{P}(y|x, z)$. Here $Y \in \mathcal{Y}$ denotes the ground-truth label. In the recidivism score prediction, $Y = 1$ means reoffending in the near future, say within two years ($Y = 0$ otherwise), while $\hat{Y}$ indicates the probability of such event being occurred. Let $\tilde{Y}$ be its hard-decision value $\tilde{Y} := \mathbf{1}\{\hat{Y} \geq \tau\}$ where τ is a certain threshold. Here the classifier is parameterized by w. We consider a supervised learning setup, so we are given m example triplets: $\{(x^{(i)}, z^{(i)}, y^{(i)})\}_{i=1}^m$.

For illustrative purpose, this tutorial focuses on the simple binary classification setting and one fairness measure DDP:

$$\mathsf{DDP} := \sum_{z\in\mathcal{Z}} |\mathbb{P}(\tilde{Y} = 1|Z = z) - \mathbb{P}(\tilde{Y} = 1)|. \tag{1}$$

Fairness-regularized optimization

A conventional classifier optimization often takes the following form:

$$\min \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) \tag{2}$$

where $\ell_{\mathsf{CE}}(y, \hat{y})$ indicates cross entropy loss:

$$\ell_{\mathsf{CE}}(y, \hat{y}) := -y \log \hat{y} - (1 - y) \log(1 - \hat{y}). \tag{3}$$

How to incorporate the fairness measure DDP? Notice that the smaller DDP, the more fair the situation is. Hence, one natural approach is to incorporate the DDP as a regularization term:

$$\min \frac{1-\lambda}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \lambda \cdot \mathsf{DDP} \tag{4}$$

where λ denotes a regularization factor that lies in between 0 and 1. One can interpret λ as a fairness tuning knob. Here a challenge arises in solving the regularized optimization (4). Recalling the definition (1) of DDP, we see that DDP is a complicated function of the optimization variable w. It turns out it is not that simple to express DDP in terms of w. One effort to address this challenge was made by Zafar et al. [1]. They introduce an easily-expressible *proxy* for the fairness measure. Specifically they employ a covariance function between $\hat{Y}$ and Z. However, this proxy serves only as a *weak* constraint because a small covariance does not necessarily imply the independence although the reverse always hold. In this tutorial, we will study another approach which introduces a different regularization term that can serve as a *strong* constraint for the independence.

Connection between DDP and mutual information

The approach is based on the popular information-theoretic measure: mutual information. To clearly see how it is relevant, let us make a concrete connection. The connection is made via the following observation:

$$\mathsf{DDP} = 0 : \tilde{Y} \perp Z \iff I(Z; \tilde{Y}) = 0. \tag{5}$$

This is because $I(Z; \tilde{Y}) = 0$ is the sufficient and necessary condition for the independence between Z and $\tilde{Y}$. The connection can also be made via the soft-decision prediction value $\hat{Y}$. Notice that

$$I(Z; \tilde{Y}) \leq I(Z; \tilde{Y}, \hat{Y}) = I(Z; \hat{Y}) \tag{6}$$

where the 1st inequality comes from the chain rule ($I(Z; \tilde{Y}, \hat{Y}) = I(Z; \tilde{Y}) + I(Z; \hat{Y}|\tilde{Y})$) and the non-negativity of mutual information; and the 2nd equality is due to the fact that $\tilde{Y}$ is a function of $\hat{Y}$ ($\tilde{Y} := \mathbf{1}\{\hat{Y} \geq \tau\}$) and hence $I(Z; \tilde{Y}|\hat{Y}) = 0$. This together with (5) yields:

$$\mathsf{DDP} = 0 : \tilde{Y} \perp Z \impliedby I(Z; \hat{Y}) = 0. \tag{7}$$

Note that $I(Z;\hat{Y})=0$ can serve as a *strong* constraint for the independence.

MI-based approach [2]

The connection (7) naturally motivates us to employ $\lambda \cdot I(Z;\hat{Y})$ as a regularization term in (4) instead of $\lambda \cdot \mathsf{DDP}$:

$$\min_w \frac{1-\lambda}{m}\sum_{i=1}^m \ell_{\mathsf{CE}}(y^{(i)},\hat{y}^{(i)}) + \lambda \cdot I(Z;\hat{Y}). \tag{8}$$

Now a question of interest is: How to express $I(Z;\hat{Y})$ in terms of the optimization variable w? It turns out there is an interesting way to do this. To figure out the way, let us massage $I(Z;\hat{Y})$ to arrive at the expression.

A careful look at mutual information

Starting with the definition of mutual information, we get:

$$\begin{aligned}
I(Z;\hat{Y}) &= H(Z) - H(Z|\hat{Y}) \\
&\overset{(a)}{=} H(Z) - (H(\hat{Y},Z) - H(\hat{Y})) \\
&\overset{(b)}{=} H(Z) - \mathbb{E}\left[\log \frac{1}{\mathbb{P}_{\hat{Y},Z}(\hat{Y},Z)}\right] + \mathbb{E}\left[\log \frac{1}{\mathbb{P}_{\hat{Y}}(\hat{Y})}\right] \\
&= H(Z) + \sum_{\hat{y},z} \mathbb{P}_{\hat{Y},Z}(\hat{y},z) \log \frac{\mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{\mathbb{P}_{\hat{Y}}(\hat{y})}
\end{aligned} \tag{9}$$

where (a) comes from the chain rule $(H(\hat{Y},Z) = H(\hat{Y}) + H(Z|\hat{Y}))$; and (b) is due to the definitions of entropy and joint entropy. Define the term placed in the last line marked in blue as:

$$D^*(\hat{y},z) := \frac{\mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{\mathbb{P}_{\hat{Y}}(\hat{y})}. \tag{10}$$

Due to the total probability law, $D^*(\hat{y},z)$ should respect the sum-up-to-one constraint w.r.t. z:

$$\sum_z D^*(\hat{y},z) = 1 \quad \forall \hat{y}. \tag{11}$$

Mutual information via function optimization

Instead of $D^*(\hat{y},z)$, one can think about another function, say $D(\hat{y},z)$, which respects only the sum-up-to-one constraint (11). It turns out $D^*(\hat{y},z)$ is the optimal choice among such $D(\hat{y},z)$ in a sense of maximizing:

$$\sum_{\hat{y},z} \mathbb{P}_{\hat{Y},Z}(\hat{y},z) \log D(\hat{y},z), \tag{12}$$

and this gives insights into expressing $I(Z;\hat{Y})$ in terms of w. To see this clearly, let me formally state that $D^*(\hat{y},z)$ is indeed the optimal choice via the following theorem.

Theorem: The mutual information $I(Z;\hat{Y})$, reflected in the last line of (9), can be represented as the following function optimization:

$$I(Z;\hat{Y}) = H(Z) + \max_{D(\hat{y},z):\sum_z D(\hat{y},z)=1} \sum_{\hat{y},z} \mathbb{P}_{\hat{Y},Z}(\hat{y},z) \log D(\hat{y},z). \tag{13}$$

The proof of this is simple. Notice that the optimization (13) is *convex* in $D(\cdot,\cdot)$, since the log function is concave and the convexity preserves under additivity. Hence, by checking the KKT condition (the optimality condition for convex optimization), one can prove that the optimal $D(\cdot,\cdot)$ indeed respects (10) and (11). Here is detail. Consider the Lagrange function:

$$\mathcal{L}(D(\hat{y},z),\nu(\hat{y})) = \sum_{\hat{y},z} \mathbb{P}_{\hat{Y},Z}(\hat{y},z)\log D(\hat{y},z) + \sum_{\hat{y}} \nu(\hat{y}) \left(1 - \sum_{z} D(\hat{y},z)\right) \tag{14}$$

where $\nu(\hat{y})$'s indicate Lagrange multipliers w.r.t. the equality constraints. Consider the KKT conditions

$$\left.\frac{d\mathcal{L}(D(\hat{y},z),\nu(\hat{y}))}{dD(\hat{y},z)}\right|_{D=D_{\sf opt},\nu=\nu_{\sf opt}} = \frac{\mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{D_{\sf opt}(\hat{y},z)} - \nu_{\sf opt}(\hat{y}) = 0; \tag{15}$$

$$\sum_{z} D_{\sf opt}(\hat{y},z) = 1. \tag{16}$$

So we get $D_{\sf opt}(\hat{y},z) = \frac{\mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{\nu_{\sf opt}(\hat{y})}$. Plugging this into (16), we obtain:

$$\sum_{z} D_{\sf opt}(\hat{y},z) = \frac{\sum_z \mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{\nu_{\sf opt}(\hat{y})} = 1, \tag{17}$$

which yields:

$$\nu_{\sf opt}(\hat{y}) = \sum_{z} \mathbb{P}_{\hat{Y},Z}(\hat{y},z) = \mathbb{P}_{\hat{Y}}(\hat{y}). \tag{18}$$

This together with (15) then gives:

$$D_{\sf opt}(\hat{y},z) = \frac{\mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{\nu_{\sf opt}(\hat{y})} = \frac{\mathbb{P}_{\hat{Y},Z}(\hat{y},z)}{\mathbb{P}_{\hat{Y}}(\hat{y})} = D^*(\hat{y},z). \tag{19}$$

This completes the proof of the theorem.

How to express $I(Z;\hat{Y})$ in terms of w?

Are we done with expressing $I(Z;\hat{Y})$ in terms of w? No. This is because $P_{\hat{Y},Z}(\hat{y},z)$ that appears in (13) is not available. To resolve this problem, we rely upon the empirical distribution instead:

$$\mathbb{Q}_{\hat{Y},Z}(\hat{y}^{(i)},z^{(i)}) = \frac{1}{m} \qquad \forall i \in \{1,\dots,m\}.$$

In practice, the empirical distribution is very likely to be uniform, since $\hat{y}^{(i)}$ is real-valued and hence the pair $(\hat{y}^{(i)},z^{(i)})$ is unique with high probability. Now by parametrizing the function $D(\cdot,\cdot)$ with another, say θ, we can approximate $I(Z;\hat{Y})$ as:

$$I(Z;\hat{Y}) \approx H(Z) + \max_{\theta:\sum_z D_\theta(\hat{y},z)=1} \sum_{i=1}^{m} \frac{1}{m} \log D_\theta(\hat{y}^{(i)},z^{(i)}). \tag{20}$$

From the above parameterization building upon the function optimization (13), we can now approximately express $I(Z;\hat{Y})$ in terms of w and θ.

Implementable optimization

Notice in (20) that $H(Z)$ is irrelevant to the introduced optimization variables (w, θ). Hence, the MI-based optimization (8) can be (approximately) translated into:

$$\min_{w} \max_{\theta: \sum_z D_\theta(\hat{y}, z)=1} \frac{1}{m} \left\{ \sum_{i=1}^{m} (1-\lambda) \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \lambda \sum_{i=1}^{m} \log D_\theta(\hat{y}^{(i)}, z^{(i)}) \right\}. \tag{21}$$

The objective function is a function of (w, θ) and hence it is implementable, for instance, via famous neural networks. Many of the neural-net-based optimizations can readily be solved via a family of gradient descent algorithms. But here we see "min max". Hence, we can apply a slight variant of gradient descent that people often call *alternating gradient descent*, in which given w, θ is updated via the inner optimization and then given the updated θ, w is newly updated via the outer optimization, and this process iterates until it converges.

The architecture of the MI-based optimization (21) is illustrated in Fig. 2. On top of a classifier,

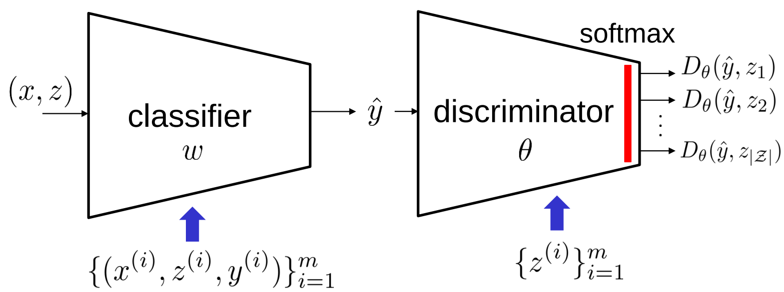

Figure 2: The architecture of the MI-based fair classifier (21). The prediction output $\hat{y}$ is fed into the discriminator wherein the goal is to figure out sensitive attribute z from $\hat{y}$. The discriminator output $D_\theta(\hat{y}, z)$ can be interpreted as the probability that $\hat{y}$ belongs to the attribute z. Here the softmax function is applied to ensure the sum-up-to-one constraint (11).

we introduce a new entity, called *discriminator*, which corresponds to the inner optimization. In discriminator, we wish to find θ^* that maximizes $\frac{1}{m}\sum_{i=1}^{m} \log D_\theta(\hat{y}^{(i)}, z^{(i)})$. On the other hand, the classifier wants to *minimize* such term. Hence, $D_\theta(\hat{y}, z)$ can be viewed as the ability to figure out z from prediction $\hat{y}$. Notice that the classifier wishes to minimize such ability for the purpose of fairness, while the discriminator has the opposite goal. So one natural interpretation that can be made on $D_\theta(\hat{y}, z)$ is that it captures the probability that z is indeed the ground-truth sensitive attribute for $\hat{y}$. Here the softmax function is applied to ensure the sum-up-to-one constraint (11).

Analogy with GAN [4]

Since the classifier and the discriminator are competing, one can make an analogy with a famous generative model: Generative Adversarial Networks (GANs), in which the generator and the discriminator also compete like a two-player game. While the fair classifier and GANs bear strong similarity in their nature, these two are distinct in their roles. See Fig. 3 for the detailed distinctions.

Extension to another fairness measure DEO

So far we have focused on one fairness measure DDP. One can also apply almost the same trick

MI-based fair classifier	**GAN**
discriminator Figure out sensitive attribute from prediction	discriminator **Goal:** Distinguish real samples from fake ones.
classifier Maximize prediction accuracy	generator Generate realistic fake samples

Figure 3: MI-based fair classifier vs. GAN. Both bear similarity in structure (as illustrated in Fig. 2), yet distinctions in role.

to another measure DEO:

$$\mathsf{DEO} := \sum_{y \in \mathcal{Y}} \sum_{z \in \mathcal{Z}} |\mathbb{P}(\tilde{Y} = 1 | Y = y, Z = z) - \mathbb{P}(\tilde{Y} = 1 | Y = y)|. \tag{22}$$

Specifically one can make a similar connection like:

$$\mathsf{DEO} = 0 : \tilde{Y} \perp Z | Y \quad \Longleftarrow \quad I(Z; \hat{Y} | Y) = 0. \tag{23}$$

This then leads to an implementable optimization:

$$\min_{w} \max_{\theta : \sum_z D_\theta(\hat{y}, z, y) = 1} \frac{1}{m} \left\{ \sum_{i=1}^{m} (1 - \lambda) \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \lambda \sum_{i=1}^{m} \log D_\theta(\hat{y}^{(i)}, z^{(i)}, y^{(i)}) \right\}. \tag{24}$$

Here the only distinction is that we read $D_\theta(\hat{y}, z, y)$ instead of $D_\theta(\hat{y}, z)$.

Experiments

We provide experimental results to demonstrate that the MI-based fair classifier offers a good fairness performance. For illustrative purpose, we focus on a single yet popular benchmark real data: COMPAS [5]. Also we consider only one baseline: a non-fair classifier which does not incorporate any fairness-regularized term. For a sensitive attribute, we consider a race type (white vs. black), so Z is binary. In COMPAS, X contains prior criminal records, e.g., felony or misdemeanour and Y denotes whether or not an associated individual reoffends within two years.

Fig. 4 exhibits accuracy-vs-DDP tradeoff performances for the non-fair and MI-based fair classifiers. Notice that the fair classifier yields a significant fairness performance (reflected in a small

	accuracy	**DDP**
non-fair classifier	68.29 ± 0.44	0.2263 ± 0.0087
MI-based *fair* classifier	67.07 ± 0.47	0.0997 ± 0.0426

Figure 4: Accuracy-vs-DDP tradeoff. The MI-based fair classifier improves DDP significantly with a marginal degradation of accuracy.

DDP) with a negligible performance degradation in prediction accuracy.

A challenge

While it offers a great tradeoff performance, it comes with a challenge. The challenge is that the min max structure in the MI-based optimization (21) may lead to *training instability*. The training instability problem indeed occurs. The problem is particularly significant when λ is around 1. See Fig. 5. Here each point represents a performance evaluated on a single seed in

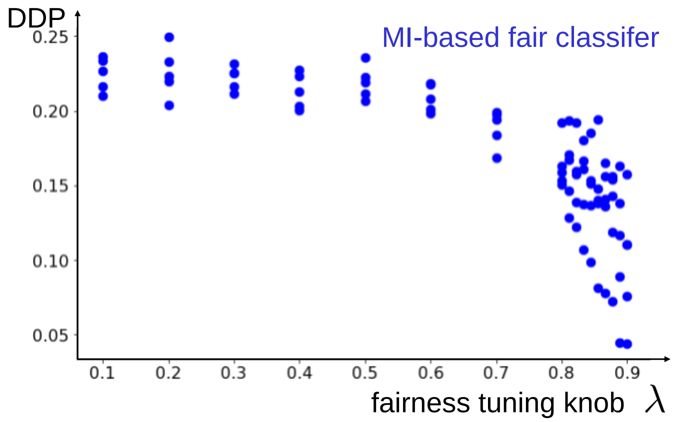

Figure 5: DDP as a function of the fairness tuning knob λ. Each blue dot corresponds to a single result w.r.t. one particular seed for training. The spreadness of the blue dots in particular near $\lambda \approx 1$ implies that the min max optimization framework (21) yields different results with distinct seeds, thereby incurring training instability.

training. We see different points spread over a wide range of DDP, implying an unstable training performance.

Look ahead

There has been a recent work [3] that addresses the training instability while offering a better tradeoff. It is based on a prominent statistical method often employed by information theorists: *kernel density estimation (KDE)*. Next lecture, we will explore the KDE-based fair classifier.

References

[1] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. *Artificial Intelligence and Statistics Conference (AISTATS)*, 2017.

[2] J. Cho, G. Hwang and C. Suh. A fair classifier using mutual information. *IEEE International Syposium on Inofrmation Theory (ISIT)*, 2020.

[3] J. Cho, G. Hwang and C. Suh. A fair classifier using kernel density estimation. *In Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems 27 (NeurIPS), 2014.

[5] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: There's software used across the country to 272 predict future criminals. And it's biased against blacks. https://www.propublica.org/article/machine-bias-risk-assessments-incriminal-sentencing, 2015.