The NNPDF open source code

ACAT 2021

Zahari Kassabov

November 30, 2021

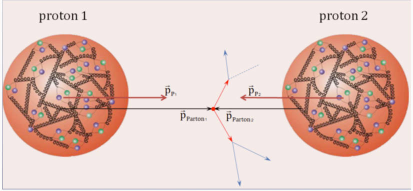
DAMTP, University of Cambridge

An open-source machine learning framework for global analyses of parton distributions

- https://github.com/NNPDF/nnpdf
- https://docs.nnpdf.science/
- arxiv:2109.02671, Eur.Phys.J.C 81 (2021)

Parton distribution functions (PDFs)

- Large Hadron Collider colludes protons.
- Fundamental particle interaction theory given in terms of *partons* (quarks, gluons).
- PDFs describe the constituents of the proton.
- Every collider analysis needs to know about them.



Cannot be determined from first principles.

PDF determination as a Machine Learning problem

Collider data should be described from physical law $\longrightarrow$ obtain physical law from data.

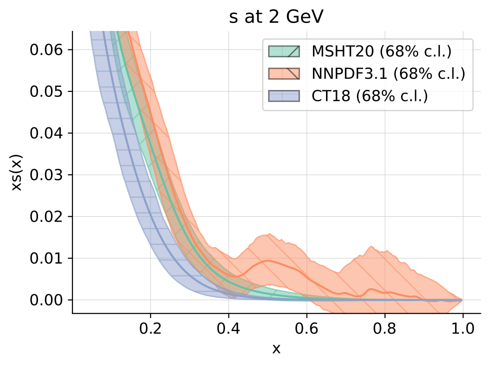
The NNPDF collaboration has been solving the problem using neural networks since the early 2000s.

Characteristics:

- "Low" data.
- Complicated data processing.
    - Need to understand both experiment and theory well.
    - Data science problem
- Uncertainty estimation is crucial.

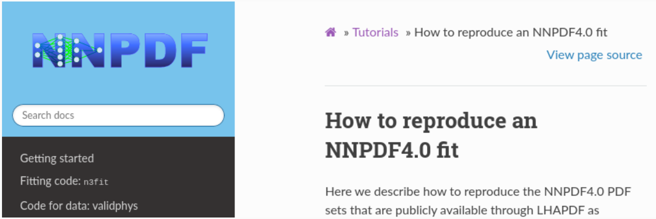
Open source PDFs

- Critical of LHC analysis necessitates from hard-to-reproduce analyses by experts.
- Differences between analyses are hard to understand



- Ability to look into the code provides reassurance.

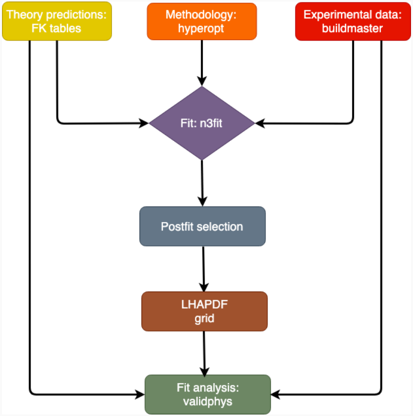
Looking inside



- Fitting code
- Data science framework
- All of the data
- Documentation
- Python code base (with some, mostly legacy C++)

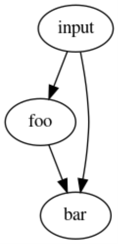
Fitting code

- The `n3fit` code, originally presented in [Carrazza, Cruz, 2019]
- Based on Tensorflow, with access to all its algorithms and target processors.
    - Order of magnitude faster than previous genetic algorithm based implementation.
- Hyperoptimization framework that helps select a good algorithm.



Analysis code

- A `reportengine` application [ZK, 2019]
    - Functional code structure
    - Runcard driven
    - Graph checks and execution
- Core data structures
- Analysis code and plots




```yaml
# runcard.yaml
input: value

actions_:
  - bar
```

```python
# mymodule.py

def foo(input):
    return ...


@check_input_is_right_for_bar
def bar(foo, input):
    return ...
```

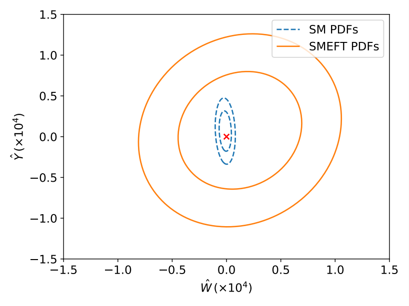
Building up

More complex features in `reportengine` (namespaces, dependency induction) allow building fairly complex data science pipelines, still reproducible with a declarative runcard.

- Typically the final output is an HTML report.
- E.g. `vp-comparefits` (included in the code) produces hundreds of plots and tables comparing two NNPDF fits.
- Comprehensive closure test framework.
- Dozens of specialized plotting tools.
- The fitting code itself.

Extensions

- The code structure opens the door for interesting avenues.
  - E.g. extend to nuclear fits
- Precise enough analyses *need* simultaneous fits [Forte, ZK, 2020]
  - Including of EFTs



Greljo et al, 2021

- Watch out the PBSP group in Cambridge in this space!