Adversarial Training against Systematic Uncertainty

Gilles Louppe,[1] **Michael Kagan**,[2] and **Kyle Cranmer**[1]

[1] *New York University*
[2] *SLAC National Accelerator Laboratory*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam commodo, enim vitae facilisis pretium, ligula justo aliquet lectus, non interdum erat lorem et nisi. Donec pharetra lectus in magna pellentesque vehicula. Praesent dapibus lorem sed enim lacinia, a vestibulum sapien mattis. Sed ornare mollis aliquet. Nulla tempor lacinia tortor, in rhoncus augue porta nec. Morbi sed convallis nibh, eu hendrerit turpis. Curabitur sit amet rhoncus purus. Curabitur eget magna lorem. Phasellus pretium nisi quis est tincidunt, faucibus vulputate augue viverra. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin a urna a ex egestas pulvinar.

I. INTRODUCTION

[GL: Distinction between statistical and systematic uncertainty.] [GL: Outline taxonomy of uncertainties.] [GL: Define nuisance parameters. See refs in Noel's papers.] [GL: We want to build an accurate classifier whose output remains invariant with respect to systematic uncertainties.] [GL: Motivate the criterion (which may not be obvious for the ML crowd). See pivotal quantity motivation.] [GL: Discuss what kind of uncertainties the proposed approach is good for.]

II. PROBLEM STATEMENT

Let assume a probability space $(\Omega, \mathcal{F}, P)$, where Ω is a sample space, $\mathcal{F}$ is a set of events and P is a probability measure. Let consider the multivariate random variables $X_z : \Omega \mapsto \mathbb{R}^p$ and $Y : \Omega \mapsto \mathcal{Y}$, where X_z denotes a dependence on a nuisance parameter Z whose values $z \in \mathcal{Z}$ define a parameterized family of its systematic uncertainties. That is, X_z and Y induce together a joint probability distribution $p(X, Y|z)$, where the conditional on z denotes X_z. For training, let further assume a finite set $\{x_i, y_i, z_i\}_{i=1}^N$ of realizations $X_{z_i}(\omega_i), Y(\omega_i)$, for $\omega_i \in \Omega$ and known values z_i of the nuisance parameter. Our goal is to learn a function $f(\cdot; \theta_f) : \mathbb{R}^p \mapsto \mathcal{Y}$ of parameters θ_f (e.g., a neural network-based classifier if $\mathcal{Y}$ is a finite set of classes) and minimizing a loss $\mathcal{L}_f(\theta_f)$ (e.g., the cross-entropy). In addition, we require that $f(X_z; \theta_f)$ should be robust to the value z of the nuisance parameter – which remains unknown at test time. More specifically, we aim at building f such that in the ideal case

$$f(X_z(\omega); \theta_f) = f(X_{z'}(\omega); \theta_f) \tag{1}$$

for all samples $\omega \in \Omega$ and all z, z' pairs of values of the nuisance parameter.

Since we do not have training tuples $(X_z(\omega), X_{z'}(\omega))$ (for the same unknown ω), we propose instead to solve the closely related problem of finding a predictive function f such that

$$\begin{aligned} &P(\{\omega | f(X_z(\omega); \theta_f) = y\}) \\ &\quad = P(\{\omega' | f(X_{z'}(\omega'); \theta_f) = y\}) \end{aligned} \tag{2}$$

for all $y \in \mathcal{Y}$. In words, we are looking for a predictive function f which is a pivotal quantity [1] with respect to the nuisance parameter. That is, such that the distribution of $f(X_z; \theta_f)$ is invariant with respect to the value z of the nuisance. Note that a function f for which Eqn. 1 is true necessarily satisfies Eqn. 2. In general, the converse is however not true, since the sets of samples $\{\omega | f(X_z(\omega); \theta_f) = y\}$ and $\{\omega' | f(X_{z'}(\omega'); \theta_f) = y\}$ do not need to be the same for the equality to hold. In order to simplify notations, and as only Eqn. 2 is of direct interest in this work, we denote from here on the pivotal quantity criterion as

$$p(f(X; \theta_f)|z) = p(f(X; \theta_f)|z') \tag{3}$$

for all $z, z' \in \mathcal{Z}$.

III. METHOD

Adversarial training was first proposed by [2] as a way to build a generative model capable of producing samples from random noise $z \sim p_Z$. More specifically, the authors pit a generative model $g : \mathbb{R} \mapsto \mathbb{R}^p$ against an adversary classifier $d : \mathbb{R}^p \mapsto \{0, 1\}$ whose antagonistic objective is to recognize real data X from generated data $g(Z)$. Both models g and d are trained simultaneously, in such a way that g learns to produce samples that are difficult to identify by d, while d incrementally adapts to changes in g. At the equilibrium, g models a distribution whose samples can be identified by d only by chance. That is, assuming enough capacity in d and g, the distribution $p_{g(Z)}$ eventually converges towards the real distribution p_X.

In this work, we repurpose adversarial training as a means to constraint the predictive model f in order to satisfy Eqn. 3. As illustrated in Figure 1, we pit f against an adversary model $r := p_{\theta_r}(z|f(X; \theta_f))$ of parameters θ_r and associated loss $\mathcal{L}_r(\theta_f, \theta_r)$. This model takes as input realizations of $f(X; \theta_f)$, for the current

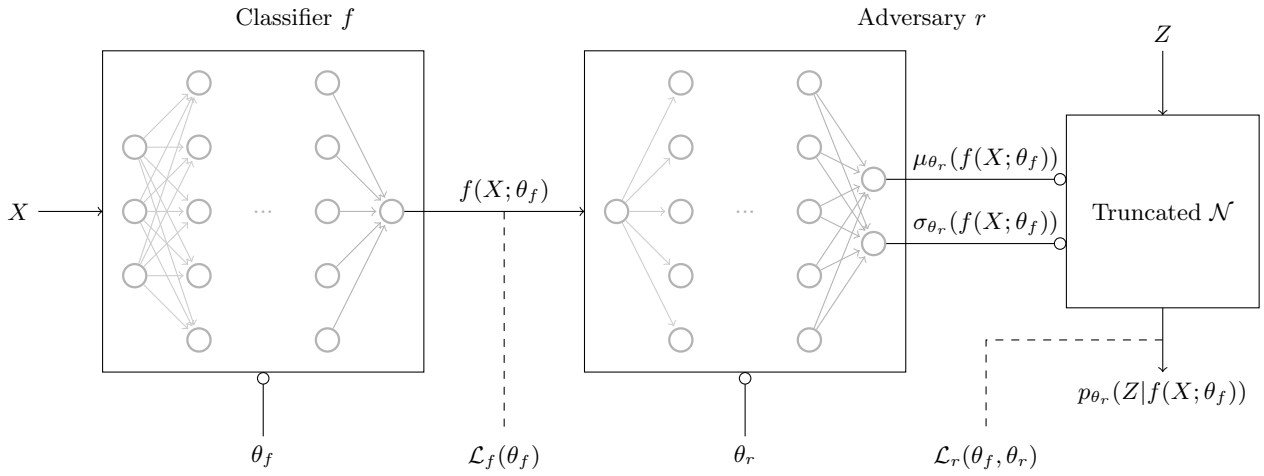

FIG. 1. Architecture for the adversarial training of a binary classifier f against a continuous parameter Z, assuming that $Z|f(X;\theta_f)$ follows a truncated gaussian distribution.

value θ_f of f parameters, and produces as output a function $p_{\theta_r}(z|f(X;\theta_f))$ modeling the posterior probability density that z parameterizes the sample X observed through $f(\cdot;\theta_f)$. Intuitively, if $p(f(X;\theta_f)|z)$ varies with z, then the corresponding correlation can be captured by r. By contrast, if $p(f(X;\theta_f)|z)$ is invariant with z, as we require, then r should perform poorly and be close to random guessing. Training f such that it additionally minimizes the performance of r therefore acts as a regularization towards Eqn. 3.

If Z takes discrete values, then p_{θ_r} can be represented e.g. as a probabilistic classifier $\mathbb{R} \mapsto \mathbb{R}^{|\mathcal{Z}|}$ whose output j (for $j = 1, \ldots, |\mathcal{Z}|$) is the estimated probability mass $p_{\theta_r}(z_j|f(X;\theta_f))$. Similarly, if Z takes continuous values and if we assume some parameteric distribution for $Z|f(X;\theta_f)$ (e.g., a truncated gaussian over a bounded support), then p_{θ_r} can be represented e.g. as network whose output j is the estimated value of the corresponding parameter of that distribution (e.g., its mean and variance). As in [3], the estimated probability density $p_{\theta_r}(z|f(X;\theta_f))$ can then be evaluated for any $z \in \mathcal{Z}$. As further explained in the next section, let us note that the adversary r may take any form, i.e. it does need to be a neural network, as long as it exposes a differentiable function $p_{\theta_r}(z|f(X;\theta_f))$ of sufficient capacity to represent the true distribution within its bounded support.

As for generative adversarial networks, we propose to train f and r simultaneously, which we carry out by considering the value function

$$E(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r) \tag{4}$$

that we optimize by finding the saddle point $(\hat{\theta}_f, \hat{\theta}_r)$ such that

$$\hat{\theta}_f = \arg\min_{\theta_f} E(\theta_f, \hat{\theta}_r), \tag{5}$$

$$\hat{\theta}_r = \arg\max_{\theta_r} E(\hat{\theta}_f, \theta_r). \tag{6}$$

Without loss of generality, the adversarial training procedure to obtain $(\hat{\theta}_f, \hat{\theta}_r)$ is formally presented in Algorithm 1 in the case of a binary classifier $f : \mathbb{R}^p \mapsto [0, 1]$ modeling $p(Y = 1|X)$. For reasons further explained in Section IV, $\mathcal{L}_f$ and $\mathcal{L}_r$ are respectively set to the expected value of the negative log-likelihood of $Y|X$ under f and of $Z|f(X;\theta_f)$ under r:

$$\mathcal{L}_f(\theta_f) = \mathbb{E}_{Y|X}[-\log p_{\theta_f}(Y|X)], \tag{7}$$

$$\mathcal{L}_r(\theta_f, \theta_r) = \mathbb{E}_{Z|f(X;\theta_f)}[-\log p_{\theta_r}(Z|f(X;\theta_f))]. \tag{8}$$

The optimization algorithm consists in using stochastic gradient descent alternatively for solving Eqn. 5 and 6.

IV. THEORETICAL RESULTS

In this section, we show that in the setting of Algorithm 1 where $\mathcal{L}_f$ and $\mathcal{L}_r$ are respectively set to expected value of the negative log-likelihood of $Y|X$ under f and of $Z|f(X;\theta_f)$ under r, the procedure converges to a classifier f which is a pivotal quantity in the sense of Eqn. 3.

In this setting, the nuisance parameter Z is considered as a random variable of bounded support, for which we require the uniform prior $p(z)$ (for $z \in \mathcal{Z}$). Importantly, classification of Y with respect to X is therefore considered in the context where Z is marginalized out, which means that the classifier minimizing $\mathcal{L}_f$ is optimal with respect to $Y|X$, but not necessarily with $Y|X, Z$. Results hold for a nuisance parameter Z taking either categorical values or continuous values within a bounded support. By abuse of notation, $H(p_Z)$ denotes the differential entropy in this latter case. Finally, propositions below are derived in a non-parametric setting, by assuming that both f and r have enough capacity.

Proposition 1. *Let $\hat{\theta}_r = \arg\max_{\theta_r} E(\theta_f, \theta_r)$ for fixed θ_f. If $p_{\theta_r}(z|f(X;\theta_f)) = p(z)$ for all $z \in \mathcal{Z}$, then f is a pivotal quantity.*

Algorithm 1 Adversarial training of a classifier f against an adversary r.

Inputs: training data $\{x_i, y_i, z_i\}_{i=1}^N$;
Outputs: $\hat{\theta}_f, \hat{\theta}_r$;
Hyper-parameters: Number T of training iterations, Number K of gradient steps to update r.

1: **for** $t = 1$ to T **do**
2: **for** $k = 1$ to K **do** ▷ Update r
3: Sample minibatch $\{x_m, z_m\}_{m=1}^M$ of size M;
4: With θ_f fixed, update r by ascending its stochastic gradient $\nabla_{\theta_r} E(\theta_f, \theta_r) :=$

$$\nabla_{\theta_r} \sum_{m=1}^{M} \log p_{\theta_r}(z_m | f(x_m; \theta_f));$$

5: **end for**
6: Sample minibatch $\{x_m, y_m, z_m\}_{m=1}^M$ of size M; ▷ Update f
7: With θ_r fixed, update f by descending its stochastic gradient $\nabla_{\theta_f} E(\theta_f, \theta_r) :=$

$$\nabla_{\theta_f} \sum_{m=1}^{M} \left[-\log p_{\theta_f}(y_m | x_m) + \log p_{\theta_r}(z_m | f(x_m; \theta_f)) \right],$$

where $p_{\theta_f}(y_m|x_m)$ denotes $1(y_m = 0)(1 - f(x_m; \theta_f)) + 1(y_m = 1) f(x_m; \theta_f))$;
8: **end for**

Proof. The optimal parameters

$$\hat{\theta}_r = \arg\max_{\theta_r} E(\theta_f, \theta_r) = \arg\min_{\theta_r} \mathcal{L}_r(\theta_f, \theta_r)$$

are such that $p_{\theta_r}(z|f(X;\theta_f)) = p(z|f(X;\theta_f))$. By assumption, $p_{\theta_r}(z|f(X;\theta_f)) = p(z)$, and therefore $p(z|f(X;\theta_f)) = p(z)$. Using the Bayes' rule, we write

$$\begin{aligned} p(f(X;\theta_f)|z) &= \frac{p(z|f(X;\theta_f))p(f(X;\theta_f))}{p(z)} \\ &= p(f(X;\theta_f)), \end{aligned}$$

which holds for all $z \in \mathcal{Z}$ and implies that f is a pivotal quantity. □

Proposition 2. *If there exists a saddle point $(\hat{\theta}_f, \hat{\theta}_r)$ for Eqn. 5 and 6 such that $E(\hat{\theta}_f, \hat{\theta}_r) = H(p_{Y|X}) - H(p_Z)$, then $f(\cdot; \hat{\theta}_f)$ is both an optimal classifier and a pivotal quantity.*

Proof. For fixed θ_f, the adversary r is optimal at $\hat{\theta}_r = \arg\max_{\theta_r} E(\theta_f, \theta_r) = \arg\min_{\theta_r} \mathcal{L}_r(\theta_f, \theta_r)$, in which case $p_{\theta_r}(z|f(X;\theta_f)) = p(z|f(X;\theta_f))$ and $\mathcal{L}_r$ reduces to the entropy $H(p_{Z|f(X;\theta_f)})$ of the conditional distribution of the nuisance. The value function E can therefore be rewritten as

$$E'(\theta_f) = \mathcal{L}_f(\theta_f) - H(p_{Z|f(X;\theta_f)}).$$

In particular, we have the lower bound

$$H(p_{Y|X}) - H(p_Z) \leq \mathcal{L}_f(\theta_f) - H(p_{Z|f(X;\theta_f)})$$

where the equality holds at $\hat{\theta}_f = \arg\min_{\theta_f} E'(\theta_f)$ only when

- $\hat{\theta}_f$ corresponds to the parameters of an optimal classifier, in which case the expected negative log-likelihood $\mathcal{L}_f$ of $Y|X$ reduces to its minimum value $H(p_{Y|X})$,
- all outcomes of $Z|f(X;\hat{\theta}_f)$ are equally likely, in which case $p(z|f(X;\hat{\theta}_f)) = p(z)$ for all $z \in \mathcal{Z}$ since we require a uniform prior by construction. Note that in the continuous case, the supremum of the differential entropy over continuous distributions on the same bounded support is also realized by the uniform distribution over that support.

Accordingly, the second condition implies that $p_{\theta_r}(z|f(X;\theta_f)) = p(z)$ and therefore that at this point, because of Proposition 1, the optimal classifier $f(\cdot;\hat{\theta}_f)$ is also a pivotal quantity.

□

Proposition 2 suggests that if at each step of Algorithm 1 the adversary r is allowed to reach its optimum given f (e.g., by setting K sufficiently high) and if f is updated to improve $\mathcal{L}_f(\theta_f) - H(p_{Z|f(X;\theta_f)})$, then f should converge to a classifier which is both optimal and pivotal, provided such a classifier exists. On many problems of interest though, such a classifier may not exist because the nuisance parameter directly shapes the decision boundary, in which cases the lower bound $H(p_{Y|X}) - H(p_Z) < \mathcal{L}_f(\theta_f) - H(p_{Z|f(X;\theta_f)})$ is strict: f can either be an optimal classifier or a pivotal quantity, but not both simultaneously. In this situation, it is natural to rewrite the value function E as

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r), \tag{9}$$

where $\lambda \geq 0$ is a hyper-parameter controlling the trade-off between the performance of f and its independence with respect to the nuisance parameter. Setting λ to a large value will preferably enforces f to be pivotal while setting λ close to 0 will rather constraint f to be optimal.

Interestingly, let us emphasize that these results hold using only the (1D) output of f (in the case of binary classification) as input to the adversary. We could similarly enforce an intermediate representation of the data to be pivotal, e.g. as in [4], but this is in fact not necessary. [GL: This also suggests we could use for r a simpler, maybe closed form, representation than a neural network (e.g. a GMM?).]

V. EXPERIMENTS

A. Toy example

[GL: Define the architecture of r when Z is categorical (r is a standard classifier) or continuous (r is e.g a 2-output NN such that the distribution of $Z|f(X)$ is modeled by a truncated Gaussian of known support). Cite Nix.]

B. Physics example

VI. RELATED WORK

To account for systematic uncertainties, experimentalists in high energy physics typically take as fixed a classifier f built from training data for a nominal value z_0 of the nuisance parameter, and then propagate uncertainty [GL: add ref] by estimating $p(f(x)|z)$ with a parameterized calibration procedure. Clearly, this classifier is however not optimal for $z \neq z_0$. In this setting, parameterized classifiers [5, 6] directly take (nuisance) parameters as additional input variables, hence ultimately providing the most statistically powerful approach for incorporating the effect of systematics on the underlying classification task. As argued in [7], such classifiers can however not be used on real data since the correct value z of the nuisance often remains unknown. This is typically not an issue in the context of parameter inference [5], where nuisance parameters are marginalized out, but otherwise often limits the range of their applications. In practice, parameterized classifiers are also computationally expensive to build and evaluate. In particular, calibrating their decision function, i.e. approximating $p(f(x,z)|z)$ as a continuous function of z, remains an open challenge. By contrast, constraining f to be pivotal yields a classifier which may not be optimal with respect to $Y|X,Z$, as discussed in Section IV, but that can otherwise be used in a wider range of applications, since knowing the correct value z of the nuisance is not necessary. Similarly, calibration needs to be carried out only once, since the dependence on the nuisance is now built-in. [GL: Shall we also discuss the relation with [7] where point estimates of the nuisance are used as inputs to f?]

In machine learning, learning a pivotal quantity can be related to the problem of domain adaptation [4, 8–12], where the goal is often stated as trying to learn a domain-invariant representation of the data. Likewise, our method also relates to the problem of enforcing fairness in classification [13, 14], which is stated as learning a classifier that is independent of some chosen attribute such as gender, color or age. For both families of methods, the problem can equivalently be stated as learning a classifier which is a pivotal quantity with respect to either the domain or the selected feature. In this context, [4, 14] are certainly among the closest to our work, in which domain invariance and fairness are enforced through an adversarial minimax setup composed of a classifier and an adversary discriminator. Following this line of work, our method can be regarded as a generalization that also supports the continuous case, which can be viewed as handling infinitely many domains, provided they can be continuously parameterized, or as enforcing fairness over continuous attributes.

[GL: Check related work of cited references to see if important related work are missing.]

VII. CONCLUSIONS

ACKNOWLEDGMENTS

[1] M. H. Degroot and M. J. Schervish, *Probability and statistics*, 4th ed. (2010).

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Advances in Neural Information Processing Systems* (2014) pp. 2672–2680.

[3] D. A. Nix and A. S. Weigend, in *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, Vol. 1 (IEEE, 1994) pp. 55–60.

[4] Y. Ganin and V. Lempitsky, ArXiv e-prints (2014), arXiv:1409.7495 [stat.ML].

[5] K. Cranmer, J. Pavez, and G. Louppe, (2015), arXiv:1506.02169.

[6] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, arXiv preprint arXiv:1601.07913 (2016), arXiv:1601.07913 [hep-ex].

[7] R. M. Neal, in *Proceedings of PhyStat2007, CERN-2008-001* (2007) pp. 111–118.

[8] J. Blitzer, R. McDonald, and F. Pereira, in *Proceedings*

of the 2006 conference on empirical methods in natural language processing (Association for Computational Linguistics, 2006) pp. 120–128.

[9] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, Neural Networks, IEEE Transactions on **22**, 199 (2011).

[10] R. Gopalan, R. Li, and R. Chellappa, in *Computer Vision (ICCV), 2011 IEEE International Conference on* (IEEE, 2011) pp. 999–1006.

[11] B. Gong, K. Grauman, and F. Sha, in *Proceedings of The 30th International Conference on Machine Learning* (2013) pp. 222–230.

[12] M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann, in *Proceedings of the IEEE International Conference on Computer Vision* (2013) pp. 769–776.

[13] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, .

[14] H. Edwards and A. J. Storkey, (2015), arXiv:1511.05897.