Changho Suh (chsuh@kaist.ac.kr)

Lecture 4: A generative model: A hinge-loss-based GAN

Recap

Yesterday we explored fairness issues in the context of prominent machine learning models: classifiers. Here are what we did in detail. First we introduced two major fairness measures that we named DDP and DEO respectively. We then studied an MI-based fair classifier building upon an interesting connection made between the fairness measures and mutual information (MI). We observed that the MI-based optimization exhibits the min max structure, thus suffering from training instability. Hence, we investigated another fair classifier based on KDE, which addresses the training instability issue.

Outline of Lectures 4/5/6

Today we will move onto a different context that I promised to cover in Lecture 1: *Fair generative models*. Specifically what we are going to do are four folded. First we will study a very famous generative model which forms the basis of the fair generative model to be explored in depth: a hinge-loss-based GAN [3]. We will then make an inspiring connection with a prominent statistical measure prevalent in information theory: *Total Variation Distance (TVD)*. Building upon the connection, we will next develop a TVD-based fair generative model and then translate it into an implementable optimization which exhibits hinge losses. Finally we will discuss a couple of other relevant issues. In Lecture 4, we will cover the first two.

Generative Adversarial Networks [1]

GAN is a very famous generative model that is shown to generate quite realistic fake samples. It consists of two entities. Let us first focus on one entity, called *discriminator*. The discriminator is introduced in an attempt to discriminate real samples from generated fake samples. Let $D(\cdot)$ be the output of the discriminator. Inspired by its role, the discriminator is intended to yield the output such that $D(\cdot)$ well captures the capability to discriminate. To this end, it is designed to well *approximate* the probability of the input $(\cdot)$ being *real* data:

$$D(\cdot) \approx \mathbb{P}((\cdot) = \text{real}) \tag{1}$$

Notice that

$$\mathbb{P}(X = \text{real}) = 1; \quad \mathbb{P}(\tilde{X} = \text{real}) = 0 \tag{2}$$

where X and $\tilde{X}$ indicate real and generated fake samples, respectively. So we want to design $D(\cdot)$ such that

$$D(X) \text{ is as large as possible, close to 1;} \tag{3}$$

$$D(\tilde{X}) \text{ is as small as possible, close to 0.} \tag{4}$$

This design principle is well illustrated in Fig. 1.

How to quantify the ability to discriminate?

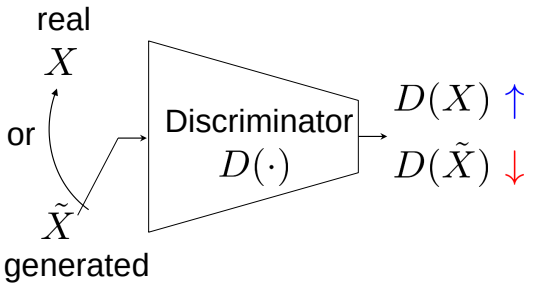

Figure 1: Discriminator wishes to output $D(\cdot)$ such that $D(X)$ is as large as possible while $D(\tilde{X})$ is as small as possible. Here the input takes either a real sample X or a generated fake sample $\tilde{X}$.

Keeping the picture in Fig. 1 in mind, one may want to quantify the ability to discriminate. To gain insights, observe that if $D(\cdot)$ can easily discriminate, then we should have:

$$D(X)\uparrow;\ \ 1-D(\tilde{X})\uparrow. \tag{5}$$

One naive way to capture the ability is simply adding the above two terms. But Goodfellow et al. [1] did not take the naive way. Instead they took the following *logarithmic* summation:

$$\max_D \mathbb{E}_{\mathbb{P}_{\text{real}}}[\log D(X)] + \mathbb{E}_{\mathbb{P}_G}\left[\log(1-D(\tilde{X}))\right] \tag{6}$$

where $\mathbb{P}_G$ denotes the distribution of a generated sample $\tilde{X}$. Actually I was wondering why they took this particular way, as the paper does not mention about the rationale behind the choice. In NeurIPS 2016, Goodfellow gave a tutorial on GANs, mentioning that the problem formulation was inspired by a paper published in AISTATS 2010 [2]. We will not touch upon the details herein. If you are interested, please take a look at the paper.

A two-player game

Goodfellow [1] then imagined a two-player game in which Player 1, Discriminator $D(\cdot)$, wishes to maximize the quantified ability (6), while Player 2, Generator $G(\cdot)$, wants to minimize (6):

$$\begin{aligned}&\text{Discriminator: } \max_D \mathbb{E}_{\mathbb{P}_{\text{real}}}[\log D(X)] + \mathbb{E}_{\mathbb{P}_G}\left[\log(1-D(\tilde{X}))\right];\\ &\text{Generator: } \quad \min_G \mathbb{E}_{\mathbb{P}_{\text{real}}}[\log D(X)] + \mathbb{E}_{\mathbb{P}_G}\left[\log(1-D(\tilde{X}))\right].\end{aligned} \tag{7}$$

Usually we employ a Gaussian random variable, say N, for the input of Generator. See Fig. 2 for the architecture of GAN. We see that the two players are competing with each other. Discriminator wishes to well distinguish real from fake samples, while Generator wants to fool Discriminator.

A variant of GAN: A hinge-loss-based GAN [3]

Since the birth of GAN [1], there has been a proliferation of GAN variants. In this tutorial, we explore one particular variant of GAN in depth, which can well be interpreted via a lens of information-theoretic tools as well as turns out to give insights into developing a fair generative model. That is, a hinge-loss-based GAN [3].

Here is how it looks. Lim et al. [3] employed different loss functions (instead of log loss): *hinge*

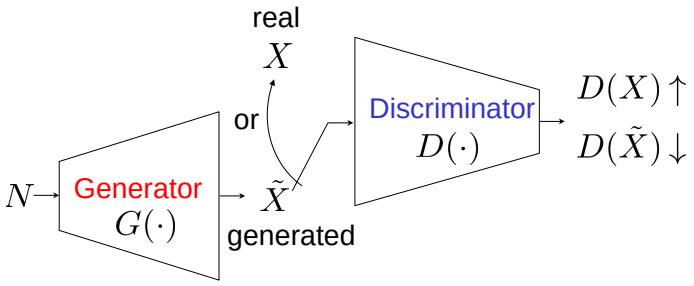

Figure 2: A two-player game: Discriminator wishes to distinguish real against fake samples, i.e., maximizes $D(X)$ while minimizing $D(\tilde{X})$; on the other hand, Generator (with a random input N) wishes to fool D, i.e., the goal is exactly opposite.

loss for Discriminator; and *linear* loss for Generator:

$$\begin{aligned} &\text{Discriminator: } \max_D \mathbb{E}_{\mathbb{P}_{\text{real}}}\left[\min(0, -1 + D(X))\right] + \mathbb{E}_{\mathbb{P}_G}\left[\min(0, -1 - D(\tilde{X}))\right]; \\ &\text{Generator: } \min_G \mathbb{E}_{\mathbb{P}_{\text{real}}}\left[D(X)\right] - \mathbb{E}_{\mathbb{P}_G}\left[D(\tilde{X})\right]. \end{aligned} \tag{8}$$

Obviously it looks like a heuristic choice that only bears strong similarity to the SVM formulation. Yet it seems to have nothing to do with any theoretical insights. Hence, some hardcore information theorists may not be interested in this seemingly-heuristic formulation. But later it has been shown that there is an interesting theoretical insight behind this choice. Specifically, it has an interesting connection with Total Variation Distance (TVD) and this turns out to give insights into developing a fair generative model. In addition, perhaps more importantly to practitioners, an advanced version [4] of the hinge-loss-based GAN offers the state-of-the-art performance in many of the real datasets such as CelebA, although there is no concrete theoretical analysis on why that is the case. Fig. 3 showcases some fake yet very high-quality realistic samples generated by the hinge-loss-based GAN.

Figure 3: Fake samples generated by an advanced version [4] of the hinge-loss-based GAN [3].

Connection with TVD [5]

Let us explore the connection with TVD in depth. The connection is formally stated below.

Theorem (Tan et al. [5]): Suppose we first find the optimal solution D^* for discriminator and then plug this into the generator's optimization to find G^*. Then, the optimal solution to

the optimization (8) of the hinge-loss-based GAN reads:

$$\begin{aligned} D^*(x) &= \mathsf{sign}(\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x)) \qquad \forall x \in \mathcal{X} \cup \tilde{\mathcal{X}} \\ G^*(\cdot) &= \arg\min_G \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} |\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x)| \end{aligned} \tag{9}$$

where $\mathsf{sign}(\cdot)$ indicates the sign function that returns $+1$ for a non-negative input; -1 otherwise. Here $\mathcal{X}$ and $\tilde{\mathcal{X}}$ denote the sets of the real and generated samples, respectively. Since $\mathbb{P}_{\mathsf{real}}(x)$ is undefined for $x \in \tilde{\mathcal{X}} \setminus \mathcal{X}$, we simply set it as $\mathbb{P}_{\mathsf{real}}(x) = 0$. Similarly for $x \in \mathcal{X} \setminus \tilde{\mathcal{X}}$, we set the undefined $\mathbb{P}_G(x)$ as 0. Notice that the blue term in (9) is a scaled version of the total variation distance [6]:

$$G^*(\cdot) = \arg\min_G \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} |\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x)| = \arg\min_G \mathsf{TV}(\mathbb{P}_{\mathsf{real}}, \mathbb{P}_G). \tag{10}$$

Proof of $D^*(x) = \mathsf{sign}(\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x))$

Let us prove the first in (9). The objective function in max optimization is written as:

$$\begin{aligned} &\mathbb{E}_{\mathbb{P}_{\mathsf{real}}}[\min(0, -1 + D(X))] + \mathbb{E}_{\mathbb{P}_G}\left[\min(0, -1 - D(\tilde{X}))\right] \\ &= \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} \mathbb{P}_{\mathsf{real}}(x) \min(0, -1 + D(x)) + \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} \mathbb{P}_G(x) \min(0, -1 - D(x)). \end{aligned} \tag{11}$$

Here we rely upon a lemma stated below:

Lemma (Lim et al. [3]): Let $f(x) = \alpha \cdot \min\{0, -1 + x\} + \beta \cdot \min\{0, -1 - x\}$. For $\alpha, \beta \geq 0$,

$$x^* = \arg\max_x f(x) = \mathsf{sign}(\alpha - \beta) \tag{12}$$

where the equality case $\alpha = \beta$ yields $x^* = +1$.

Note 1: One can readily prove this lemma by doing case studies: Case I ($\alpha \geq \beta$); Case II ($\alpha < \beta$). For each case, we split into three sub-cases: (i) $x > 1$; (ii) $-1 \leq x \leq 1$; (iii) $x < -1$. The proof is easy yet tedious. Hence, we omit the detailed proof herein. Please check it if you are not convinced.

Note 2: The maximizer x^* may not be unique. In such as case, we take x^* as one of the maximizers.

Let us apply this lemma to the 2nd line in (11). Notice that for each x, $\mathbb{P}_{\mathsf{real}}$ and $\mathbb{P}_G$ correspond to α and β, respectively. Hence, by using the lemma, we obtain the maximizer $D^*(x)$ as:

$$D^*(x) = \mathsf{sign}(\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x)). \tag{13}$$

For all of the other x's, we apply the same, thus completing the proof.

Proof of $G^*(\cdot) = \arg\min_G \mathsf{TV}(\mathbb{P}_{\mathsf{real}}, \mathbb{P}_G)$

Let us prove the second in (9). Plugging $D^*(x)$ into the objective function in Generator's min optimization, we get:

$$\begin{aligned} \mathbb{E}_{\mathbb{P}_{\mathsf{real}}}[D^*(X)] - \mathbb{E}_{\mathbb{P}_G}\left[D^*(\tilde{X})\right] &= \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} \mathbb{P}_{\mathsf{real}}(x) D^*(x) - \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} \mathbb{P}_G(x) D^*(x) \\ &= \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} (\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x)) D^*(x) \\ &= \sum_{x \in \mathcal{X} \cup \tilde{\mathcal{X}}} |\mathbb{P}_{\mathsf{real}}(x) - \mathbb{P}_G(x)| = 2 \cdot \mathsf{TV}(\mathbb{P}_{\mathsf{real}}, \mathbb{P}_G). \end{aligned} \tag{14}$$

This completes the proof.

Look ahead

Next time, we will investigate a TVD-based fair generative model and then translate it into an implementable optimization represented via hinge loss.

References

[1] 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.

[2] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *AISTATS* 2010.

[3] J. H. Lim and J. C. Ye. Geometric GAN. *arXiv:1705.02894*, 2017.

[4] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. *International Conference on Learning Representations*, 2019.

[5] Z. Tan, Y. Song, and Z. Ou. Calibrated adversarial algorithms for generative modelling. *Stat*, 2019.

[6] L. Tierney. Introduction to general state-space markov chain theory. *Markov chain Monte Carlo in practice*, 1996.