GENIE Systematic Uncertainties with the Multi-universe Approach in CAFAna

1. Overview

The standard approach for drawing the NOvA systematic error band from GENIE cross section models is to vary the GENIE tunable physics parameters by N σ's, $N \in \{-2, -1, 1, 2\}$, input this number through the shift parameter of the CAFAna Spectrum class, and obtain shifted spectra as the boundaries of the error band. There are tens of tunable GENIE parameters, hereafter referred to as GENIE knobs, but only a few of them having the biggest effects are varied. This leaves somewhat arbitrary where to leave out the knobs.

A consistent way of treating the GENIE systematic uncertainties across all analyses, known as the multi-universe approach, was proposed, which varies all the knobs at the same time. One advantage of this approach is it enables bin-to-bin correlation studies which could be in turn followed by dimensionality reduction analyses such as principal component analysis (PCA) [1].

In this note details of the implementation of this approach under the CAFAna framework is given, followed by sanity checks of this approach compared with the conventional way. Then some known issues are discussed. In the end the computing resources consumed by this class is presented.

2. Implementation

The code can be found, under NOvA's offline repository, in `CAFAna/XSec/GenieMultiverseSyst.[h,cxx]`. There are several classes defined in the files, where `MultiverseSpectra` takes on a central role.

Since there are tens of GENIE tunable knobs, to switch the knobs on and off easily, a so called knob configuration file is input into the class with a default name of `knob_config.txt` and a default path of the current directory.

With the development of theories, new knobs could be added into GENIE. To account for the possible difference in the tunable parameters used in different production datasets, a script is offered to make a configuration file based on the underlying GENIE versions of the production datasets, located in `CAFAna/XSec/Utilities/make_template_knob_config.py`. To use this scrip, simply do:

(i) setup nova

(ii) `$ python make_template_knob_config.py -d <sam_defname>`

This script generates a complete set of GENIE knobs with the name of a knob followed by a number. The names are extracted from the source file `$NUTOOLS_DIR/source/NuReweight/ReweightLabels.h`, with the leading "`fReweight`" or "`kReweight`" stripped. The number following a knob name assumes 3 numbers, namely 0, 1, or 2. 0 means it is disabled, like it does not exist at all. If it is 1, a random number is drawn from a normal distribution $\mathcal{N}(0,1)$, and used to construct a `SystShifts` object as an input argument to the `Spectrum` class. If it is 2, it means the knob chooses between two alternative models. In this case, a random number is drawn from $\{0,1\}$ with equal probability, where 0 represents the nominal model, and 1 represents the alternative model. Note that even for this kind of knobs mode 1 can still be used. In this case CAFAna does a linear evolution from the weight of the first model to that of the second model.

A default configuration file is available in `CAFAna/XSec/Utilities/knob_config.txt`. By and large it enables all available knobs, and uses mode 2 whenever the knob is for alternative models. However, some knobs tune the same physical parameter. For example, `MaCCQE` tunes the normalization and shape for CCQE at the same time, and `NormCCQE` and `MaCCQEshape` tune normalization and shape of CCQE, respectively. To avoid double counting, the three knobs should not be enabled at the same time. According to the former studies [‡], decision was made to enable separate knobs and disable the combined one.

2.1. Interface

The interface is designed to mimic that of the `Spectrum` class as close as possible, with two more constructor arguments. One is the number of universes one would

[‡] Please refer to page 15 in DocDB-15214.

like to generate, and the other is the pathname of the knob configuration file. Please consult the doxygen page for detailed information.

Once you have created a multi-universe object and filled them with the `SpectrumLoader::Go()` function call, you can start to extract the needed information, namely, the error band.

To obtain the spectrum above the nominal spectrum, use `MultiverseSpectra::UpperSigma()`. Similarly, `MultiverseSpectra::LowerSigma()` gives you the spectrum below the nominal one. The boundaries of the error band are obtained in the following way. For each bin of the spectrum, each universe has a different number of events.

References

[1] Bannanje Nitish Nayak, PPFX Systematics, `https://nova-docdb.fnal.gov:441/cgi-bin/ShowDocument?docid=18996`