Adversarial Training against Systematic Uncertainty

Gilles Louppe
New York University
g.louppe@nyu.edu

Michael Kagan
SLAC National Accelerator Laboratory
makagan@slac.stanford.edu

Kyle Cranmer
New York University
kyle.cranmer@nyu.edu

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit...

1   Introduction

[GL: Distinction between statistical and systematic uncertainty.] [GL: Define nuisance parameters.] [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.]

2   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

$$P(\{\omega|f(X_z(\omega); \theta_f) = y\}) = P(\{\omega'|f(X_{z'}(\omega'); \theta_f) = y\}) \text{ for all } y \in \mathcal{Y}. \tag{2}$$

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

$$p(f(X; \theta_f)|z) = p(f(X; \theta_f)|z') \text{ for all } z, z' \in \mathcal{Z}. \tag{3}$$

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**

---

3 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. In particular, 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 value θ_f of f parameters, and produces as output probability estimates $p_{\theta_r}(z | f(X; \theta_f))$ that $f(X; \theta_f)$ is generated from the nuisance value z. 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.

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 4, $\mathcal{L}_f$ (resp. $\mathcal{L}_r$) is set to the expected value of the negative log-likelihood of $Y | X$ under f (resp. of $Z | f(X; \theta_f)$ under r). The optimization algorithm consists in using stochastic gradient descent alternatively for solving Eqn. 5 and 6.

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

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

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

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

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

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
$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r), \tag{7}$$
where $\lambda \ge 0$ 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 preferrably enforces f to be pivotal while setting λ close to 0 will rather constraint f to be optimal.

5 Experiments

[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).]

6 Related work

[GL: Alternative to learning a parameterized classifier for LLR.]

[GL: Similar to domain adaptation, but with infinitely many domains, as parameterized by Z, also related to transfer learning.]

[GL: Other applications: removing implicit bias in data (e.g. gender bias).]

7 Conclusions

Acknowledgments

References

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

[2] 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*, pp. 2672–2680, 2014.