Hyperoptimization - detecting overfitting

Roy Stegeman

University of Milan and INFN Milan

NNPDF meeting, 11 April 2022, Amsterdam

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

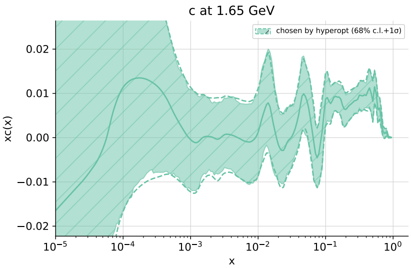
Hyperparameter selection

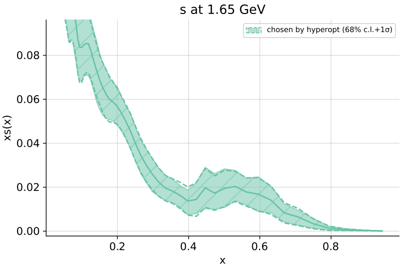
Currently: k-folds Hyperoptimization

This results in the possiblity of overfitted or underfitted setups, in part due to fluctuations (think preprocessing exponents)

To get a “nice” PDF we do a manual selection after the automated hyperoptimization, re-introducing human bias

To reduce bias we would like a numerical objective metric for overfitting or underfitting





The idea

Ideally, we have an objective metric that is not relative (such as arc-length), but absolute

Correlation between PDFs and validation data suggests overfitting

How can we detect when this happens?

The idea

Realization: for any PDF the validation loss χ^r_{val} should be equal to the "validation loss" calculated for any other pseudodata set $\chi^{\hat{r}}_{\mathsf{val}}$ (with the same tr/vl mask)

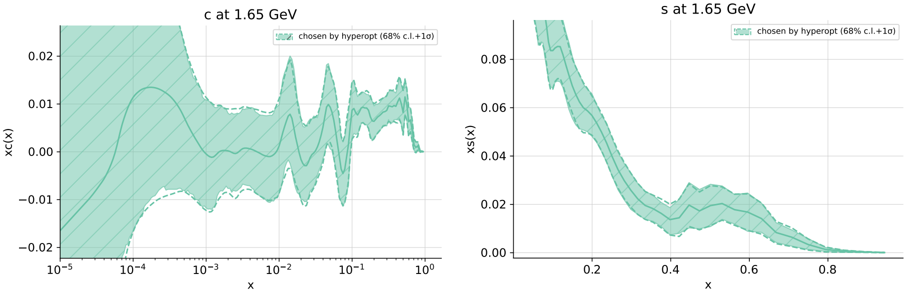
Thus as a metric for overfitting we might consider

$$\Delta\chi^2_{\mathsf{overfit}} = \langle \chi^2_{\mathsf{val},\hat{r}} - \chi^2_{\mathsf{val,r}} \rangle \quad (< 0 \text{ if overfitted})$$

While **underfitted** setups will be filtered due to their higher χ^2 values

(How) does this work?

Let's have a look at a clearly overfitted PDF (preferred by hyperopt):

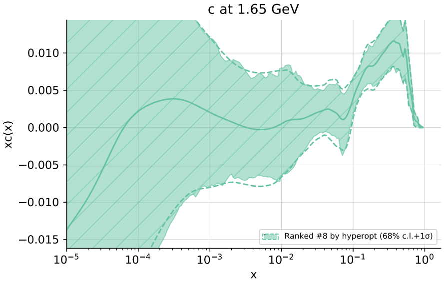


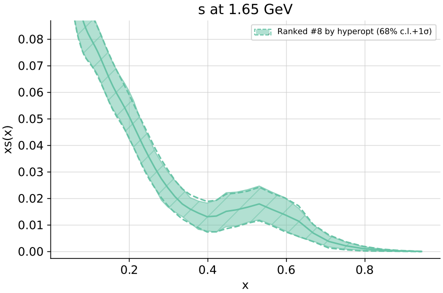
$$\Delta\chi^2_{\text{overfit}} = -0.0459 \pm 0.0078 \qquad 5.9\sigma \text{ from } 0$$

The $\Delta\chi^2_{overfit}$ values and bootstrap errors in these slides are determined using PDFs with $N_{rep} = 100$

(How) does this work?

And now for a PDF that is a bit smoother (ranked #8 by hyperopt):



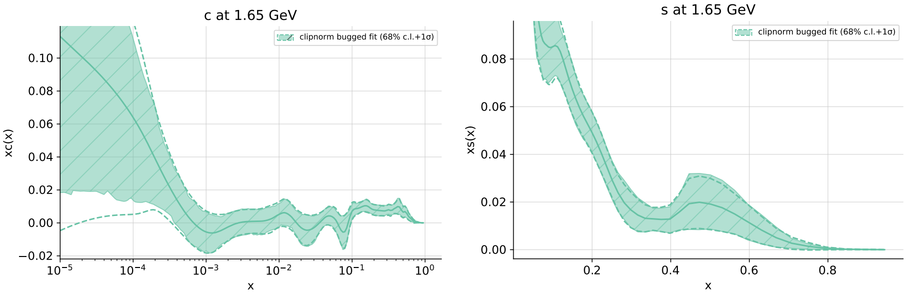


$$\Delta\chi^2_{\text{overfit}} = -0.0168 \pm 0.0105 \qquad 1.6\sigma \text{ from } 0$$

The distance from 0 decreases as expected

(How) does this work?

In the past we have had a scenario where this metric would have helped a lot:

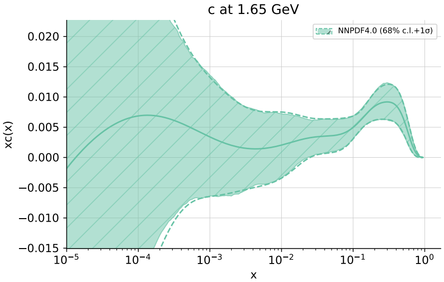


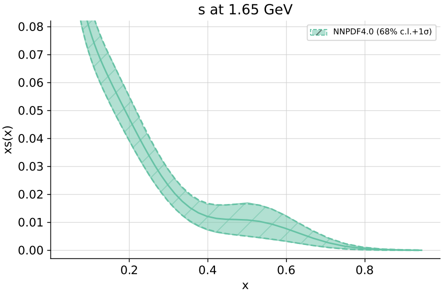
$$\Delta\chi^2_{\text{overfit}} = -0.0236 \pm 0.0126 \qquad 1.9\sigma \text{ from } 0$$

A clear indicator that the clipnorm bugged fit is overfitted!

(How) does this work?

And what about NNPDF4.0?





$$\Delta\chi^2_{\text{overfit}} = -0.0012 \pm 0.0130 \qquad 0.1\sigma \text{ from } 0$$

How can this be used in NNPDF?

As an a-posteriory check similar to (but cheaper than) the closure test

1. Run hyperoptimization
2. Select N best setups and do full 100 replica fits for each
3. Calculate the estimators for all
4. Discard setups with e.g. $R_{\text{overfit}} < -1$
5. 5.1 Increase number of replicas and repeat...
   5.2 or select the best of the remaining fit

5.1: If the bootstrap error becomes small enough we will likely always get a negative $\Delta\chi^2_{\text{overfit}}$
5.2: What is an acceptable $\Delta\chi^2_{\text{overfit}}$?
How do we define the best fit (χ^2_{val}, χ^2_{tr}, χ^2_{exp}, $\ldots$)?

How can this be used in NNPDF?

As an a-posteriory check similar to (but cheaper than) the closure test

1. Run hyperoptimization
2. Select N best setups and do full 100 replica fits for each
3. Calculate the estimators for all
4. Discard setups with e.g. $R_{\text{overfit}} < -1$
5. 5.1 Increase number of replicas and repeat...
   5.2 or select the best of the remaining fit

5.1: If the bootstrap error becomes small enough we will likely always get a negative $\Delta\chi^2_{\text{overfit}}$
5.2: What is an acceptable $\Delta\chi^2_{\text{overfit}}$?
How do we define the best fit (χ^2_{val}, χ^2_{tr}, χ^2_{exp}, . . .)?

Conclusion: The $\Delta\chi^2_{\text{overfit}}$ provides a metric for overfitting that can be used to flag overfitted hyperparameter setups and thereby reduce human bias

Backup

Hyperopt demonstration

Demonstration file