

Astropy_fitting

November 18, 2024

1 Astropy models and fitting

If you need to do least square fitting for data to a model a good place to start is **astropy**'s modeling and fitting code.

1.1 Packages being used

- **astropy**: for modeling and fitting
- **matplotlib**: for plotting

1.2 Relevant documentation

- **astropy**: <http://docs.astropy.org/en/stable/modeling/index.html>

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from astropy.modeling import models, fitting
import mpl_style
%matplotlib inline
plt.style.use(mpl_style.style1)
```

1.3 1-D model fitting

For an example lets look at the problem of fitting a 1-D model to a spectral line. First we need to create some fake data:

```
[2]: x = np.linspace(-5., 5., 200)
y = 3 * np.exp(-0.5 * (x - 1.3)**2 / 0.8**2)
y += np.random.normal(0., 0.2, x.shape)
```

1.3.1 A trapezoid model

```
[3]: t_init = models.Trapezoid1D(amplitude=1.0, x_0=0.1, slope=0.5)
fit_t = fitting.LevMarLSQFitter()
t = fit_t(t_init, x, y)
print(t)
```

Model: Trapezoid1D
Inputs: ('x',)

Outputs: ('y',)

Model set size: 1

Parameters:

amplitude	x_0	width	slope
2.989104244502686	1.2958859390176691	0.44447017265771593	1.9357634226923799

1.3.2 A Gaussian model

```
[4]: g_init = models.Gaussian1D(amplitude=1., mean=0, stddev=1.)  
fit_g = fitting.LevMarLSQFitter()  
g = fit_g(g_init, x, y)  
print(g)
```

Model: Gaussian1D

Inputs: ('x',)

Outputs: ('y',)

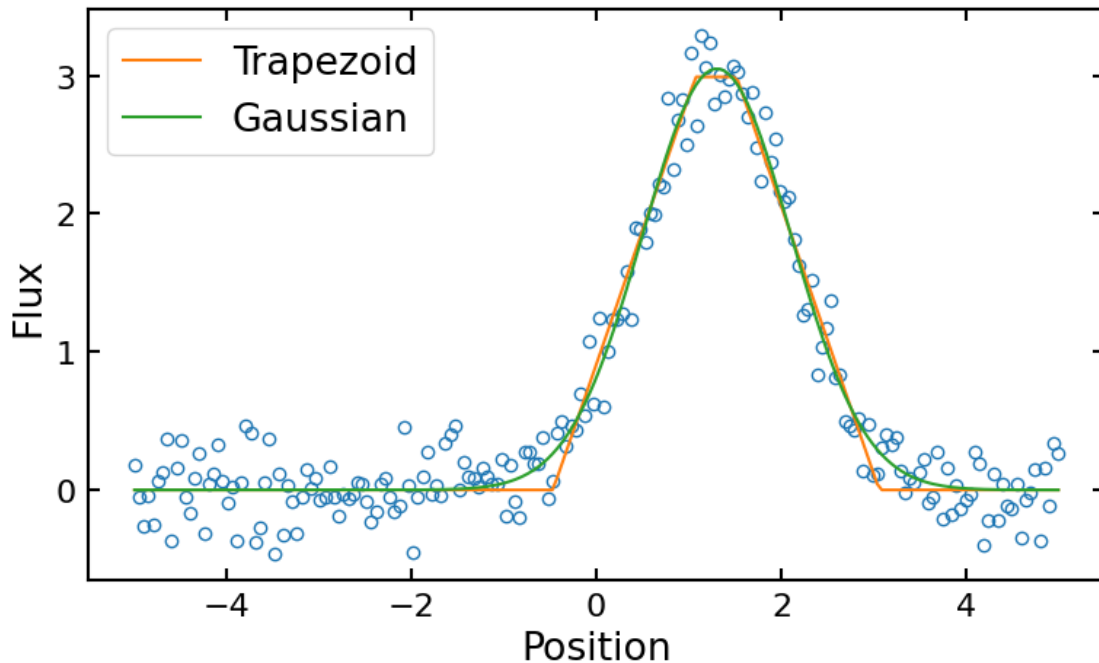
Model set size: 1

Parameters:

amplitude	mean	stddev
3.0484580762232105	1.3032931707955397	0.8022073529057546

1.3.3 Plotting the results

```
[5]: plt.figure(1, figsize=(8,5))  
plt.plot(x, y, 'o', mfc='none')  
plt.plot(x, t(x), label='Trapezoid')  
plt.plot(x, g(x), label='Gaussian')  
plt.xlabel('Position')  
plt.ylabel('Flux')  
plt.legend(loc=2)  
plt.tight_layout()
```



1.4 Compound models

Models can also be ‘added’ together before fitting. To demonstrate let's make a new dataset made up to two Gaussians.

```
[6]: np.random.seed(42)
g1 = models.Gaussian1D(1, 0, 0.2)
g2 = models.Gaussian1D(2.5, 0.5, 0.1)
x = np.linspace(-1, 1, 200)
y = g1(x) + g2(x) + np.random.normal(0., 0.2, x.shape)
```

1.4.1 Make the model

The model can be ‘added’ just like arrays:

```
[7]: gg_init = models.Gaussian1D(1, 0, 0.1) + models.Gaussian1D(2, 0.5, 0.1)
fit_gg = fitting.SLSQPLSQFitter()
gg = fit_gg(gg_init, x, y)
print(gg)
```

```
Optimization terminated successfully      (Exit mode 0)
      Current function value: 6.832859352651305
      Iterations: 15
      Function evaluations: 128
      Gradient evaluations: 15
```

```
Model: CompoundModel
```

```

Inputs: ('x',)
Outputs: ('y',)
Model set size: 1
Expression: [0] + [1]
Components:
  [0]: <Gaussian1D(amplitude=0.98118974, mean=0.00597661, stddev=0.20738563)>

  [1]: <Gaussian1D(amplitude=2.48566491, mean=0.5041208, stddev=0.10000035)>
Parameters:
      amplitude_0      mean_0      ...      stddev_1
-----
0.9811897354569954 0.005976606880260532 ... 0.10000035347831682

/Users/coleman/anaconda3/envs/lensing/lib/python3.10/site-
packages/scipy/optimize/_optimize.py:404: RuntimeWarning: Values in x were
outside bounds during a minimize step, clipping to bounds
  warnings.warn("Values in x were outside bounds during a ")

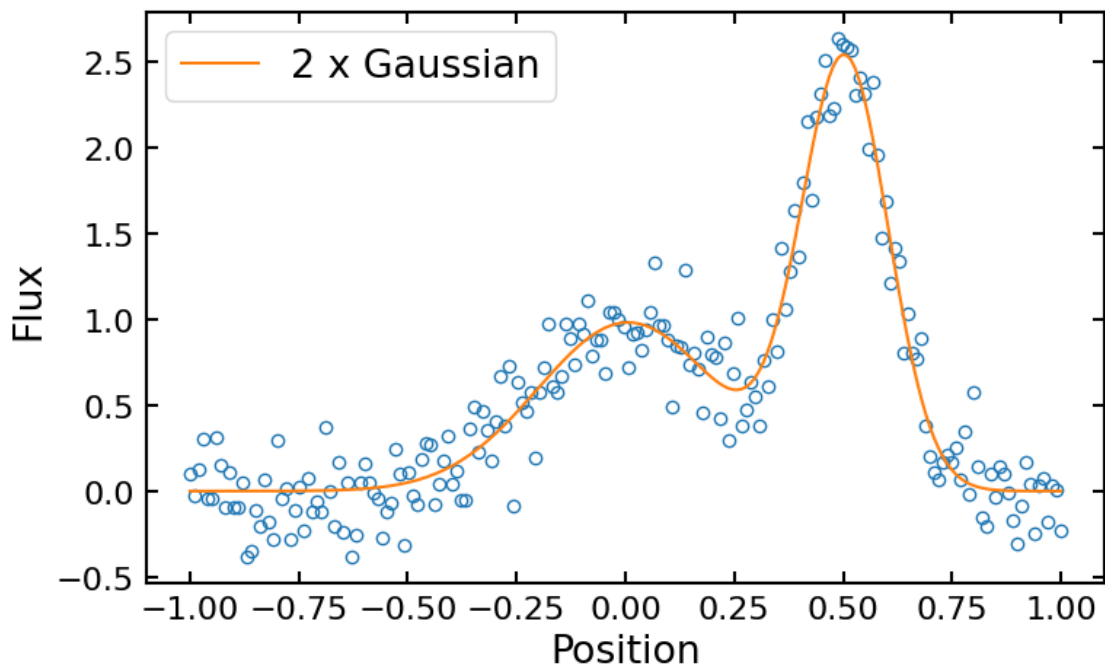
```

1.4.2 Plot the result

```

[8]: plt.figure(2, figsize=(8, 5))
plt.plot(x, y, 'o', mfc='none')
plt.plot(x, gg(x), label='2 x Gaussian')
plt.xlabel('Position')
plt.ylabel('Flux')
plt.legend(loc=2)
plt.tight_layout()

```



1.5 Astropy's models

Astropy has a large number of 1- and 2-D models built in. Check out https://docs.astropy.org/en/stable/modeling/index.html#module-astropy.modeling.functional_models for a full list. If the model you are looking for is not built in, you can always define your own: <https://docs.astropy.org/en/stable/modeling/new-model.html>.

1.6 Limitations

- Uses OLS (or similar) to maximize an objective function (and all the assumptions about the data that go into this, e.g. Gaussian errors)
- Cov of fit only returned for some fitters (found on the `fitter.fit_info()` method)

[]: