Likelihood preservation and statistical reproduction of searches for new physics

Matthew Feickert
matthew.feickert@cern.ch

CHEP 2019

November 7th, 2019
Summary of Invited Talk
Information and Guidelines

From the CHEP page on Instructions to speakers

- **Topic**: Likelihood preservation and statistical reproduction of searches for new physics
  - [ATL-PHYS-PUB-2019-029](#)

- ATLAS Talk?: Yes

- Time Limit: 15 minutes (All parallel session talks are **12 minutes + 3 minutes** for questions)

- Slides must be prepared for projection 16:9 format (longer side along the horizontal direction) ✓
Why is the likelihood important?

- High information-density summary of analysis
- Almost everything we do in the analysis ultimately affects the likelihood and is encapsulated in it
  - Trigger
  - Detector
  - Systematic Uncertainties
  - Event Selection
- Unique representation of the analysis to preserve

![Diagram showing the relationship between CP, Ops, Analysis, Likelihood, Limits, Measurements, Data/MC plots, and Yield Tables]
Likelihood serialization...

...making good on 19 year old agreement to publish likelihoods

Massimo Corradi

It seems to me that there is a general consensus that what is really meaningful for an experiment is likelihood, and almost everybody would agree on the prescription that experiments should give their likelihood function for these kinds of results. Does everybody agree on this statement, to publish likelihoods?

Louis Lyons

Any disagreement? Carried unanimously. That’s actually quite an achievement for this Workshop.

(1st Workshop on Confidence Limits, CERN, 2000)

This hadn't been done in HEP until now

- In an "open world" of statistics this is a difficult problem to solve
- What to preserve and how? All of ROOT?
- Idea: Focus on a single more tractable binned model first
Enter HistFactory

- A flexible p.d.f. template to build statistical models from binned distributions and data
- Developed by Cranmer, Lewis, Moneta, Shibata, and Verkerke [1]
- Widely used by the HEP community for standard model measurements and BSM searches
HistFactory Template

\[ f(\vec{n}, \vec{\alpha}|\vec{\eta}, \vec{\chi}) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}_c} \text{Pois}(n_{cb}|\nu_{cb}(\vec{\eta}, \vec{\chi})) \prod_{\chi \in \vec{\chi}} c_\chi(a_\chi|\chi) \]

\[ \nu_{cb}(\vec{\eta}, \vec{\chi}) = \sum_{s \in \text{samples}} \left( \sum_{\kappa \in \vec{\kappa}} \kappa_{scb}(\vec{\eta}, \vec{\chi}) \right) \left( \nu_{scb}^0(\vec{\eta}, \vec{\chi}) + \sum_{\Delta \in \vec{\Delta}} \Delta_{scb}(\vec{\eta}, \vec{\chi}) \right) \]

**Use:** Multiple disjoint channels (or regions) of binned distributions with multiple samples contributing to each with additional (possibly shared) systematics between sample estimates

**Main pieces:**

- **Main Poisson p.d.f.** for simultaneous measurement of multiple channels
- **Constraint p.d.f. (+ data)** for "auxiliary measurements"
  - encoding systematic uncertainties (normalization, shape, etc)
- **Event rates** \( \nu_{cb} \) from nominal rate \( \nu_{scb}^0 \) and rate modifiers \( \kappa \) and \( \Delta \)
HistFactory Template

\[ f(\vec{n}, \vec{a}|\vec{\eta}, \vec{\chi}) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}_c} \text{Pois}(n_{cb}|\nu_{cb}(\vec{\eta}, \vec{\chi})) \prod_{\chi \in \chi} c_{\chi}(a_{\chi}|\chi) \]

This is a mathematical representation! Nowhere is any software spec defined.

Until now, the only implementation of HistFactory has been in RooStats+RooFit.

- Preservation: Likelihood stored in the binary ROOT format
  - Challenge for long-term preservation (i.e. HEPData)
  - Why is a histogram needed for an array of numbers?

- To start using HistFactory p.d.f.s first have to learn ROOT, RooFit, RooStats
  - Problem for our theory colleagues (generally don't want to)

- Difficult to use for reinterpretation
**pyhf: HistFactory in pure Python**

- First non-ROOT implementation of the HistFactory p.d.f. template
  - DOI: 10.5281/zenodo.1169739
- Pure-Python library as second implementation of HistFactory
  - `pip install pyhf`
  - No dependence on ROOT!
- Has a JSON spec that **fully** describes the HistFactory model
  - JSON: Industry standard, parsable by every language, human & machine readable, versionable and easily preserved (HEPData is JSON)
- Open source tool for all of HEP
  - Originated from a DIANA/HEP project fellowship and now an IRIS-HEP supported project
  - Used for reinterpretation in phenomenology paper [2]
  - Used internally in ATLAS for pMSSM SUSY large scale reinterpretation
Example pyhf JSON spec

JSON defining a single channel, two bin counting experiment with systematics

```json
{
  "channels": [  
    { 
      "name": "singlechannel",
      "samples": [  
        { "name": "signal",
          "data": [5.0, 10.0],
          "modifiers": [ { "name": "mu", "type": "normfactor", "data": null } ]
        },
        { "name": "background",
          "data": [50.0, 60.0],
          "modifiers": [ {"name": "uncorr_bkguncrt", "type": "shapesys", "data": [5.0, 12.0]} ]
        }
      ]
    }  
  ],
  "observations": [  
    { "name": "singlechannel", "data": [50.0, 60.0] }  
  ],
  "measurements": [  
    { "name": "Measurement", "config": {"poi": "mu", "parameters": []} }  
  ],
  "version": "1.0.0"  
}  
```
Example using `pyhf CLI`

```bash
$ pyhf cls example.json
{
  "CLs_exp": [0.07807427911686152, 0.17472571775474582, 0.35998495263681274, 0.6343568235898907, 0.8809947004472013],
  "CLs_obs": 0.3599845631401913
}
```
JSON Patch for new signal models

Original model

```
{
  "channels": [
    { "name": "singlechannel",
      "samples": [
        { "name": "signal",
          "data": [5.0, 10.0],
          "modifiers": [ { "name": "mu", "type": "normfactor", "data": null } ]
      }
    ]
  },
  # Rest of the model...
}
```

New Signal (JSON Patch file)

```
[
  { "op": "replace",
    "path": "/channels/0/samples/0/data",
    "value": [5.0, 6.0]
  }
]
```

Reinterpretation

```
{
  "channels": [
    { "name": "singlechannel",
      "samples": [
        { "name": "signal",
          "data": [5.0, 6.0],
          "modifiers": [ { "name": "mu", "type": "normfactor", "data": null } ]
      }
    ]
  },
  # Rest of the model...
}
```
JSON Patch for new signal models

Original analysis (model A)

Recast analysis (model B)
Likelihoods preserved on HEPData

- Background-only model JSON stored
- Signal models stored as JSON Patch files
- Together are able to fully preserve the model

Note to reviewer: Here will go a screenshot of the HEPData page but we are waiting on the SUSY conveners to upload the likelihood
...can be streamed from HEPData

```
# Nominal
$ curl -sL https://git.io/fjxE  | \ 
  pyhf cls  | \ 
  jq .CLs_obs
0.3599845631401913

# New signal
$ curl -sL https://git.io/fjxE  | \ 
  pyhf cls --patch <(curl -sL https://git.io/JeWTx)  | \ 
  jq .CLs_obs
0.4764263982925686
```
 Likelihood serialization and reproduction

- ATLAS PUB note on the JSON schema for serialization and reproduction of results (ATL-PHYS-PUB-2019-029)
  - Contours: original ROOT+XML, pyhf JSON, JSON converted back to ROOT+XML
    - Overlay of contours nice visualization of near perfect agreement
  - Serialized likelihood and reproduced results of ATLAS Run-2 search for sbottom quarks (CERN-EP-2019-142) and published to HEPData
  - Shown to reproduce results but faster! ROOT: 10+ hours pyhf: < 30 minutes
Summary

Through pyhf are able to provide:

- **JSON specification** of likelihoods
  - human/machine readable, versionable, HEPData friendly, orders of magnitude smaller

- **Bidirectional translation** of likelihood specifications
  - ROOT workspaces ↔ JSON

- Independent **pure-Python implementation** of HistFactory + hypothesis testing

- Publication for the first time of the **full likelihood** of a search for new physics

---

Massimo Corradi

It seems to me that there is a general consensus that what is really meaningful for an experiment is **likelihood**, and almost everybody would agree on the prescription that experiments should give their likelihood function for these kinds of results. **Does everybody agree on this statement, to publish likelihoods?**

Louis Lyons

Any disagreement? **Carried unanimously. That’s actually quite an achievement for this Workshop.**

(1st Workshop on Confidence Limits, CERN, 2000)

---

ATLAS PUB Note

ATL-PHYS-PUB-2019-029
5th August 2019

Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods

The ATLAS Collaboration

(ATLAS, 2019)
Backup
pyhf team

Lukas Heinrich  
CERN

Matthew Feickert  
Illinois

Giordon Stark  
UCSC SCIPP

Kyle Cranmer  
NYU

Core Developers

Advising
ROOT + XML to JSON and back

original workspace
ROOT HistFactory

Measurement::PrintXML()

original
ROOT+XML

pyhf xml2json

round-tripped
ROOT HistFactory

hist2workspace

pyhf json2xml

pyhf workspace
JSON HistFactory

CLs

round-tripped
ROOT+XML

CLs
Best-fit parameter values
JSON Patch files for new signal models

$ pyhf cls example.json | jq .CLs_obs
0.3599845631401913

$ cat new_signal.json
[
{
    "op": "replace",
    "path": "/channels/0/samples/0/data",
    "value": [5.0, 6.0]
}
]

$ pyhf cls example.json --patch new_signal.json | jq .CLs_obs
0.4764263982925686
...which can be streamed from HEPData

```bash
$ curl -sL https://git.io/nominal | \n  pyhf cls --patch <(curl -sL https://git.io/newsignal)

# Nominal
$ curl -sL https://git.io/fjxXE | \n  pyhf cls | \n  jq .CLs_obs
0.3599845631401913

# New signal
$ curl -sL https://git.io/fjxXE | \n  pyhf cls --patch <(curl -sL https://git.io/JeWTx) | \n  jq .CLs_obs
0.4764263982925686
```
References


The end.