IRIS-HEP Fellowship Proposal:

Designing and implementing a converting tool for statistical models between pyhf and CMS combine

Peter Ridolfi
University of Pittsburgh

April 2022

Project Details:

The use of statistical models to accurately represent and interpret data from high-energy physics experiments is crucial to gaining a better understanding of the results of those experiments. However, there are many different methods and models that researchers are using for these representations, and although they often generate results that are useful for everyone in the field of HEP, they also often slightly deviate in results between different models, and that deviation is difficult to interpret. Fortunately, many statistical models use similar frameworks, as well as the same mathematics, so it is quite feasible to convert a model generated in one environment to a different environment. This will allow the results to be more consistently replicated across models, as well as give deeper insight into certain differences in results between models. In addition, both phyf and CMS offer unique tools within their respective models to interpret them, and an easy conversion would allow someone that is familiar with one environment to develop the model in that environment, and then transfer it to the other in order to take advantage of both sets of tools. The desire for such a converter is shown in the thread in [2].

So, the purpose of this project will be to create a conversion method to accomplish this task, specifically converting between pyhf models and CMS combine models. Both pyhf and CMS Combine employ statistical models that are constructed from a set of building blocks. Many of these building blocks are mathematically equivalent in both cases, but serialized differently. For cases where the building blocks are equivalent, the aim of the project is to translate the serialized statistical model correctly.

For some model aspects there is no direct translation possible. For these cases, the project will examine whether an approximate translation is possible by employing similar building blocks. For model aspects that cannot be translated in a useful manner, the converter should catch this and inform the users. Developing a way to determine an approximate translation and determining if one is not possible is the main challenge which the bulk of this project will be based on. For example, pyhf generates model predictions with the following formula [3]:

$$f(\boldsymbol{x}|\boldsymbol{\phi}) = f(\boldsymbol{x}|\overbrace{\boldsymbol{\eta}}^{\text{free}}, \underbrace{\boldsymbol{\chi}}_{\text{constrained}}) = f(\boldsymbol{x}|\overbrace{\boldsymbol{\psi}}^{\text{parameters of interest}}, \underbrace{\boldsymbol{\theta}}_{\text{nuisance parameters}})$$

While CMS Combine uses a similar overall structure, using parameters and uncertainty factors to calculate expected data[1], there is a slight difference in the way that parameters of interest/nuisance parameters are declared in the model, for example, so those features cannot be directly converted from phyf to CMS. Thus, there is a possibility of information loss when transferring the model. As such, the conversion will need to be developed to bridge these gaps, thus allowing users to take full advantage of the unique aspects of each model. As previously mentioned though, the mathematical methods behind the models are so similar that such a conversion is a very realistic possibility.

Another factor that will need to be considered is that a perfect converter is not possible for these two environments, mainly because CMS Combine covers a much larger statistical scope than pyhf, and thus contains some factors that don't contribute at all to pyhf models. However, the conversion is attainable to a certain degree of accuracy for the subset of models that pyhf implements. Potential differences in the likelihood created by CMS Combine and pyhf, for example due to aspects like different interpolation algorithms between templates, will need to be understood and quantified for some example setups.

This research will be done under the mentorship of Prof Kyle Cranmer, Dr. Matthew Feickert, and Dr. Alexander Held.

Timeline:

- Weeks 1-2: Familiarize myself with the pyhf framework and the statistical methods that it implements. Do the same with CMS Combine, and keep track of significant differences and similarities. Create a public GitHub repository for the project.
- Weeks 3-4: Generate sample models in both pyhf and CMS Combine to get comfortable with the programming structure and syntax, and also deepen my understanding of both types of models.
- Weeks 5-6: Research each aspect of both models and try to make a mathematical connection to the other model and compare models between both implementations. Sort building blocks to plan an order in which to address their translation in, and start planning the converter structure.
- Week 7: Implement a converter that can translate a subset of building blocks from the JSON model from pyhf and converting it to Combine's datacard structure, as well as parsing through models in the datacard structure and putting them into the pyhf JSON format.
- Weeks 8-10: Expand the converter with additional building blocks. . After a successful converter has been created, test it on specific models and compare the results to show

its effectiveness as well as to analyze differences in the likelihoods when converting models.
- Weeks 11-12: Spend time thoroughly documenting the code and motivation behind the implementation. Create a presentation that explains the converter and demonstrates its practicality and accuracy. Create tutorial material with usage examples.

It should be noted that I do not have any academic or professional commitments between the dates of 5/9 and 8/19, so this timeline is very flexible and can be adjusted to fit the needs of the project to ensure its completion.

References:

[1] CMS Combine Documentation. Retrieved April 12, 2022, from https://cms-analysis.github.io/HiggsAnalysis-CombinedLimit/part2/settinguptheanalysis/

[2] Possible to integrate CMS's Combine workflow? 2018. Retrieved April 14, 2022, from https://github.com/scikit-hep/pyhf/issues/344

[3] pyhf v0.6.3 Documentation. 2018. Retrieved April 12, 2022, from https://pyhf.readthedocs.io/en/v0.6.3/