Machine Learning LHC Likelihoods.

Humberto Reyes-González
University of Genoa

LHC reinterpretation workshop
CERN, 14/12/2022
Introduction

- Likelihood functions (full statistical models) parametrise the full information of an LHC analysis; whether it is New Physics (NP) search or an SM measurement.
- Their preservation is a key part of the LHC legacy.

Usage:
- Resampling
- Reinterpretation with different statistical approaches.
- Reinterpretation in the context of different NP models.
- …

Challenges:
- LHC likelihoods are often high-dimensional complex distributions.
- We want precise descriptions that can be efficiently reinterpreted.

Important steps forward:
- Release of the pyhf package to construct statistical models 10.21105/joss.02823, L Heinrich, M Feickert, G Stark
- Supervised learning with DNN likelihood arxiv:1911.03305 A Coccaro, M. Perini, L Silvestrini, R Torre

Our approach: Unsupervised Learning with Normalizing Flows
LHC likelihoods in a nutshell

Bayes theorem:

\[ P(\Theta, x) = P_x(x \mid \Theta) \pi_\Theta(\Theta) = P_{\Theta}(\Theta \mid x) \pi_x(x) \]

LHC Statistical model:

\[ P(\mu, \theta; \text{data}) = \prod_{k=1}^{n_c} P[n_i; \mu, \epsilon_{i,k}(\vec{\theta})] N_{S,i,k}(\vec{\theta}) + B_{i,k}(\vec{\theta})] \prod_{j=1}^{n_{syst}} G(\theta_{j,\text{obs}}; \theta_{j}; 1) \]

Test Statistic:

\[ t(\mu) = -2 \log \frac{L(\mu; \hat{\theta}(\mu))}{L(\hat{\mu}, \hat{\theta}(\hat{\mu}))} \]

Best-fits:

\[ L(\hat{\mu}; \hat{\theta}) \]

Where \( \mu \) are observables.

Nuisance parameters (uncertainties)
Parameters of Interest (signal strength, observables, etc.)
(Observed) data
(Auxiliary) data

Best-fit \( \theta(\mu) \)
BASIC PRINCIPLE:
Following the change of variables formula, perform a series of **bijective, continuous, invertible** transformations on a *simple* probability density function (pdf) to obtain a *complex* one.

\[
Z = f(Y)
\]

\[
Y = g(Z)
\]

See review
Ivan K. et. al.
arXiv:1908.09257
Choosing the transformations

THE OBJECTIVE:
To perform the right transformations to accurately estimate the complex underlying distribution of some observed data.

THE RULES OF THE GAME:
• The transformations (bijectors) must be invertible
• They should be sufficiently expressive
• And computationally efficient (including Jacobian)

THE STRATEGY
Let Neural Networks learn the parameters of Autoregressive Normalizing Flows.

Evaluation metric.

- Two-sample 1D Kolgomonov - Smirnov test (ks test):

\[ D_{n,m} = \sup_x |F_n(x) - F_m(x)| \]

- Computes the p-value for two sets of 1D samples coming from the same unknown distribution.
- We average over ks test estimations and compute the median over dimensions.
- Optimal value 0.5
Example Likelihoods

\[ P_\Theta(\Theta | x = \text{obs}) \]

**ElectroWeak fit Likelihood**
- EW observables.
- Including recent measurements of top mass (CMS) and W mass (CDF).
- 8 parameters of interest (Wilson coefficients of SMEFT operators)
- 32 nuisance parameters.
- Ref. arXiv:2204.04204

**Flavor fit likelihood.**
- Flavor observables related to \( b \to s l^+ l^- \) transitions
- 12 parameters of interest (Wilson coefficients of SMEFT operators)
- 77 nuisance parameters.
- Ref. arXiv:1903.09632
ElectroWeak fit Likelihood

Hyperparameters:

<table>
<thead>
<tr>
<th>$N_{\text{train}}$</th>
<th>Flow</th>
<th>N bij</th>
<th>N knots</th>
<th>Range</th>
<th>Hidden layers</th>
<th>L1 factor</th>
<th>N epochs</th>
<th>N iters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>A-RQS</td>
<td>2</td>
<td>6</td>
<td>[-6,6]</td>
<td>$128 \times 3$</td>
<td>0</td>
<td>200</td>
<td>12</td>
</tr>
<tr>
<td>$10^5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2 \cdot 10^5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results:

<table>
<thead>
<tr>
<th>$N_{\text{train}}$</th>
<th>KS-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>0.453</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.4803</td>
</tr>
<tr>
<td>$2 \cdot 10^5$</td>
<td>0.486</td>
</tr>
</tbody>
</table>

Test sample : 300k
Flavor fit Likelihood

Hyperparameters:

<table>
<thead>
<tr>
<th>$N_{train}$</th>
<th>Flow</th>
<th>N bij</th>
<th>N knots</th>
<th>Range</th>
<th>Hidden layers</th>
<th>L1 factor</th>
<th>N epochs</th>
<th>N iters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^5$</td>
<td>A-RQS</td>
<td>2</td>
<td>16</td>
<td>[-6,6]</td>
<td>$1024 \times 3$</td>
<td>$10^{-4}$</td>
<td>1200</td>
<td>12</td>
</tr>
<tr>
<td>$5 \cdot 10^5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results:

<table>
<thead>
<tr>
<th>$N_{train}$</th>
<th>KS-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^5$</td>
<td>0.457</td>
</tr>
<tr>
<td>$5 \cdot 10^5$</td>
<td>0.482</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.4806</td>
</tr>
</tbody>
</table>

Test sample : 500k
Flavor fit Likelihood
EXTRA: Supervised Learning Profiled Likelihoods

\[ P_{\text{profiled}}(n_s) = P(n_s | \mu = 1, \hat{\theta}(\mu = 1)) \]

EXAMPLE ATLAS-SUSY-2018-04, 2 SRS*:

*Data generated with SModelS' Pyhf interface
Conclusions

• The preservation of LHC likelihoods is of vital importance (for theorists also).
• Introduced unsupervised learning of Likelihoods with Normalizing Flows.
• Normalizing Flows show great capacity of learning complex high dimensional functions.
• Complementary, we can directly learn profiled likelihoods; useful for fast NP-search reinterpretation.

Outlook

• User friendly Tensorflow implementation of NFs in dev: https://github.com/riccardotorre/NFTF2_dev
• Learning full statistical models with Conditional Normalizing Flows.
• Learning profiled likelihoods from Pyhf statistical models.
THANK YOU!
LHC-like new physics search Likelihood.

Test sample: 300k

Hyperparameters:

<table>
<thead>
<tr>
<th>$N_{train}$</th>
<th>Flow</th>
<th>Flow</th>
<th>Hidden layers</th>
<th>L1 factor</th>
<th>N epochs</th>
<th>N iters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>MAF</td>
<td>2</td>
<td>256 × 3</td>
<td>0</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>$10^5$</td>
<td>MAF</td>
<td>2</td>
<td>128 × 3</td>
<td>$10^{-4}$</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>$2 \cdot 10^5$</td>
<td>MAF</td>
<td>2</td>
<td>64 × 3</td>
<td>$10^{-4}$</td>
<td>20</td>
<td>4</td>
</tr>
</tbody>
</table>

Results:

<table>
<thead>
<tr>
<th>$N_{train}$</th>
<th>KS-test</th>
<th>W-distance</th>
<th>F- norm</th>
<th>HPDIE$_1$</th>
<th>HPDIE$_2$</th>
<th>HPDIE$_3$</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>0.479</td>
<td>$1.083 \cdot 10^{-2}$</td>
<td>0.913</td>
<td>$2.211 \cdot 10^{-2}$</td>
<td>$1.374 \cdot 10^{-2}$</td>
<td>$1.3003 \cdot 10^{-2}$</td>
<td>86.65</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.502</td>
<td>$5.33 \cdot 10^{-3}$</td>
<td>0.527</td>
<td>$2.157 \cdot 10^{-2}$</td>
<td>$8.147 \cdot 10^{-3}$</td>
<td>$1.07 \cdot 10^{-2}$</td>
<td>317.89</td>
</tr>
<tr>
<td>$2 \cdot 10^5$</td>
<td>.507</td>
<td>$4.82 \cdot 10^{-3}$</td>
<td>0.316</td>
<td>$1.883 \cdot 10^{-2}$</td>
<td>$9.355 \cdot 10^{-3}$</td>
<td>$9.903 \cdot 10^{-3}$</td>
<td>561.82</td>
</tr>
</tbody>
</table>

Test sample: 300k
BACK UP
Autoregressive Flows

Coupling Flows:

- Dimensions are divided in two sets: $x^A$ and $x^B$
- We transform $x^B$ with bijectors trained with $x^A$.
- The bijector parameters are functions of a NN.
- The Jacobian $J$ is triangular $\implies \det J = \prod J_{ii}$
- Jacobian is easily computed!
- Direct sampling AND density estimation.
- Less expressive.

Autoregressive Flows:

- Dimension $x^i$ is transformed with bijectors trained with $y_{1:i-1}$
- Bijector parameters are trained with Autoregressive NNs.
- The Jacobian $J$ is also triangular thus…
- Jacobian is easily computed!
- Direct sampling OR density estimation.
- More expressive.

The loss function:
$$-\log(p_{AF}(\text{target}_{dist}))$$

arXiv:1908.09257
Let’s get formal...

- If $Z$ is a random variable with pdf $p_Z$, $g$ is an invertible function such that $Y = g(Z)$ and $f = g^{-1}$, then we can obtain the pdf $p_Y$ of the random variable $Y$ as

$$p_Y(y) = p_Z(f(y)) \cdot \det(Df(y)) = p_Z(f(y)) \cdot \det(Dg(f(y)))^{-1} \quad \text{where} \quad Dg(z) = \frac{\partial g}{\partial z}, \quad Df(y) = \frac{\partial f}{\partial y}$$

- N transformations are possible since...

$$f = f_1 \circ \ldots \circ f_{N-1} \circ f_N$$

$$\det Df(y) = \Pi_{i=1}^N \det(Df_i(x_i)) \quad \text{where} \quad x_i = g_i \circ \ldots \circ g_1(z) = f_{i+1} \circ \ldots \circ f_N(y)$$

- Since $p_Z$ is parametrised by $\phi$ and the bijector $g$ by $\theta$, we can compute the log probability of some measured data $\mathcal{D} = \{y^{(i)}\}_{i=1}^M$ given the parameters $\Theta = (\theta, \phi)$ as

$$\log p(\mathcal{D} | \Theta) = \sum_{i=1}^M \log p_Y(y^{(i)} | \Theta) = \sum_{i=1}^M \log p_Z(f(y^{(i)} | \theta) | \phi) + \log |\det Df(y^{(i)} | \theta)|$$