Averaging Tool

Thomas Kuhr

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

10.02.2021

TUM LMU ORIGINS Excellence Cluster

Minutes from Last Meeting

- https://codimd.web.cern.ch/4xM0n0vkSye86zYN_8LQgQ#

25.01.2021

- The json format seems fine. Using a directory structure for the organization of parameters and human readable parameter identifiers are preferred. A layout configuration independent of the parameter definition is more flexible.
- Eli will contact somebody from PDG and ask about the long term stability of the API and if the decay channel number could be added to the json file. Maybe a PDG member could even join our meeting.
- The general idea of constructing and minimizing a likelihood was accepted. The input should be specified in a way that is as close as possible to what is quoted in the papers.
- The requirements from the rare decays group are:
  - Marking of preliminary results and results that are not included in the PDG average.
  - Footnotes for parameters/averages, publications, measurements. The same footnote may be shared among multiple of those.
  - Parameters in ranges of kinematic variables. May be included in the parameter name (instead of an extra column).
- Thomas will make a proposal of work packages.
- The next meeting will be in the week of February 8.

Work Packages

- Revision of Measurement class to allow for specification of dependencies on input/nuisance parameters and the construction of a negative log likelihood (NLL)
- Revision of Publication class to allow for marking of preliminary/new results, change identifier to inspire ID
- Revision of Parameter class to support input/nuisance parameters and reference to PDG
- Performant likelihood fit
- More modular code structure
- More flexible scheme for configuration of output generation, decouple output generation from averaging
- Scheme for presenting correlations and changes due to updated inputs
- Change data format from xml to json
- Conversion from xml to json
- Document design
- Documentation for users
- Update inspire import to new API
- API for our averages
- Code review
- Beta testing
- Unit / integration tests?

Parameter Class

```python
def decode_json(dct):
    if 'name' in dct:
        return Parameter(dct)
    elif 'measurements' in dct:
        return Publication(dct)
    elif 'result' in dct or 'nll' in dct:
        return Measurement(dct)
    else:
        return dct

with open('parameters.json') as parameter_file:
    parameters = json.loads(parameter_file.read(),
                            object_hook=decode_json)
```

```python
class Parameter:
    """Representation of a parameter."""

    def __init__(self, dct, name = None):
        """Initialize the parameter object from a dictionary."""

        self.name = name      # identifier
        self.text = None      # text representation
        self.latex = None     # latex representation
        self.pdgId = None     # identifier used by PDG
        self.pdgOrd = None    # ordinal number used by PDG
        self.comment = None   # a literal comment or an identifier of a comment that will be displayed as a footnote

        for attr in self.__dict__:
            if attr in dct.keys():
                self.__setattr__(attr, dct[attr])

        self.pdgPubs = []     # list of inspire IDs of publications that are included in the PDG average,
                              # obtained from PDGIdentifiers-references.json
        self.index = -1       # index of fit parameter
```

```
{"name": "BR_B0bar_DS-_D+",
 "pdgId": "S042:Desig=50",
 "latex": "{\\cal{B}} ( \\bar{B}^{0} \\to D^{*}(2010)^{-} D^{+} )"},
```

- Explicit marking of nuisance parameters maybe not needed

Publication Class

```
class Publication:
    """Representation of a publication with measurements."""

    def __init__(self, dct):
        """Initialize the publication object from a dictionary."""

        self.inspire = None      # inspire identifier
        self.arxiv = None        # arXiv identified
        self.journal = None      # journal information: title, volume, id, year
        self.doi = None          # digital object identifier
        self.altref = None       # array of reference name and link that will be used if no journal reference or arXiv
                                 # preprint is available
        self.experiment = None   # name of collaboration
        self.comment = None      # a literal comment or an identifier of a comment that will be displayed as a footnote
        self.superseded = False  # a flag that indicates if the result is superseded
        self.bibtex = None       # a string containing the bibtex entry
        self.measurements = []   # array of measurements

        for attr in self.__dict__:
            if attr in dct.keys():
                self.__setattr__(attr, dct[attr])

        for entry in self.measurements:
            entry.publication = self
```

Measurement Class

```
class Measurement:
    """Representation of a measurement."""

    def __init__(self, dct):
        """Initialize the measurement object from a dictionary."""

        self.comment = None      # a literal comment or an identifier of a comment that will be displayed as a footnote
        self.superseded = False  # a flag that indicates if the result is superseded
        self.result = None       # a string representation of the measured quantity and its central value and uncertainties
        self.inputs = None       # array of string representations of values and uncertainties used as external input
        self.nll = None          # negative log likelihood function

        for attr in self.__dict__:
            if attr in dct.keys():
                self.__setattr__(attr, dct[attr])

        self.publication = None  # publication quoting the measurement
```

- Construction of NLL from result/inputs string not implemented yet

Fake Example


```
{
    "inspire": 747271,
    "arxiv": "hep-ex/0703040",
    "journal": {"title": "Phys. Rev.", "volume": "D75", "id": "091102", "year": 2007},
    "doi": "10.1103/PhysRevD.75.091102",
    "experiment": "Belle",
    "bibtex": "@article{Zupanc:2007pu,\n      author         = \"Zupanc, A. and Abe, K.
    "measurements": [
        {
            "result": "BR_B0bar_Ds-_D+ = 7.5 +-0.2 +-0.8 +-0.8(CorrBrDs) x 10-3",
            "inputs": ["+*f00 = 0.5"]
        },
        {
            "result": "BR_B0bar_Ds-_Ds+ < 3.6e-5 @ 90% CL"
        }
    ]
}
```


- Inputs: ‘+’ for additional systematics not incl. in result string, ’*’ for daughter BR, ‘/’ for normalization BR
- Corr nuisance parameters if used value/uncertainty unknown, only correlation known: factor (1 + error*par), par normal distr.

Sharing of Work

- Revision of Measurement class to allow for specification of dependencies on input/nuisance parameters and the construction of a negative log likelihood (NLL)
- Revision of Publication class to allow for marking of preliminary/new results, change identifier to inspire ID
- Revision of Parameter class to support input/nuisance parameters and reference to PDG
- Performant likelihood fit
- More modular code structure
- More flexible scheme for configuration of output generation, decouple output generation from averaging
- Scheme for presenting correlations and changes due to updated inputs
- Change data format from xml to json
- Conversion from xml to json
- Document design
- Documentation for users
- Update inspire import to new API
- API for our averages
- Code review
- Beta testing
- Unit / integration tests?

Backup: Dependency on Inputs

- Basic idea: specify likelihood with common input / nuisance par.

- Example: Measurement of BR(B → DX) where BR(D → Kπ) was used
- External input parameter with neg. log likelihood:

$$B_D := \mathcal{B}(D \to K\pi) = m_D \pm \sigma_D \qquad \Rightarrow NLL = 0.5 \left(\frac{B_D - m_D}{\sigma_D} \right)^2$$

- Measurement of BR(B → DX):

$$B_B := \mathcal{B}(B \to DX) = m_B \pm \sigma_B(\text{stat+syst}) \pm \sigma_{DK\pi}(\text{syst. for D BR})$$

where $\mathcal{B}(D \to K\pi) = m'_D \pm \sigma'_D$ was used, so $\frac{\sigma'_D}{m'_D} = \frac{\sigma_{DK\pi}}{m_B}$

- Resulting NLL:

$$\Rightarrow NLL = 0.5 \left(\frac{B_B \cdot m'_D / B_D - m_B}{\sigma_B} \right)^2$$