@KyleCranmer New York University

Department of Physics Center for Data Science CILVR Lab

Center for Cosmology and particle physics

THREE FOUR APPROACHES TO SYSTEMATICS WITH MACHINE LEARNING

Introduction

There is a lot of activity in utilizing machine learning in the analysis of particle physics data… duh.

- I'll primarily focus on classification and regression tasks
- Supervised learning: function $f(x)$ to predict target y based on data x
- Classification: particle identification, signal vs. background discrimination
- Regression: estimate a particle's energy or momentum given detector readout

functions $f(h) = f(h(x))$

•Classically physicists design some useful features / "observables" / summary statistics $h(x)$ motivated by domain knowledge and then design relatively simple

• e.g. for classification "cuts" (a decision tree designed by hand)

2

incorporate systematic uncertainties.

•Two notions of "incorporate":

- Don't be wrong: view analysis chain as fixed and propagate systematic uncertainty through it.
	- e.g. control rate of type-I error in the presence of nuisance parameters
- Try to be "optimal": adjust the training of ML components so that the analysis is sensitive after accounting for systematics
	- e.g. minimize rate of type-II error / maximize power

We want to take advantage of the power of machine learning, but we need to

Humans, heuristics, hubris, and humility

•While ML can usually beat humans in classification tasks when neglecting systematic uncertainty, physicists are very good at designing features and "cuts" that are robust to systematic uncertainty.

summary statistics that are robust or insensitive to the underlying sources of uncertainty

- Physicists leverage various heuristics along the way to design "observables" / features /
	- e.g. if you know some variable is poorly modeled, don't use it
	- combination.

In the context of machine learning, we need to **formalize** what we mean by "incorporate systematic uncertainty" into an objective so that we can **operationalize** this

• e.g. if you know uncertainty leads to some (anti-)correlation, form the appropriate linear

- e.g. a modified objective for optimization
- This is not so easy.
- Also, physicists usually are thinking about multiple downstream use cases

Formalism & notation

In classic supervised learning, the training data is $\{x_i, y_i\}_{i=1,...,N}$

- e.g. binary classification where $y = 0$ for background, and $y = 1$ for signal
- For background: $y = 0 : x \sim p_{bkg}(x)$ or $p_{bkg}(x) = p(x | y = 0)$
- For signal: $y = 1 : x \sim p_{\text{sig}}(x)$ or $p_{\text{sig}}(x) = p(x | y = 1)$

 $\text{Training data drawn from a joint distribution: } (x_i, y_i) \sim p(x, y) = p(x \mid y)p(y)$

- In many physics problems, physicists don't know the (prior) distribution $p(y)$ in data
	- often $p(y = 1)$ is the quantity of interest (is there a signal present or not)
- Physicists often try to generate balanced training data, even though in the real data the classes are typically very imbalanced.
- \bullet it can be useful to call $p(y)$ a "proposal" instead of a "prior" to avoid confusion with down-stream Bayesian inference and to remember physicists typically don't take prediction literally as $p(y|x)$

$$
g(x) = p(x | y = 0)
$$

$$
p(x | y = 1)
$$

Formalism & notation

The primary notion of systematic uncertainty that physicists worry about is the lack of knowledge of the distribution $p(x|y)$ and how that uncertainty influences downstream inference

- This gives us a generative model *p*(*x* | *y*, *ν*)
- Lukas Heinrich's talk. Frequentist: *p*(*a*|*ν*), Bayesian: *p*(*ν*|*a*) ∝ *p*(*a*|*ν*)*p*(*ν*)

• e.g. training data often generated from a simulator $p_{\text{sim}}(x \, | \, y)$ and that simulator isn't perfect. Or training data comes from a control region in the data $p_{\text{control}}(x)$ that is assumed to be a good proxy for $p(x|y)$, but may not match for the region of the data

being analyzed.

•Typically we list what kinds of things might go wrong and parameterize their effect with nuisance parameters *ν*

• Uncertainty on the nuisance parameters is factorized from the effect they have, see

Formalism & notation

target y and systematic variation parameterized by ν

- e.g. $y =$ "signal events" and $\nu =$ "jet reconstruction efficiency is off by 7%"
- e.g. $y = "20$ GeV electron" and $\nu = "energy$ calibration off by 5%"

Typically we have some best estimate or nominal settings ν_0

- Typical ML training is based on $(x_i, y_i) \sim p(x | y, \nu = \nu_0)p(y)$
- This leads to a trained model *f*(*x*)
- And then we think about the distribution of the output
	- For the nominal $p(f|y, \nu = \nu_0)$ and systematic variations $p(f|y, \nu)$

Now we have a generative model $p(x|y, \nu)$ that can generate data x for each

An example from the archives

Fig. 2. (a) The output η of the neural network b tag for radiative returns to the Z for 161 GeV q \bar{q} Monte Carlo (histogram) compared to the data at 161 GeV (points). The shaded region shows the contribution from generated b-jets. (b) The performance of the neural network b tag (solid line) for Monte Carlo events, presented in terms of the efficiency for identifying b-jets versus the efficiency for rejecting light quark jets. The performance of the single most powerful b tagging input variable to the neural network is shown for comparison (dashed curve). $R_{\rm eff}$ and $R_{\rm eff}$ and $R_{\rm eff}$ d, $R_{\rm eff}$ and $R_{\rm eff}$ $\mathfrak{m}\varepsilon$ \mathfrak{L} for Tot GeV qq monte Carlo (histogram \mathcal{L} disserts e, H. Drevermann e, R.W. Forty e, M. Frank e, R.W. Forty e, M. Frank e, R. Hagelberg e, R. m erated ν -jets. (b) I. He performance of the ne I_n department of L_n define examples at μ_n and \mathcal{L}^n J.-F. Pusztaszeri e,4, F. Ranjard e, G. Rizzo e, L. Rolandi e, D. Rousseau e, \mathcal{C} to the indical network is shown for comparisor

- train on samples with $\nu = \nu_0$ to obtain fixed classifier $f(x)$
- uncertainty in ν modifies location and width of peak
- the classifier not optimal for $\nu \neq \nu_0$, but we can propagate uncertainty

$$
(x|v)
$$

- train on samples with $\nu = \nu_0$ to obtain fixed classifier $f(x)$
- uncertainty in ν modifies location and width of peak
- the classifier not optimal for $\nu \neq \nu_0$, but we can propagate uncertainty

$$
(x|v)
$$

- train on samples with $\nu = \nu_0$ to obtain fixed classifier $f(x)$
- uncertainty in ν modifies location and width of peak
- the classifier not optimal for $\nu \neq \nu_0$, but we can propagate uncertainty

$$
(x|v)
$$

An example from the archives

Here is an example comparing the nominal efficiency (calculated from $p(f | y, \nu = \nu_0)$) to a variational sample $p(f | y, v)$ taken from data.

Fig. 3. Comparison of the neural network b tag efficiency for a single jet in Z peak data and Monte Carlo, after the smearing correction, as a function of the cut on the jet neural network output: (a) absolute tag efficiency; (b) difference between data and Monte Carlo.

 $\overline{1}$

$$
\epsilon_{\mathbf{b}-\mathbf{jet}}(c,\nu) = \int_0^c p(f|y=1,\nu) \mathrm{d}f
$$

$$
\frac{\epsilon_{b-jet}(c,\nu) - \epsilon_{b-jet}(c,\nu = \nu_0)}{c \cdot \mu_0}
$$

$$
\epsilon_{b-jet}(c,\nu=\nu_9)
$$

Propagating uncertainty to downstream inference

After characterizing how the efficiency depends on the "working point" c and the nuisance parameters ν the experiments use this information in the downstream statistical analysis.

For example: $\epsilon_{\text{sig}}(\nu) =$ *c* 0 $p(f|y = 1, \nu) df$ $\epsilon_{bkg}(\nu) =$

And later one might form a statistical model for the number of events *n* that have $f(x) > c$, where s, b are some nominal number of signal and background events that would be produced, μ is the parameter of interest, ν is the nuisance parameter, and $p(a|\nu)$ is the likelihood associated to some auxiliary measurement used to estimate *ν* $p(n, a | \mu, \nu) = \text{Pois}(n | \mu \epsilon_{\text{sig}}(\nu)s + \epsilon_{\text{bkg}}(\nu)b)p(a | \nu)$

11

$$
\epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y=0,\nu) \, \text{d}f
$$

A visual example

•…and later one might form a statistical model for the number of events that *n* have $f(x) > c$

12

$$
\epsilon_{\text{sig}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df \qquad \epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df
$$

 $p(n, a | \mu, \nu) = \text{Pois}(n | \mu \epsilon_{sig}(\nu)s + \epsilon_{bkg}(\nu)b)p(a | \nu)$

$$
\epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y=0,\nu) \, \text{d}f
$$

A visual example

•…and later one might form a statistical model for the number of events that *n* have $f(x) > c$

12

$$
\epsilon_{\text{sig}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df \qquad \epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df
$$

 $p(n, a | \mu, \nu) = \text{Pois}(n | \mu \epsilon_{sig}(\nu)s + \epsilon_{bkg}(\nu)b)p(a | \nu)$

$$
\epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y = 0, \nu) \, \text{d}f
$$

A visual example

•…and later one might form a statistical model for the number of events that *n* have $f(x) > c$

12

$$
\epsilon_{\text{sig}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df \qquad \epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df
$$

 $p(n, a | \mu, \nu) = \text{Pois}(n | \mu \epsilon_{sig}(\nu)s + \epsilon_{bkg}(\nu)b)p(a | \nu)$

$$
\epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y = 0, \nu) \, \text{d}f
$$

inference on parameter of interest μ is not wrong... (e.g. coverage)

Totally factorized from training of the ML model $f(x)$

• Therefore no reason to think that $f(x)$ is optimal from the point of view of power or sensitivity on μ even if $f(x)$ was optimal for the supervised learning task with data generated from the nominal scenario ν_0

Ok, this is all standard stuff and background. Where do we go from here?

- The propagation of uncertainty approach is meant to ensure that downstream
	- $p(n, a | \mu, \nu) = \text{Pois}(n | \mu \epsilon_{\text{sig}}(\nu)s + \epsilon_{\text{bkg}}(\nu)b)p(a | \nu)$
		-

13

Where do we go from here?

New PDG Chapter on ML $N \sim N$. PDG Chanter on MI 41.6.4 Automatic Dierentiation and Back propagation 50

11. Mochina Laarning $11.6.1111110.1200111110.1200111110$ **41. Machine Learning**

Revised August 2019 by K Cranmer (NVII) II Seljak (IIC Berkeley: LBNI) and K Terao (SLAC) 41.6.10 Transfer learning: pre-training and fine-tuning 53 Revised August 2019 by K. Cranmer (NYU), U. Seljak (UC Berkeley; LBNL) and K. Terao (SLAC).

propagation of errors: one works with a model $f(x)$ and simply characterizes how uncertainty in the data distribution propagate through the function to the down-stream task irrespective of how it was trained.

multiple domains by sampling from some distribution over ν .

- **data augmentation:** one trains a model $f(x)$ in the usual way using training data from
- domain adaptation: one incorporates knowledge of the distribution for domains (or the parameterized family of distributions $p(x|y, \nu)$ into the training procedure so that the performance of $f(x)$ for the down-stream task is robust or insensitive to the uncertainty in ν .
- **parameterized models:** instead of learning a single function of the data $f(x)$, one learns a family of functions $f(x;\nu)$ that is explicitly parameterized in terms of nuisance parameters and then accounts for the dependence on the nuisance parameters in the down-stream

task.

16

• Note: this requires a prior / proposal distribution *p*(*ν*) this requires a prior / proposal distribution $p(\nu)$

17

Data augmentation gmentation shift, is simply the training data so that it is not that it includes examples corresponding to an i

•An intuitive approach to incorporate systematics into training is to train on " in mearing approach to mediporate systematics into training is to train on ²²⁵³ ing to several values of the nuisance parameter or systematic variations. As before one can

$$
x_i, y_i \sim p(x, y)
$$

²²⁵⁵ simply discards this information. This corresponds to sampling from the marginal distribution

 $f_{\sf smeared}(x)$

$$
p(x) = \int p(x|\nu)p(\nu)d\nu
$$

 $= \int d\nu p(x, y|\nu)p(\nu)$

- train on **smeared** samples with $\nu \sim p(\nu)$ to obtain fixed classifier $f_{\text{smeared}}(x)$
- we can propagate the fixed learner, but classifier not optimal for any *ν*

$$
p(x) = \int p(x | \nu) p(\nu) d\nu
$$

- train on **smeared** samples with $\nu \sim p(\nu)$ to obtain fixed classifier $f_{\text{smeared}}(x)$
- we can propagate the fixed learner, but classifier not optimal for any *ν*

$$
p(x) = \int p(x | \nu) p(\nu) d\nu
$$

$$
\overline{}
$$

- train on **smeared** samples with $\nu \sim p(\nu)$ to obtain fixed classifier $f_{\text{smeared}}(x)$
- we can propagate the fixed learner, but classifier not optimal for any *ν*

$$
p(x) = \int p(x | \nu) p(\nu) d\nu
$$

$$
\overline{}
$$

Reminder of standard statistical procedures in HEP

 p_0 to test background hypothesis

THUMBNAIL OF THE STATISTICAL PROCEDURE

Follow LHC-HCG Combination Procedures

 CL_s to test signal hypothesis

μ² to estimate signal strength

THE PROFILE LIKELIHOOD RATIO

Define **profile likelihood ratio**

- where $\theta(\mu; \mathcal{D}, \mathcal{G})$ is best fit with μ fixed (the constrained maximum likelihood estimator, depends on data) $\hat{\hat{\rho}}$ $\hat{\theta}$ $\theta(\mu; \mathcal{D}, \mathcal{G})$
- \rightarrow and $\hat{\theta}$ and $\hat{\mu}$ are best fit with both left floating (unconstrained) *̂*
- ‣ *D* denotes observed data and *G* denotes "global observables" (central values for nuisance parameters)

Consider statistical model with parameters of interest *µ* and nuisance (here called *θ*)

the true value of *μ* is *μ⁰* converges to a chi-square distribution

- ‣ distribution is known and it is independent of *θ* !
- ‣ ⇒ robust to uncertainty, a quantity like this is called a "**pivot**"
- Wilks' Theorem: under certain conditions the distribution of *-2* ln *λ (μ=μ0)* given that

21

$$
\lambda(\mu)=\frac{L(\mu,\hat{\hat{\theta}}(\mu))}{L(\hat{\mu},\hat{\theta})}=\frac{f(\mathcal{D},\mathcal{G}|\mu,\hat{\hat{\theta}}(\mu;\mathcal{D},\mathcal{G}))}{f(\mathcal{D},\mathcal{G}|\hat{\mu},\hat{\theta})}
$$

Propagating uncertainty with a pivotal classifier

•If we have a pivotal classifier, then efficiencies are independent of *ν*

analysis

 \ldots but that still doesn't mean that $f(x)$ is optimizing power / sensitivity.

•How do we obtain a pivotal classifier?

$$
\epsilon_{\text{sig}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df \qquad \epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y = 1, \nu) \, df
$$

$$
\epsilon_{\text{bkg}}(\nu) = \int_0^c p(f|y=0,\nu) \, \text{d}f
$$

Thus one won't "pay more" when accounting for systematics in the downstream

- $p(n, a | \mu, \nu) = \text{Pois}(n | \mu \epsilon_{\text{sig}}(\nu)s + \epsilon_{\text{bkg}}(\nu)b)p(a | \nu)$
	- \Rightarrow Pois(*n* | $\mu \epsilon_{\text{sig}} s + \epsilon_{\text{bkg}} b$)
		-

Learning to pivot with adversarial networks tagonistic objective is to recognize real data *X* from generated data *g*(*Z*). Both models *g* and *d* are trained simul s_{1} ITH *j*to pivot with duversal ning to pivot with adversar solution of Eqn. 4, let us note that formal guarantees train a neural network classifier *f* minimizing *L^f* (✓*^f*) Inetworks Learning to pivot with advers without considering its adversary *r*. The network aral networks l earning to pivot with adversar the former theoretical characterization of the minimax of generating training data *{xi, yi, zi}^N* rning to pivot with adversarial networks \mathbf{L} ϵ its decision scores
Superior scores of the number of the numbe r
al
nt i
y
c n
cla
... g
Dimensional to pivot with adversa t
el
c $\int f(x)$ $\begin{pmatrix} x \\ y \end{pmatrix}$)
tr
ut V
air
to line
activations t
d
e while output of the mixture of the mixture coefficient in th V
m
er tion.
In
Initialized obtained previously, adversarial training e iz
e a
a
to In particular, the conditions of the conditions of the condition S **V**
is the decision scores er
er r
Dency

Typically classifier $f(x)$ trained to minimize loss L_f . adapts to changes in *g*. At the equilibrium, *g* models a t the mean α of α and mixing coefficients of its compoany classifier $f(x)$ trained to minimize 10ss **Lf**. the classifier $f(x)$ trained to minimize loss L_f $\sum_{i=1}^n \sum_{i=1}^n \sum_{i$ Typically classifier $f(x)$ trained to minimize loss L_f . τ in the Eqn. 4, let us note that formal guarantees ϵ iypically classifier $f(x)$ trained to minimize loss E_{f} .

- · want classifier output to be insensitive to systematics (nuisance parameter **ν**) distribution whose samples can be identified by *d* only by chance. That is, assuming enough capacity in *d* and isance parameter **v** ance parametery $\mathsf{isance\ parameter\ } \mathbf{v})$ Indicance remains to be proven. (nuisance parameter v)
The assumption of existence of existence of existence of existence of existence of existence of the contract of the con oj
va
va ıssifi
sifie m
ti
	- **introduce an adversary r** that tries to predict v based on f . f because the nuisance parameter directly shapes the decioduce an **adversary r** that tries to predict duce an **adversary** r that *V*(*D, G*) = Ex⇠*p*data [log(*D*(x))] + Ez⇠*p*noise [log(1 *D*(*G*(z)))]; \bullet introduce an **adversary r** that tries to predict \vee based $\mathsf{O}(1)$. $\text{on } f$. tence of an optimal and pivotal classifier may not hold nt
bre
ct Ice an adversary r that n
mi)*,* (11) \lim_{θ} tri
m e se *f* \int $\mathbf v$
- setup as a minimax game: satisfy Eqn. 1. As illustrated in Fig. 1, we pit *f* against *H*(*Y |X*) *H*(*Z*) *< L^f* (✓*^f*) *H*(*Z|f*(*X*; ✓*^f*)) (10) <mark>p as a minim</mark> setup as a minimax game. \bullet setup as a minimax game: sion boundary. In this case, the lower bound \mathcal{S} \overline{p} as *r*

\n- introduce an adversary r that tries to predict v based on f.
\n- setup as a minimax game:
\n- $$
\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f} \max_{\theta_r} E(\theta_f, \theta_r).
$$
\n
$$
E_{\lambda}(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r)
$$
\n
	\n- $\sum_{i=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}_j \sum_{i=1}^{n} \theta_i$
	\n- $\hat{\theta}_f$
	\n\n
\n

respectively matricely processes and relationship of the relationship of the relationship of the relationship of the relations of the relations, $\frac{1}{2}$ $\overline{}$ $\begin{array}{|c|c|c|c|c|c|c|c|} \hline \bullet & \mu_0 & & \end{array}$ $\begin{array}{|c|c|c|c|c|c|c|c|c|} \hline \multicolumn{1}{|c|}{0.8} & \multicolumn{1}{|c|}{0.8} &$ $\frac{1}{2}$ $\begin{array}{|c|c|c|c|c|}\hline \text{1} & \text{$ $\frac{10}{100}$ $\frac{V=0}{100}$ $\frac{10}{100}$ $\frac{V=0}{100}$ $\frac{V=0}{100}$ $\frac{100}{100}$ $\frac{V=0}{100}$ at the nominal value *Z* = 0 would also not be pivotal. $\mathcal{L} = \mathcal{L} \setminus \mathcal{L}$ is the joint training of \mathcal{L} and $\$ adversary romants in planning the minimal contract of modeling <u>7</u> *f* 1.0 −0.5 0.0 0.5 1.0 1.5 2.0 ^{− 1.0} [−] [−] 1.0 −0.5 0.0 0.5 1.0 1.5 2.0 ^{− 0.12} $n = 0$ ers of 20 nodes with ReLU activations, followed by an $p(f(X) | \mathbf{v}=0)$
 $p(f(X) | \mathbf{v}=0)$
 $p(f(X) | \mathbf{v}=0)$
 $p(f(X) | \mathbf{v}=0)$ $\begin{bmatrix} 5 \end{bmatrix}$ $\left[\begin{matrix} 1 & \cdots & 1 \\ 0 & \cdots & 1 \end{matrix}\right]$ $\left[\begin{matrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 1 \end{matrix}\right]$ $\begin{bmatrix} 1 & \frac{1}{2} & \frac{$ ations with exponential activations of the contractions of the while output $\begin{bmatrix} 1 & 0 & 0 \ 0 & 0 & 0 \end{bmatrix}$ output $\begin{bmatrix} 1 & 0 \ 0 & 0 \end{bmatrix}$ implementation in planetation $\begin{bmatrix} 1 & 0 \ 0 & 1 \end{bmatrix}$ the softmax function to ensure positivity and normaliza-0.0 0.2 0.4 0.6 0.8 1.0 0.0 0.2 0.4 0.6 0.8 1.0
 $f(X)$ normal training adversarial training 3.0 $\begin{array}{|c|c|c|c|c|c|}\n\hline\n\cdot & \mu_0 & \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_6 & \mu_7 & \mu_8 & \mu_9 & \mu_1 & \mu_2 & \mu_3 & \mu_7 & \mu_8 & \mu_9 & \mu_1 & \mu_1 & \mu_2 & \mu_3 & \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_6 & \mu_7 & \mu_8 & \mu_9 & \mu_1 & \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_7 & \mu_8 & \mu_9$ 2.3 $\mu_1 |v|$
2.0 $\mu_2 |v|$ in Fig. 2.3 $\mu_3 |v|$ in Fig. 2.0 $\mu_1 |v|$ in Fig. 2.72 is not pivotal, as the conditional probability densities of $\begin{array}{|c|c|c|c|c|c|}\hline \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{0.48} \\\hline \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{0.48} \\\hline \multicolumn{1}{c|}{0.36} & \multicolumn{1}{c|}{0.36} & \multicolumn{1}{c|}{0.36} \\\hline \end{array}$ $\begin{bmatrix} 0.0 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} \sqrt{10} & \sqrt{10} & \sqrt{10} \\ \sqrt{10} & \sqrt{10} & \sqrt{10} \\ \sqrt{10} & \sqrt{10} & \sqrt{10} \end{bmatrix}$ $\begin{bmatrix} 0.3 & 0.0 \\ -0.5 & -0.5 \end{bmatrix}$ $\begin{array}{ccc} -0.5 \\ -1.0 \\ -0.5 \end{array}$ (a) $\begin{array}{ccc} 0.2 & -0.5 \\ 0.1 & -1.0 \\ -1.0 \end{array}$ (a) $\begin{array}{ccc} -0.5 & -0.5 \\ -1.0 & -0.5 \end{array}$ (b) $\begin{array}{ccc} 0.24 \\ 0.12 \\ -1.0 \end{array}$ 4.0 $P(f(X) | Y = +1)$
 $p(f(X) | Y = 0)$
 $\begin{bmatrix} 1 & 0 \end{bmatrix}$
 $p(f(X) | Y = -1)$
 $\begin{bmatrix} 2 & 0 \end{bmatrix}$
 $p(f(X) | Y = 0)$
 $p(f(X) | Y = -1)$ adverse as a mixture density $\begin{bmatrix} 2.5 \end{bmatrix}$ in the mixture density network $\begin{bmatrix} 2.5 \end{bmatrix}$ $\begin{bmatrix} 2 & 2 & 0 \ 2 & 2 & 0 \ 1 & 5 & 1 \end{bmatrix}$ as a mixemative of $\begin{bmatrix} 2 & 2 & 0 \ 3 & 2 & 0 \ 1 & 5 & 1 \end{bmatrix}$. network architecture of research architecture of research and research architecture of research and research a
The comprise of research and research architecture of research and research and research and research and resea $e^{0.5}$ $e^{0.2}$ notes $e^{0.2}$ $e^{0.2}$ notes $e^{0.8}$ 10 $f(X)$ f(X) normal training adversarial training $\begin{array}{|c|c|c|c|c|}\hline \text{r} & \mu_0 & & \text{ } \end{array} \qquad \qquad \begin{array}{|c|c|c|c|c|}\hline \text{r} & \text{r} & \mu_0 & \text{ } \end{array} \qquad \qquad \begin{array}{|c|c|c|c|c|}\hline \text{r} & \mu_0 & & \text{ } \end{array} \qquad \qquad \begin{array}{|c|c|c|c|c|}\hline \text{r} & \mu_0 & & \text{ } \end{array} \qquad \qquad \begin{array}{|c|c|c|c|c|c|}\hline \text{r} & \text{r$ $\begin{array}{|c|c|c|c|c|c|}\hline \bullet & \mu_1|& \vee & \bullet & \downarrow & \bullet & \bullet & \bullet & \bullet \ \hline \end{array}$ and $\begin{array}{|c|c|c|c|c|c|}\hline \bullet & \mu_1|& \vee & \bullet & \downarrow & \bullet & \bullet & \bullet \ \hline \end{array}$ and $\begin{array}{|c|c|c|c|c|c|}\hline \bullet & \mu_1|& \vee & \bullet & \downarrow & \bullet & \bullet & \bullet \ \hline \end{array}$ $\begin{array}{|c|c|c|c|c|c|}\hline \multicolumn{1}{|c|}{0.7}& & & & & \multicolumn{1}{|c|}{1.5}& & & \multicolumn{1}{|c|}{0.6}& & & \multicolumn{1}{|c|$ $\begin{array}{|c|c|c|c|c|c|}\hline \text{1} & \text{0.5} & & \text{1.0} & & \text{1.0} & \text{1.0}$ $\begin{array}{|c|c|c|c|c|}\hline \rule{0pt}{1ex} \rule{0pt}{2ex} \rule{$ the state of the nuisance parameters. The number of th $\begin{array}{|c|c|c|c|c|}\n\hline\n\text{if} & \text{if} &$ $\Box p(f(X) | \mathbf{v}=0)$
 $\Box p(f(X) | \mathbf{v}=0)$ $\begin{array}{|c|c|c|c|c|}\hline \rule{0pt}{2ex} \rule{$ $\sum_{\alpha=1}^{\infty} \frac{1}{\alpha}$ and $\sum_{\alpha=1}^{\infty} \frac{1}{\alpha}$ an network architecture of *r* comprises 2 dense hidden layers of 20 nodes with ReLU activations, followed by an output layer of 15 nodes corresponding to the means, SE
Interestingly emphasize that the
 b $p(f(X))$ the data to be pivotal, e.g. as in [19], but this is in fact $\frac{1}{\sqrt{2}}$ S
tic \mathcal{S} r i \tilde{c} $\begin{bmatrix} 1 \\ 2.5 \end{bmatrix}$ s
Sponding increase of the sponding increase of the sponding increase of the sponding increase of the sponding i
Sponding increase of the sponding increase of the sponding increase of the sponding increase of the sponding
 insensitive! \bullet $v=+1$ $v=-1$ $p(f(X))$ v=-1 $p(f(X))$ $v=+1$ $v=0$ $\cdot \quad \nu=+1$ \vee \vee \vee $v=-1$ $p(f(X)|\mathbf{v}=\mathbf{0})$ $p(f(X))$ v=-1 $p(f(X))$ $v=+1$ $p(f(X)|\mathbf{v}=0)$ $\begin{array}{c|c|c|c|c|c|c|c|c} \hline \mu_1|\nu & & & & \end{array}$ $\begin{array}{c|c|c} \hline \rule{0pt}{12pt} & 0.9 & & 2.5 & & \mu_1|\nu \end{array}$

almost independent on the number on the number of the endard deviations and mixture controls.

G. Louppe, M. Kagan, K. Cranmer, arXiv:1611.01046

The Adversarial model

…

24

the γ_1 , γ_2 , ... are the mean, standard deviation, and α amplitude for the Gaussian Mixture Model. models the distribution *p*(*z|f*(*X*; ✓*^f*) = *s*) of the nuisance parameters as observed only through the output *f*(*X*; ✓*^f*) of the $\mathbf{L} \cdot \mathbf{L} = \mathbf{L} \cdot \mathbf{L} + \mathbf{L} \cdot \math$ $fude for the Gaussian Nikture Wood. \nI $f(V|J)$$

• the neural network takes in f and predicts Y₁, Y₂,

An example of learning to pivot

Technique allows us to tune λ , the tradeoff between classification power and robustness to systematic uncertainty

25

An example:

background: 1000 QCD jets signal: 100 boosted W's

Train W vs. QCD classifier

Pileup as source of uncertainty

Simple cut-and-count analysis with background uncertainty.

Domain adaptation

In machine learning literature, the setting where training data doesn't match real world data is referred to as "domain shift" and techniques to mitigate the loss in performance are called "domain adaptation" *label predictor* (blue), which together form a standard feed-forward architecture. connected to the feature extractor via a *gradient reversal layer* that multiplies

A similar adversarial technique was introduced in <u>[arxiv:1505.07818](https://arxiv.org/pdf/1505.07818.pdf)</u> where adversary tries to get distribution of hidden state features to be invariant. This works for discrete domains, but doesn't generalize well to continuous nuisance parameters. prediction loss (for source examples) and the domain classification loss (for all \sim 100 μ s was introduced in <u>arxiv. roub.uzo to</u> wriere adver $a + \mu$ atures to be invariant. This work

• adversary works on some low-level features (not just the class prediction)

https://arxiv.org/pdf/1505.07818.pdf

GANIN, USTINOVA, AJAKAN, GERMAIN, LAROCHELLE, LAVIOLETTE, MARCHAND AND LEMPITSKY

that is not the from the gradients from the class prediction,

•One way of interpreting the mini-max game is to minimize a **regularized** loss term $L(\theta_f) = \argmax E_{\lambda}(\theta_f, \theta_r)$ where the optimization with respect to θ_r is not exposed $\boldsymbol{\widetilde{I}}$ $(\theta_f) = \arg \max_{\alpha}$ *θr* $E_{\lambda}(\theta_f, \theta_r)$ $\hat{\theta}$ $\theta_f,$ $\hat{\theta}_e$ $\theta_r = \arg \min_{\theta_f} \max_{\theta_r}$ θ_r $E(\theta_f, \theta_r).$

This motivates another approach in which the regularization is not achieved through a learned adverary, but some other measure of discrepancy

Gregor Kasieczka^{1, *} and David Shih^{2, 3, 4, \dagger}

 $L = L_{classification}(\vec{y}, \vec{y}_{true}) + \lambda \, dCorr_{y_{true}=0}^{2}(\vec{m}, \vec{y})$

27

 Ω po way of interpreting the mini put way of mittightening the minin-

Ehys.Rev.Lett. 125

DisCo Fever: Robust Networks Through Distance Correlation

$$
\begin{aligned} \mathrm{dCov}^2(X,Y) &= \langle |X-X'||Y-Y'|\rangle \\ &+ \langle |X-X'|\rangle \langle |Y-Y'|\rangle \\ &- 2\langle |X-X'||Y-Y''|\rangle \end{aligned}
$$

Is there a better way?

THE PROFILE LIKELIHOOD RATIO

Define **profile likelihood ratio**

- $\lambda(\mu) = \frac{L(\mu, \mu)}{L(\mu, \mu)}$ $\hat{\hat{\Delta}}$ $\hat{\theta}$ $\theta(\mu))$ $L(\hat{\mu},$ $\hat{\theta}^{^{\cdot}}$ $\theta)$ =
- where $\theta(\mu; \mathcal{D}, \mathcal{G})$ is best fit with μ fixed (the constrained maximum likelihood estimator, depends on data) $\hat{\hat{\rho}}$ $\hat{\theta}$ $\theta(\mu; \mathcal{D}, \mathcal{G})$
- \rightarrow and $\hat{\theta}$ and $\hat{\mu}$ are best fit with both left floating (unconstrained) *̂*
- ‣ *D* denotes observed data and *G* denotes "global observables" (central values for nuisance parameters)

- The data $\mathcal{D} = \{x_1, ..., x_N\}$ are iid, so the likelihood is just a product over events. ‣ Profiling introduces a coupling across events: $\hat{\hat{\rho}}$ $\hat{\theta}$ $\theta(\mu; \mathcal{D}, \mathcal{G})$
- ‣ But we can postpone profiling to the final inference stage and frame optimal ML model (eg. classifier) at the event-level by targeting the likelihood ratio

Consider statistical model with parameters of interest *µ* and nuisance (**here called** *θ*)

$$
\frac{f(\mathcal{D},\mathcal{G}|\mu,\hat{\theta}(\mu;\mathcal{D},\mathcal{G}))}{f(\mathcal{D},\mathcal{G}|\hat{\mu},\hat{\theta})}
$$

Likelihood Ratio Trick

$L[s] = \mathbb{E}_{p(x|H_1)}[-\log s(x)] + \mathbb{E}_{p(x|H_0)}[-\log(1-s(x))]$

Likelihood Ratio Trick

$L[s] = \mathbb{E}_{p(x|H_1)}[-\log s(x)] + \mathbb{E}_{p(x|H_0)}[-\log(1-s(x))]$

$$
\approx \frac{1}{N} \sum_{i=1}^{N} -y_i \log s(x_i) - (1 - y_i) \log(1 - s(x_i))
$$

Parametrizing the Likelihood Ratio Trick

Can do the same thing for any two points θ_0 & θ_1 in parameter space Θ .

K.C., G. Louppe, J. Pavez: Approximating Likelihood Ratios with Calibrated Discriminative Classifiers [\[arXiv:1506.02169](http://arxiv.org/abs/1506.02169)] 31

$$
r(x; \theta) = \frac{p(x|\theta)}{p_{\text{ref}}(x)} = 1 - \frac{1}{s(x; \theta)}
$$

I call this a parametrized classifier.

$$
r(x; \theta_0, \theta_1) = \frac{p(x | \theta_0)}{p(x | \theta_1)} = 1 - \frac{1}{s(x; \theta_0, \theta_1)}
$$

Or train to classify data from $p(x | \theta)$ versus some fixed reference $p_{\text{ref}}(x)$

Visualizing the parameterized classier

•We want a learner parametrized by *ν*

• augment training data $(x,y) \rightarrow (x,\nu,y)$ to obtain $f(x;\nu)$

• problem: how do we evaluate on testing data when *ν* is unknown?

Visualizing the parameterized classier

•We want a learner parametrized by *ν*

• augment training data $(x,y) \rightarrow (x,\nu,y)$ to obtain $f(x;\nu)$

• problem: how do we evaluate on testing data when *ν* is unknown?

Visualizing the parameterized classier

•We want a learner parametrized by *ν*

• augment training data $(x,y) \rightarrow (x,\nu,y)$ to obtain $f(x;\nu)$

• problem: how do we evaluate on testing data when *ν* is unknown?

Amortized likelihood ratio

 θ once we've learned the likelihood ratio $r(x; \theta)$, we can apply it to any data x.

- It is amortized, we pay biggest computational costs up front
- Great for calibrated frequentist confidence intervals with guaranteed coverage
- Here we repeat inference thousands of times & check asymptotic statistical theory

K.C., G. Louppe, J. Pavez: <http://arxiv.org/abs/1506.02169>

ibrated Discriminative Classifiers LarVis Cranmer, Louppe, Pavez: Approximating Likelihood Ratios with Calibrated Discriminative Classifiers [\[arXiv:1506.02169](http://arxiv.org/abs/1506.02169)] [Dalmasso, Izbicki, Lee, ICML2020 arXiv:2002.10399]

 F_1) with data from the simulator calibrates the rations ϵ Example \sim / **Figure** 5: Example for the back FIGURE CONSTRIBUTIONS **for the test sample.** The distribution the classifier outputs for the Estimating the density of $s(x; \theta_0, \theta_1)$ with data from the simulator calibrates the ration

Some parameterized classifier history

•2015 NeurIPS ML & Physics workshop:

- http://yandexdataschool.github.io/aleph2015/
- <https://indico.cern.ch/event/465572/> •2016 ATLAS Statistics & ML workshop:
- <https://indico.cern.ch/event/577209/> inference
- Slack channel I use for ML work is called "systematics" and has 115 members and >175,000 messages!

ALEPH Workshop @ NIPS 2015

Applying (machine) Learning to Experimental Physics (ALEPH) and «Flavours of Physics» challenge

When: 11th of December 2015, 8:30 - 18:30 Where: room 515 bc, NIPS, Montreal, Canada

•Thinking about systematics in ML is what inspired my work in simulation-based

Messages and files

now information is shared in your workspace

CO SYSTEMATICS

• Messages from member

All time Messages from members: 175,292

Some parameterized classifier history

•Thinking about systematics in ML is what inspired my work in simulation-based

Messages and files

Messages from members: 175,29

ALEPH Workshop @ NIPS 2015

Applying (machine) Learning to Experimental Physics (ALEPH) and «Flavours of Physics» challenge

When: 11th of December 2015, 8:30 - 18:30 Where: room 515 bc, NIPS, Montreal, Canada

inference

• Slack channel I use for ML work is called "systematics" started in 2016, has 115 members and >175,000 messages!

•2015 NeurIPS ML & Physics workshop:

- http://yandexdataschool.github.io/aleph2015/
- <https://indico.cern.ch/event/465572/>

•2016 ATLAS Statistics & ML workshop:

• <https://indico.cern.ch/event/577209/>

A review with other approaches

To appear in "Artificial Intelligence for Particle Physics", World Scientific Publishing Co https://inspirehep.net/literature/1807719

Istituto Nazionale di Fisica Nucleare - Sezione di Padova, Via Marzolo 8, 35131 Padova - Italy, $to mmaso. \textit{dorigo@cern.ch*} \quad \textit{pablo.de}.\textit{castro@cern.ch}$

Dealing with Nuisance Parameters using Machine Learning in High Energy Physics: a Review

T. Dorigo and P. de Castro Manzano

In this work we discuss the impact of nuisance parameters on the effectiveness of machine learning in high-energy physics problems, and provide a review of techniques that allow to include their effect and reduce their impact in the search for optimal selection criteria and variable transformations. The introduction of nuisance parameters complicates the supervised learning task and its correspondence with the data analysis goal, due to their contribution degrading the model performances in real data, and the necessary addition of uncertainties in the resulting statistical inference. The approaches discussed include nuisanceparameterized models, modified or adversary losses, semi-supervised learning approaches, and inference-aware techniques.

Tradition meets differentiable programming

³⁸ <https://github.com/pyhf/neos>

Recent efforts in particle physics to maintain traditional approaches to likelihood estimation with summaries, but optimize summary statistics with automatic differentiation statistic *s* and a statistical procedure to obtain an unbiased interval estimate of the parameter of interest approaches to likelinood estimation with its width *Ê*⁰ = ˆ*Ê*⁺ ⁰ ≠ *Ê*ˆ[≠] ⁰ , defined by some criterion so as to contain on average, upon repeated of the interval depends on the summary statistic *s* chosen: in general, summary statistics that are more informative about the parameters of interest will provide narrower confidence or credible intervals on

- **Connects to differentiable programming paradigm** *s*ú = argmin*sÊ*0*.* (2)
- Optimization objective is power of full statistical analysis, which involves backproping through statistical procedure volume or any other function of the resulting confidence or credible regions. 3 Method in detail. The method seeks to minimise the expected variance of the parameters of interest obtained
- Does not exploit i.i.d. property, optimization is "global" with an automatic differentiation framework, when $\mathbf r$ details of the statistical model as well as the expected effect of nuisance parameters.

INFERNO: de Castro & Dorigo, [arXiv:1806.04743]

Nathan Simpson @ CERN

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)

10:58 AM · Mar 5, 2020 · Twitter Web App

Tradition meets differentiable programming

³⁸ <https://github.com/pyhf/neos>

Recent efforts in particle physics to maintain traditional approaches to likelihood estimation with summaries, but optimize summary statistics with automatic differentiation statistic *s* and a statistical procedure to obtain an unbiased interval estimate of the parameter of interest approaches to likelinood estimation with its width *Ê*⁰ = ˆ*Ê*⁺ ⁰ ≠ *Ê*ˆ[≠] ⁰ , defined by some criterion so as to contain on average, upon repeated of the interval depends on the summary statistic *s* chosen: in general, summary statistics that are more informative about the parameters of interest will provide narrower confidence or credible intervals on

- **Connects to differentiable programming paradigm** *s*ú = argmin*sÊ*0*.* (2)
- Optimization objective is power of full statistical analysis, which involves backproping through statistical procedure volume or any other function of the resulting confidence or credible regions. 3 Method in detail. The method seeks to minimise the expected variance of the parameters of interest obtained
- Does not exploit i.i.d. property, optimization is "global" with an automatic differentiation framework, when $\mathbf r$ details of the statistical model as well as the expected effect of nuisance parameters.

INFERNO: de Castro & Dorigo, [arXiv:1806.04743]

Nathan Simpson @ CERN

I'm *very* excited to share with you what I've been working on recently in collaboration with @lukasheinrich !

We've developed a module that performs end-to-end learning with respect to statistical inference in particle physics.

try it yourself at github.com/pyhf/neos! :)

10:58 AM · Mar 5, 2020 · Twitter Web App

Conclusion

Traditional (non-ML) approaches used by physicists design analyses to be robust to systematic uncertainties through heuristics

• While not formalized, physicists are quite good at this

First wave of ML in physics optimized for a nominal scenario and then propagate uncertainty through the ML components in downstream inference

In order to optimize sensitivity in an ML context, we need to formalize what we want (or some heuristic) so that we can operationalize it

• "Not optimal, but not wrong"

- e.g. a pivotal classifier. Tradeoff between dependence on nuisance parameter and classification/regression performance
- likelihood ratio test in high dimensions

• a parameterized classifier: more moving parts, but closest to the ideal single-step

39

A comment on Aleatoric and Epistemic Uncertainty

•"roughly speaking, aleatoric (aka statistical) uncertainty refers to the notion of randomness, that is, the variability in the outcome of an experiment which is due to inherently random effects", while "epistemic (aka systematic) uncertainty refers to uncertainty caused by a lack of knowledge (about the best model)".

41

In the literature on Uncertainty Quantification (UQ), which is more closely connected to physics given the role of computer simulations, the terminology is more fine grained and less ambiguous.

That community uses the terms:

- parameter uncertainty (i.e. nuisance parameters),
- structural uncertainty (i.e. mismodelling),
- algorithmic uncertainty (i.e. numerical uncertainty),
- and
- due to lack of computational resources).

• experimental uncertainty (i.e. uncertainty from experimental resolution and statistical fluctuations),

• interpolation uncertainty (i.e. uncertainty due to interpolating between different parameter values

42

Perhaps a more important distinction between the perspective of physicists and machine learning researchers has to do with the use of the term "model" and what exactly is uncertain. In physics, the systematic and epistemic uncertainty is typically associated to our understanding of the underlying physics and "the model" usually refers to the physics model, detector model encapsulated in a simulation. In contrast, for machine learning research, "the model" usually refers to the trained model f ∈ F used as described in Section 41.2.1 (or the class of functions F itself). This makes sense if we recall that in the bulk of machine learning research, one has little insight into the process that generated the data (e.g. images of cats and dogs, natural language, etc.). In that sense, the epistemic uncertainty in machine learning is usually associated to uncertainty in the model parameters φ after training, which would be reduced if one could collect more training data (see Ref. [328] for this point of view).

