

Template Fitting Including Systematics

Belle II Summer School

Markus T. Prim

4th August 2022

Universität Bonn - Physikalisches Institut

Method of Maximum Likelihood

The method of maximum Likelihood is a technique for estimating the values of model parameters given a finite sample of data.

$$\mathcal{L}(\theta) = \prod_{i=1}^n f(x_i, \theta), \quad (1)$$

The Likelihood function \mathcal{L} is the joint probability distribution of the sample, **only depending on the model parameter(s) θ** , viewing the measured data x_i as fixed. We determine the model parameter θ which maximizes the probability to draw the sample data by **maximizing** the Likelihood

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = 0, \quad i = 1, \dots, m. \quad (2)$$

To get the best possible result for the parameter estimation, ideally all information for the model is encoded into the Likelihood.

Method of Maximum Likelihood

We usually use the negative logarithm of the Likelihood function:

$$-2 \ln \mathcal{L}(\theta) = -2 \sum_{i=1}^n \ln f(x_i, \theta). \quad (3)$$

The reasons are:

- There are only minimizes on the market.
- Calculating the product of many small values is not computational stable.
- The 2 is just convention, but important in the common way to interpret Likelihood Profiles (more later).

Method of Maximum Likelihood for Templates i

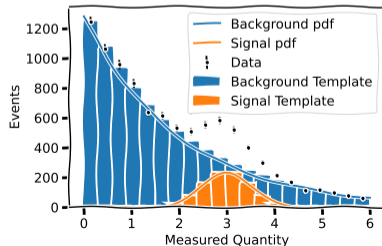
In this talk, we will discuss the method of maximum Likelihood when we use templates for our model. This is useful, e.g. when we do **not have a good parameterization for the model** and we want to rely on the event generator plus detector simulation.

⇒ The probability density function $f(x_i|\theta)$ of our model is approximated by a histogram with n bins. The content of each bin follows a Poisson distribution $\mathcal{P}(n_i|\nu_i)$.

The Likelihood function becomes

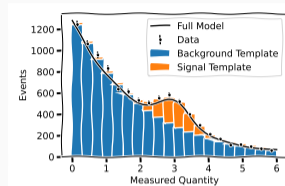
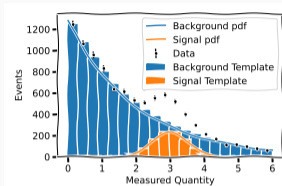
$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i|\nu_i). \quad (4)$$

Let us decompose this ...



Method of Maximum Likelihood for Templates ii

$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i | \nu_i). \quad (5)$$

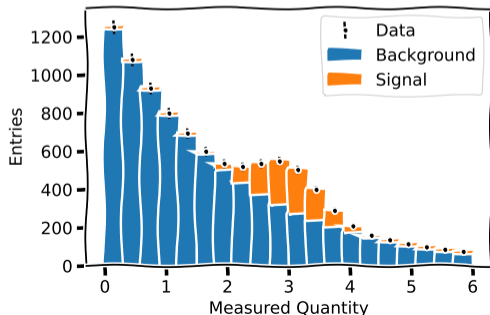


- The Poisson distribution \mathcal{P} , the number of measured events n_i in bin i and the number of expected events ν_i in bin i .
- The expectation $\nu_i = \sum_k^{\text{templates}} f_{ik} \eta_k$, where η_k is the total number of expected events for template k and f_{ik} is the fraction of events expected in bin i of template k .
- The fractions are defined as $f_{ik} = \frac{\eta_{ik}}{\sum_{j=1}^{\text{bins}} \eta_{jk}}$

Asimov Data

A useful tool for developing and testing your fits is the Asimov Data. You substitute the "recorded data" with a data sample which **corresponds your expectation on MC**.

- In comparison to a toy sample, your expectations on the fit result are known. This helps debugging. Toy MC is still a useful test to check your fit e.g. for bias.
- Caveat: You have to round your MC expectation to integer values.

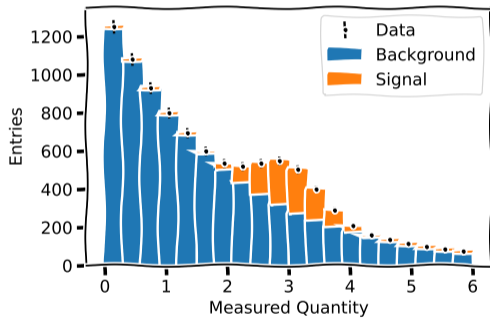


$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i | \nu_i) \quad (6)$$

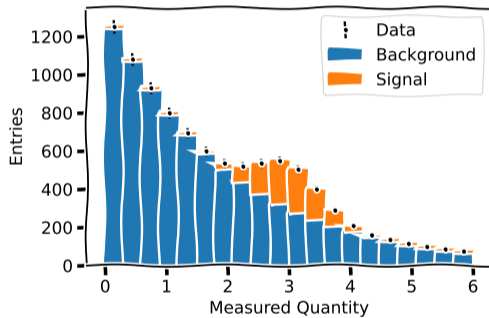
- n_i : data in bin i
- ν_i : background plus signal expectation in bin i

Fit Result

Pre-Fit

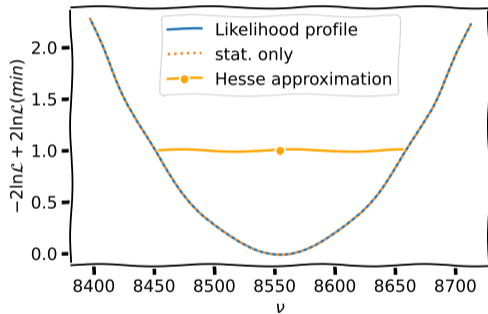


Post-Fit

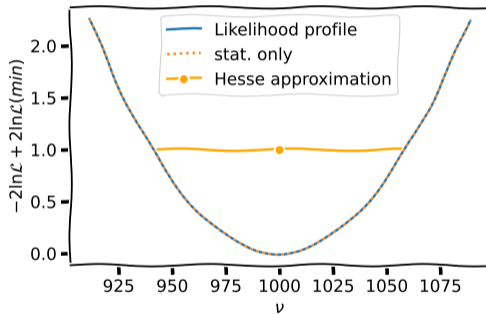


Determining Uncertainty on the Fitted Parameter

ν_{bkg}



ν_{sig}



First Summary

- We minimize the Likelihood function \mathcal{L} to determine our model parameter.
- So far, we did not include any uncertainties on our fit model.
- Uncertainties can arise from different sources:
 - Limited statistics for our templates (we can not produce infinite MC).
 - Theory uncertainties, e.g. uncertainty on a parameter which is used in the event generator.
 - Uncertainties on tracking efficiency or lepton ID.
- In the following we provide **recipes** for including systematic uncertainties into the fit.

Adding Systematics

Bin-Wise Uncertainties: MC Statistics i

The limited MC statistics is an uncorrelated uncertainty for each bin (pure statistical errors are not correlated).

- For each template k , we have a statistical uncertainty σ_i^k in each bin i .
- The correlation matrix Σ is diagonal. $\Sigma^k = \mathbb{I}$,

To include this uncertainty into the fit, we introduce a vector of nuisance parameters $\vec{\theta} = \vec{\theta}_k, k = 1, \dots, n\text{Templates}$ and modify our Likelihood function:

$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i | \nu_i(\vec{\theta})) \times \prod_k^{\text{templates}} \mathcal{N}(\vec{\theta}_k | 0, \Sigma^k) \quad (7)$$

Let us decompose this ...

$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i | \nu_i(\theta)) \times \prod_k^{\text{templates}} \mathcal{N}(\theta | 0, \Sigma^k) \quad (8)$$

- The normal distribution $\mathcal{N}(\theta_k | 0, \Sigma^k)$ ensures that the **nuisance parameters only vary within their Gaussian uncertainty**. We need one Normal distribution and correlation matrix for each template. In general, we can include other constraints, e.g. a Log-Normal, if the pull is only allowed to be positive.
- $\vec{\theta} = \vec{\theta}_k, k = 1, \dots, n\text{Templates}$ (here $k=2$).
- The expectation value has to be modified by $\nu_{ik} \rightarrow \nu_{ik}(\theta)$

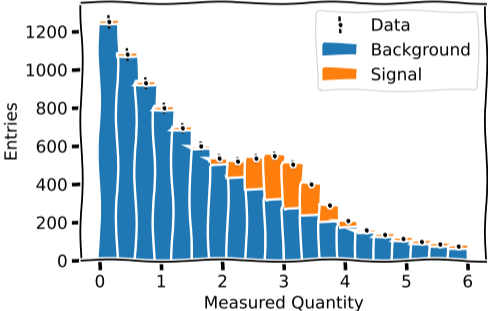


$$\nu_i(\theta) = \sum_k^{\text{templates}} f_{ik}(\theta_k) \eta_k \quad (9)$$

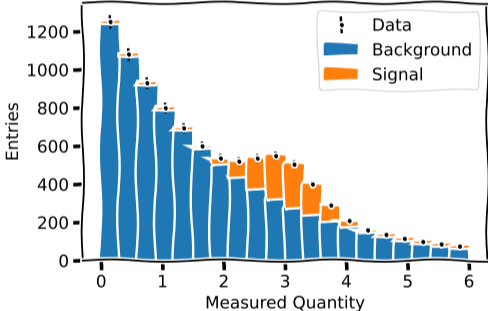
$$f_{ik}(\theta_k) \rightarrow \frac{\eta_{ik}(1 + \theta_{ik}\epsilon_{ik})}{\sum_k^{\text{bins}} \eta_{jk}(1 + \theta_{jk}\epsilon_{jk})} \quad (10)$$

Fit Result

Pre-Fit

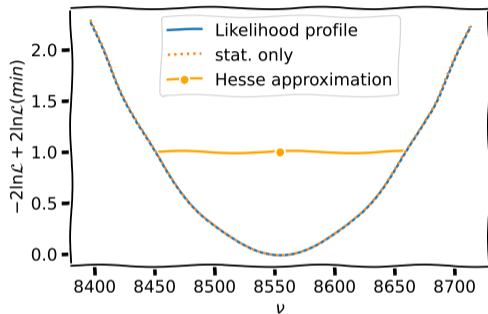


Post-Fit

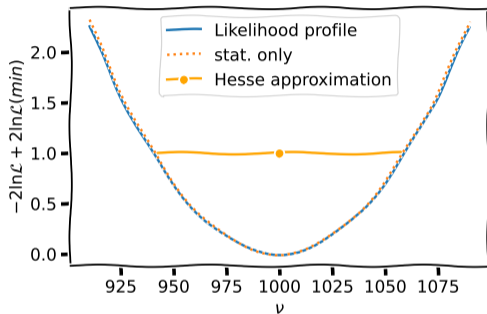


Determining Uncertainty on the Fitted Parameter

ν_{bkg}

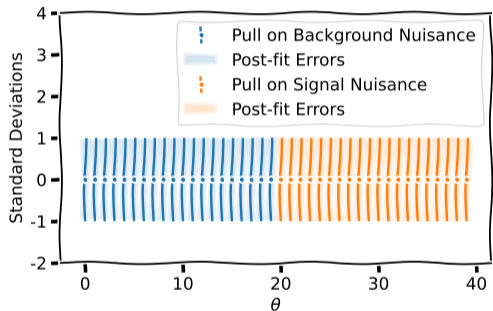


ν_{sig}



- No difference between the full Likelihood profile and statistical only.
- We have a lot of MC statistics, so we do not expect and influence here.

Pulls on Nuisance Parameters



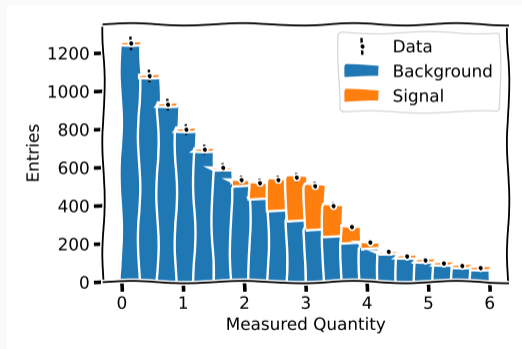
- We fit an Asimov sample, no pulls expected.
- The fit can not constrain any nuisance parameters here.

Our background model is based on an exponential function: $\mathcal{E}(\lambda = 2)$. What if we are not sure about λ , i.e. $\lambda = 2 \pm 0.2$? We incorporate this uncertainty into our background template.

- We have a simple model, which only depends on one model parameter.
- To determine the influence on the template, we vary the model parameter and generate new templates

Bin-Wise Uncertainties: Theory Uncertainty ii

Each bin has an uncertainty σ_i^{model} . To determine the total uncertainty from the model parameters in each bin:



- Diagonalize the covariance matrix, vary the covariance matrix \pm an eigenvalue, transform back.
- Generate new templates with the up and down variation.
- $\sigma_i = \text{sign}(\sigma_i^-)(|\sigma_i^-| + |\sigma_i^+|)/2$
- This uncertainty is 100% correlated.
- The covariance matrix is generated by $C = \sigma_i^{\text{model}} \otimes \sigma_i^{\text{model}}$.

Bin-Wise Uncertainties: Theory Uncertainty iii

We already introduced nuisance parameters per bin before. We now simply modify what we use in the Likelihood for our background template.

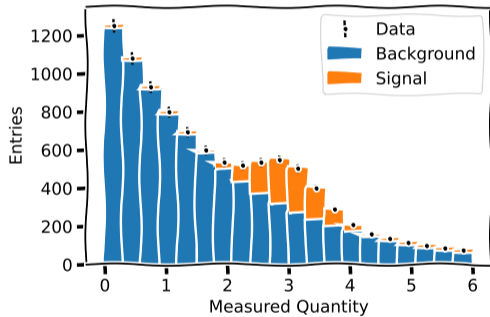
- Add up covariance matrices from different sources $C^{\text{bkg}} = C_{\text{stat}}^{\text{bkg}} + C_{\text{model}}^{\text{bkg}}$
- $\rightarrow \epsilon_{i,\text{bkg}} = \epsilon_{i,\text{bkg}}^{\text{stat}} + \epsilon_{i,\text{bkg}}^{\text{model}}$

Done! We re-run the fit.

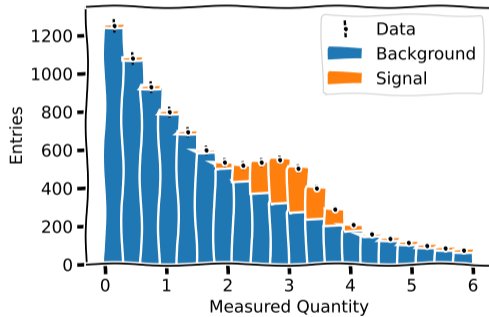
Caveat: You have to invert the covariance matrix in the Likelihood. A covariance matrix with 100% correlation is singular. However, we already added the statistical uncertainties which are uncorrelated, resolving this issue.

Fit Result

Pre-Fit

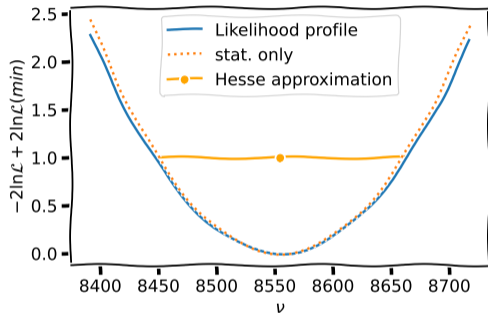


Post-Fit

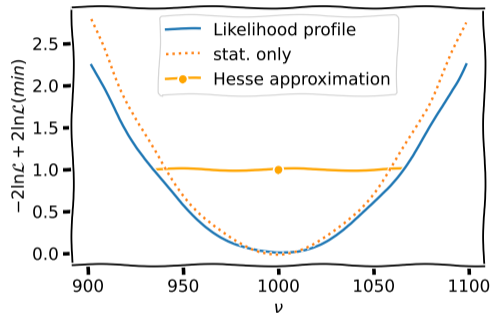


Determining Uncertainty on the Fitted Parameter

ν_{bkg}

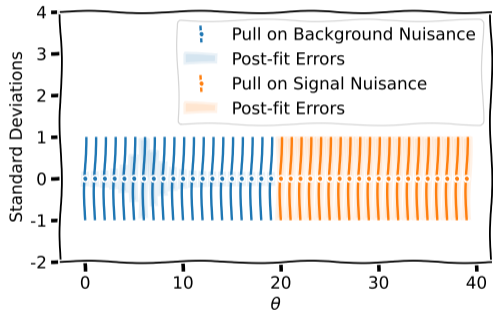


ν_{sig}



- The model uncertainty has an impact on our Likelihood profile (we expect this, the uncertainty is 10%)

Pulls on Nuisance Parameters



- We fit an Asimov sample, no pulls expected.
- The fit is able to constrain the nuisance parameter which describe the model uncertainty.

Source-Wise Uncertainties: Model Uncertainty i

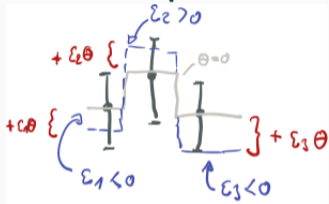
An **alternative approach** to include the uncertainty of our background model $\mathcal{E}(x|\lambda = 2 \pm 0.2)$ is to include it with a single nuisance parameter. If the systematic is correlated, we can add the systematic uncertainty not bin-wise but source wise.

$$\mathcal{L} \rightarrow \mathcal{L} \times \prod_j^{\text{sources}} \mathcal{N}(\theta_j|0, 1), \quad (11)$$

with $\nu_i \rightarrow \nu_i(1 + \epsilon_i\theta)$. This introduces a **single** nuisance parameter per source. The error amplitude has to encode the (anti-)correlation. 100% correlation is implied

Source-Wise Uncertainties: Model Uncertainty ii

Nuisance per source



- Post-fit you can check on which error source the fit tries to pull.

Nuisance per bin



- Post-fit you can check on which bin the fit tries to pull.

Source-Wise Uncertainties: Tracking Uncertainty i

The tracking uncertainty has an impact on the uncertainty of the signal yield. To include this, we add the tracking uncertainty into the Likelihood by transforming

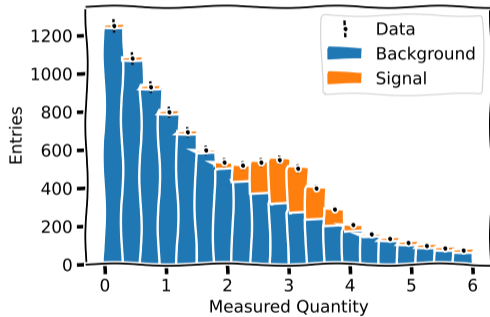
$$\eta_k \rightarrow \eta_k(1 + \epsilon_{\text{tracking}}\theta_{k,\text{tracking}}), \quad k = \text{sig}. \quad (12)$$

- This is a single nuisance parameter in that case.
- We assume it is constant over the range of the measured quantity.
- We can add more than one uncertainty this way:
 $\eta_k \rightarrow \eta_k \prod_m^{\text{uncertainties}} (1 + \epsilon_m \theta_{k,m})$, adding a nuisance parameter for each source.

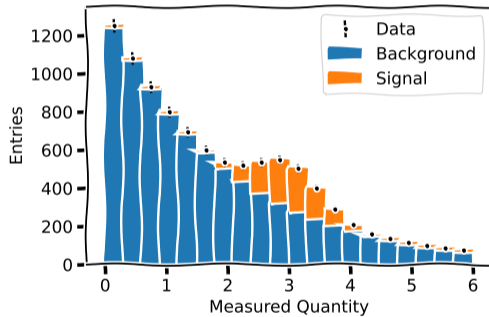
All done! We re-run the fit.

Fit Result

Pre-Fit

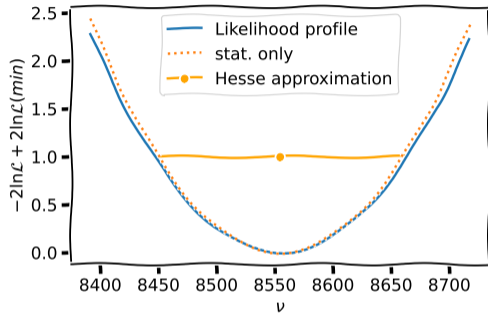


Post-Fit

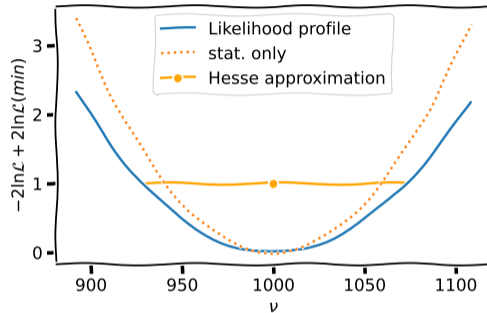


Determining Uncertainty on the Fitted Parameter

ν_{bkg}

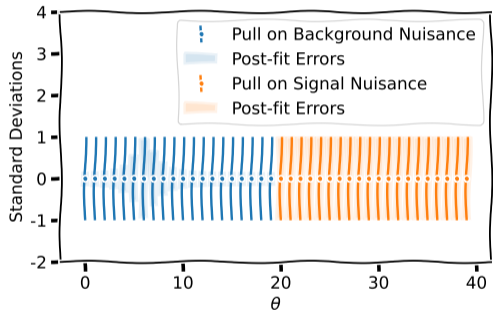


ν_{sig}



- Now the Likelihood gets even broader, as we introduced an additional systematic uncertainty.
- For multiplicative uncertainties, the best fit point will be $\theta_{k,tracking} = 0$, as it is always cheaper to vary ν_{sig} than the constraint nuisance parameter.

Pulls on Nuisance Parameters



- We fit an Asimov sample, no pulls expected.
- The fit is able to constrain the nuisance parameter which describe the model uncertainty.

Validating the Fit Model

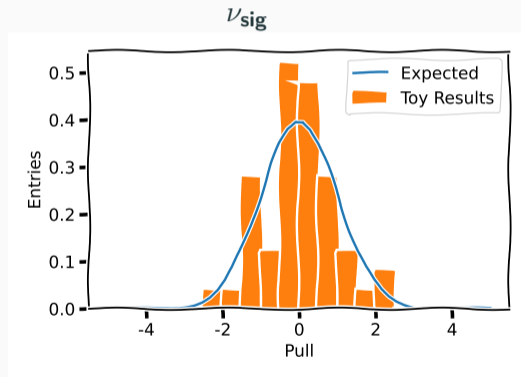
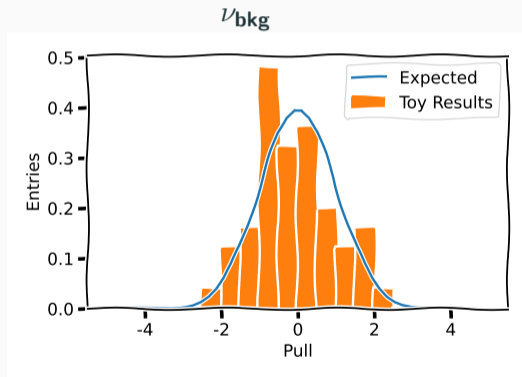
Validating the Fit Model i

So far, we used the Asimov sample to develop our fit. This is nice, because we can compare the result with our expectation. However, it is not sufficient to test our fit for bias. To validate that our fit is **unbiased and stable** we can draw so-called toy samples from our expectation.

- Draw from a Poisson distribution $\mathcal{P}(n|\nu_{\text{expected}}^{\text{total}})$ for each bin.
- We perform the fit for N_{toy} times and check our parameter of interest ν^{sig} via
$$\text{Pull} = \frac{\nu_{\text{fit}}^{\text{sig}} - \nu_{\text{expected}}^{\text{sig}}}{\sigma(\nu_{\text{fit}}^{\text{sig}})}$$
 for each fit result.
- We can do this with every parameter in the model.
- This distribution has to follow a normal distribution with $\mathcal{N}(0, 1)$.

Validating the Fit Model ii

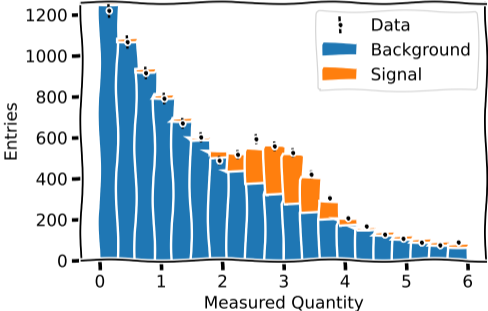
Applying this method to our problem, we find that the fit is stable and unbiased.



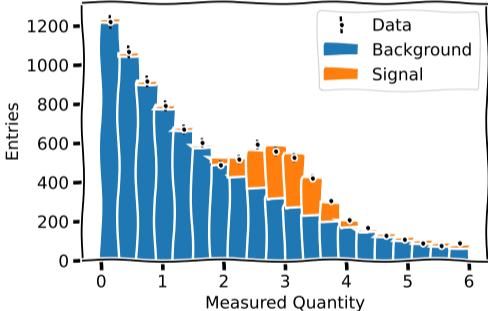
Unleash the Fit on Real Data

Fit Result

Pre-Fit

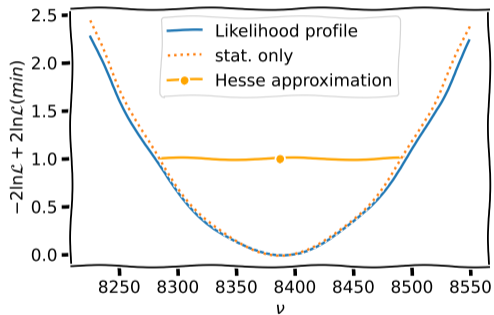


Post-Fit

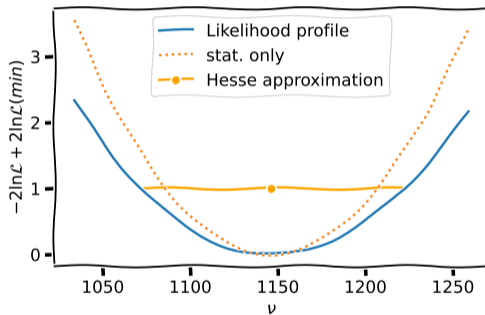


Determining Uncertainty on the Fitted Parameter

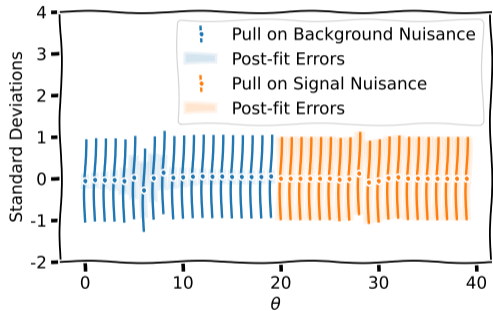
ν_{bkg}



ν_{sig}



Pulls on Nuisance Parameters



- The fit tries to pull on the nuisance parameters.
- The fit is able to constrain the nuisance parameter which describe the model uncertainty.

We discussed how to incorporate different systematics in different ways into the Likelihood. However, what was presented are **recipes**. **Choosing the right recipe for your problem is part of the work.**

- It depends on the complexity of your problem. Bin-wise systematics can explode for higher dimensions, e.g. 2D templates.
- It depends on what information you want to have access to post-fit, e.g. do you want to see the pulls per bin or per error source?
- The presented methods are **not** an exhaustive list of methods, but can be applied in many instances.

Backup

Including Bin-Wise Systematic Uncertainties

$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i | \nu_i(\theta)) \times \prod_k^{\text{templates}} \mathcal{N}(\theta | 0, \Sigma^k) \quad (13)$$

is equivalent to

$$\mathcal{L} = \prod_{i=1}^{\text{bins}} \mathcal{P}(n_i | \nu_i(\theta)) \times \prod_k^{\text{templates}} \mathcal{N}(\theta | 0, C^k) \quad (14)$$

when we use the transformation

$$\nu_i = \sum_k^{\text{templates}} f_{ik}(\theta_k) \eta_k \quad (15)$$

$$f_{ik} \rightarrow \frac{\eta_{ik}(1 + \theta_{ik})}{\sum_k^{\text{bins}} \eta_{jk}(1 + \theta_{jk})} \quad (16)$$

instead.

Adding Multiplicative Uncertainties

Adding the uncertainty via

$$\eta_k \rightarrow \eta_k(1 + \epsilon_{\text{tracking}}\theta_{k,\text{tracking}}), \quad k = \text{sig} \quad (17)$$

is equivalent to adding the uncertainty after the fit to the signal yield:

$$\nu_{\text{sig}} = \mu_{\text{fit}} \pm (\sigma_{\text{fit}} \oplus \sigma_{\text{tracking}}) \quad (18)$$

This works because the fit can not pull on the nuisance parameter because of the penalty term. It is always cheaper to vary the free parameter ν_{sig} .

A useful cross-check is, if the nuisance parameter for the tracking error is 0 after the fit at the best-fit point.