Machine learning the parton distribution functions

Reducing methodological bias

Roy Stegeman

University of Milan and INFN Milan

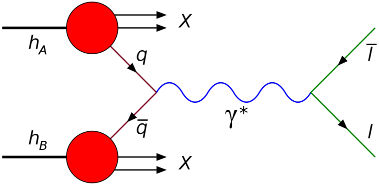
Congresso Nazionale SIF
14-18 September 2020

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 740006.

Motivation

Parton distribution functions (PDFs)

- PDFs are required to provide theoretical predictions
- Hadron collisions are factored into a ’hard part’ $\hat{\sigma}$ and a normalization provided by the PDFs
- We cannot calculate PDFs



$$\overbrace{\sigma_X}^{\text{experiment}} = \sum_{a,b} \int dx_1 dx_2 \overbrace{f_{a/h_A}(x_1) f_{b/h_B}(x_2)}^{\text{PDFs}} \overbrace{\hat{\sigma}_{ab\to X}}^{\text{theory}}$$

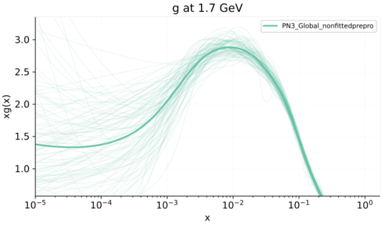
Methodology

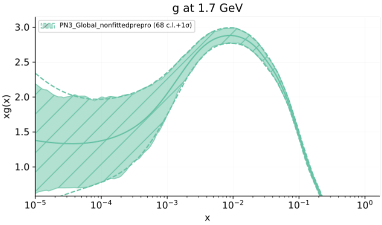
The NNPDF collaboration

- NNPDF provides PDF determination using Neural Networks

The NNPDF methodology

- Using a Neural Network reduces bias from the functional form
    - $f_i = A_i x^{\alpha_i}(1-x)^{\beta_i}\mathrm{NN}_i(x, \log x)$
- Monte Carlo set of PDFs
- Minimization using gradient descent
- Stopping based on training/validation data





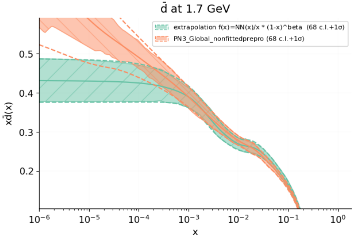
Removing bias

Bias

The methodology is still not completely free of bias

$f_i = A_i x^{\alpha_i} (1 - x)^{\beta_i} \mathrm{NN}_i(x, \log x)$

If preprocessing is removed, we observe saturation at small-x:

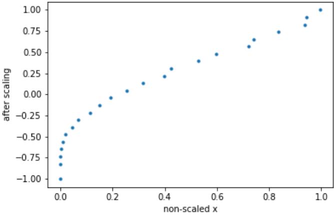


Feature Scaling

Solution:

1. Scale the input xgrid such that it is homogeneously distributed
2. Select one in *n* points
3. Provide a monotonically increasing interpolation

Result: $f_i = \mathrm{A}_i \left(\mathrm{NN}_i(x') - \mathrm{NN}_i(1)\right)$

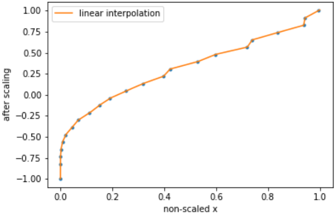


Feature Scaling

Solution:

1. Scale the input xgrid such that it is homogeneously distributed
2. Select one in *n* points
3. Provide a monotonically increasing interpolation

Result: $f_i = \mathrm{A}_i\,(\mathrm{NN}_i(x') - \mathrm{NN}_i(1))$

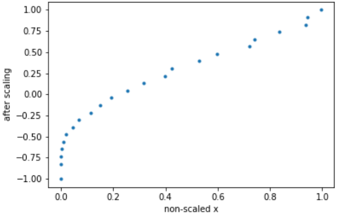


Feature Scaling

Solution:

1. Scale the input xgrid such that it is homogeneously distributed
2. Select one in n points
3. Provide a monotonically increasing interpolation

Result: $f_i = \mathrm{A}_i\,(\mathrm{NN}_i(x') - \mathrm{NN}_i(1))$

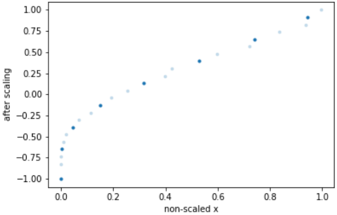


Feature Scaling

Solution:

1. Scale the input xgrid such that it is homogeneously distributed
2. Select one in n points
3. Provide a monotonically increasing interpolation

Result: $f_i = \mathrm{A}_i \left(\mathrm{NN}_i(x') - \mathrm{NN}_i(1)\right)$

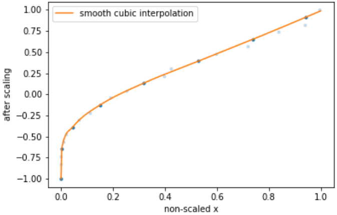


Feature Scaling

Solution:

1. Scale the input xgrid such that it is homogeneously distributed
2. Select one in n points
3. Provide a monotonically increasing interpolation

Result: $f_i = \mathrm{A}_i\,(\mathrm{NN}_i(x') - \mathrm{NN}_i(1))$

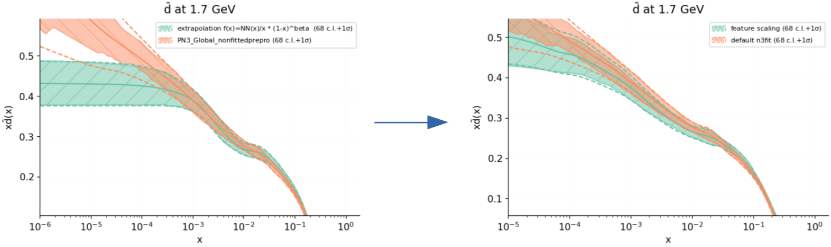


Feature Scaling

Solution:

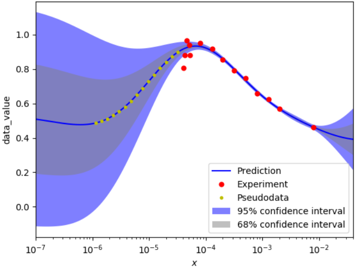
1. Scale the input xgrid such that it is homogeneously distributed
2. Select one in n points
3. Provide a monotonically increasing interpolation

Result: $f_i = \mathrm{A}_i\left(\mathrm{NN}_i(x') - \mathrm{NN}_i(1)\right)$



Outlook: the extrapolation region

- Now we no longer have any prediction for the extrapolation region
- Use Gaussian Process Regression to fit observables + uncertainty
- Generate observables in the extrapolation region
- Include this Gaussian pseudodata in the NNPDF fit



Thank you!