notebook_09

December 19, 2018

1 Aufgabe 25

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        from numpy.polynomial.polynomial import polyval
        import uncertainties.unumpy as unp
        from uncertainties.unumpy import nominal_values as noms
        from uncertainties.unumpy import std_devs as stds
        import pandas as pd
```

a) Bestimme die Parameter mit der Methoder der kleinsten Quadrate:

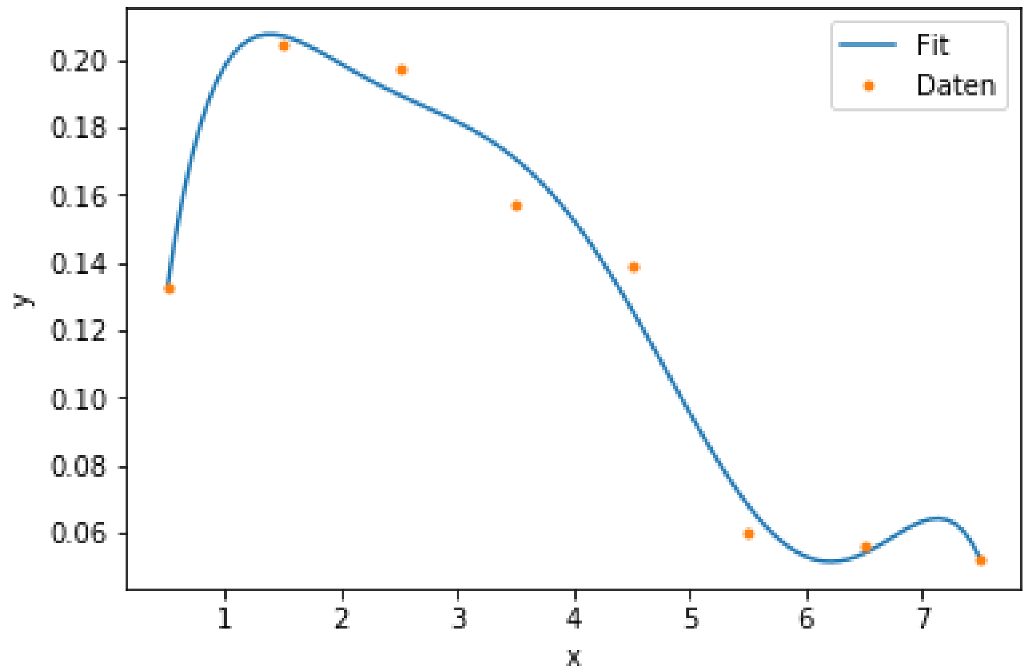
```
In [2]: #read data
        x, y = np.genfromtxt('aufg_a.csv', delimiter = ',', unpack = True)

        #design matrix
        A = np.array([x**i for i in range(7)]).T

        #parameters with least square
        best_a = np.linalg.inv(A.T @ A) @ A.T @ y

        for i in range(7):
            print(f'a_{i} ~ {best_a[i]:.4f}')
```

```
a_0 ~ -0.0674
a_1 ~ 0.6096
a_2 ~ -0.5137
a_3 ~ 0.2106
a_4 ~ -0.0452
a_5 ~ 0.0048
a_6 ~ -0.0002
```

Stelle das Ergebnis graphisch dar:

```
In [3]: xplot = np.linspace(x[0], x[-1], 100)
        plt.plot(xplot, polyval(xplot, best_a), label = 'Fit')
        plt.plot(x, y, '.', label = 'Daten')
```

```python
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



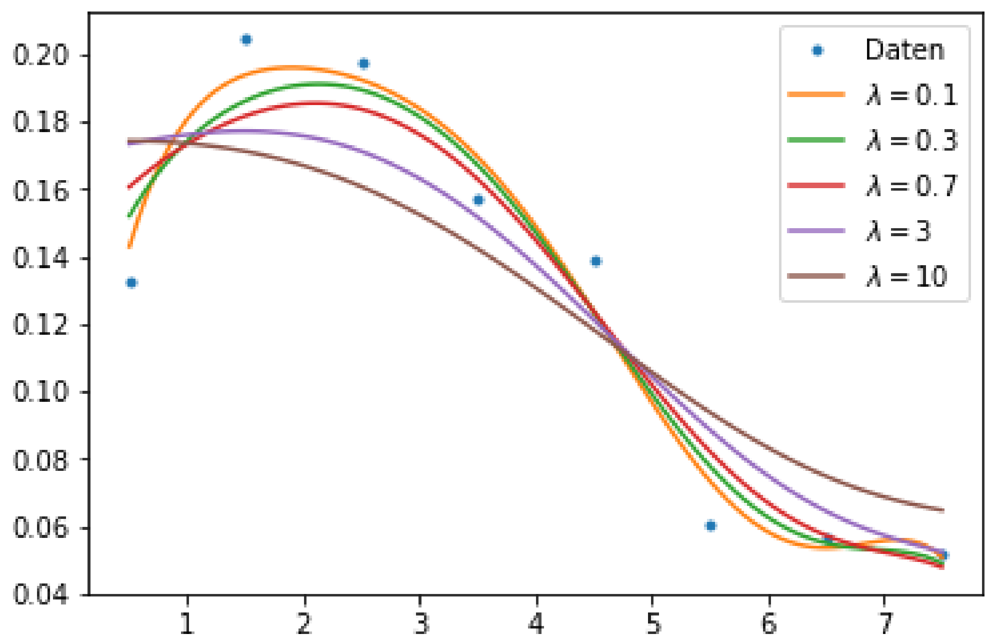
b) Erstelle zunächst die Matrix C, mit der die numerische zweite Ableitung bestimmt wird:

```python
In [4]: C = np.zeros((np.shape(A)[0], np.shape(A)[0]))
        np.fill_diagonal(C, -2)
        np.fill_diagonal(C[1:], 1)
        np.fill_diagonal(C[:, 1:], 1)
        C[0, 0] = -1
        C[-1, -1] = -1
        C
```

```
Out[4]: array([[-1.,  1.,  0.,  0.,  0.,  0.,  0.,  0.],
               [ 1., -2.,  1.,  0.,  0.,  0.,  0.,  0.],
               [ 0.,  1., -2.,  1.,  0.,  0.,  0.,  0.],
               [ 0.,  0.,  1., -2.,  1.,  0.,  0.,  0.],
               [ 0.,  0.,  0.,  1., -2.,  1.,  0.,  0.],
               [ 0.,  0.,  0.,  0.,  1., -2.,  1.,  0.],
               [ 0.,  0.,  0.,  0.,  0.,  1., -2.,  1.],
               [ 0.,  0.,  0.,  0.,  0.,  0.,  1., -1.]])
```

Gibt es dafür eine fertige Methode?
Stelle die Ergebnisse der Regularisierung für verschiedene λ dar:

```
In [5]: plt.plot(x, y, '.', label = 'Daten')

        for lam in [0.1, 0.3, 0.7, 3, 10]:
            gamma = np.sqrt(lam) * C @ A
            best_a_reg = np.linalg.inv(A.T @ A + gamma.T @ gamma) @ A.T @ y
            plt.plot(xplot, polyval(xplot, best_a_reg),
                     label = f'$\lambda = {lam}$')

        plt.legend()
        plt.show()
```

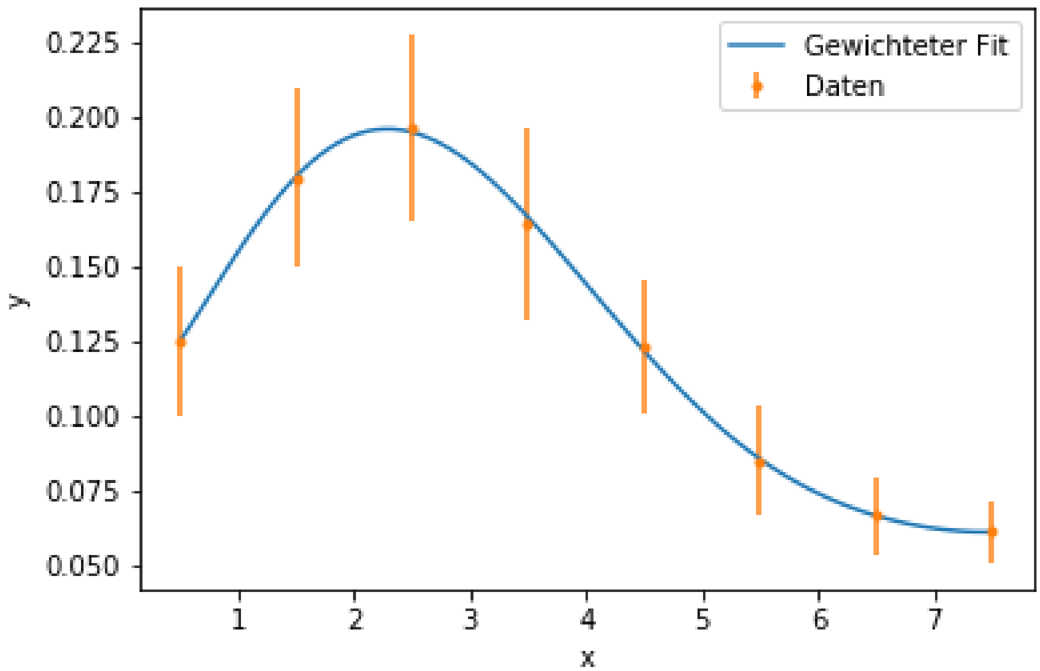


c)

```
In [6]: #read data
        data = pd.read_csv('aufg_c.csv')
        x = data['x']

        #calculate mean and error for y
        y = unp.uarray(data.drop(columns = 'x').T.mean(),
                       data.drop(columns = 'x').T.std())

        #weight matrix
        W = np.zeros((np.shape(A)[0], np.shape(A)[0]))
        np.fill_diagonal(W, 1 / stds(y)**2)

In [7]: #calculate parameters
        best_a_weight = np.linalg.inv(A.T @ W @ A) @ A.T @ W @ noms(y)
```

Stelle Ergebnisse in einem Plot dar:

```
In [8]: plt.plot(xplot, polyval(xplot, best_a_weight), label = 'Gewichteter Fit')
        plt.errorbar(x = x, y = noms(y),
                     yerr = stds(y),
                     label = 'Daten', marker = '.', linestyle = '')
        plt.xlabel('x')
        plt.ylabel('y')
        plt.legend()
        plt.show()
```



Nitschke
Grisard

Aufgabe 27:

a) Likelihood allgemein:

$$\mathcal{L}(\vec a) = f(\vec x_1 | \vec a) \cdot \ldots \cdot f(\vec x_n , \vec a) = \prod_{i=1}^{n} f(\vec x_i | \vec a)$$

hier 1D:

$$\mathcal{L}(b) = \prod_{i=1}^{n} f(x_i | b)$$

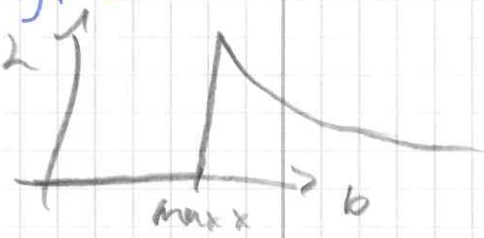
und
$$f(x) = \begin{cases} \frac{1}{b} , & x \in [0, b] \\ 0 , & \text{sonst} \end{cases}$$

für ein gegebenes Sample ist die Likelihood also 0, wenn b so gewählt wird, dass min. 1 x_i nicht in $[0, b]$ liegt.

Wenn $b \geq x_i \quad \forall i \in \{1, \ldots, n\}$, dann

$f(x_i | b) = \frac{1}{b}$ und

$$\mathcal{L}(b) = \frac{1}{b^n}$$



$\frac{1}{b^n}$ fällt monoton mit b, daher ist

$\mathcal{L}(\max_{x \in \{x_i\}} x)$ maximal und der beste Schätzer ist $b = \max_{x \in \{x_i\}} (x)$

b)

$$E\left[\max_{x \in \{x_i\}} x\right] = \int_0^{\max_{x \in \{x_i\}}(x)} \frac{1}{\max_{x \in \{x_i\}}(x)} \, x \, dx = \frac{\max_{x \in \{x_i\}}(x)}{2}$$

Für Erwartungstreue muss ‚Erwartungswert vom Schätzer = Schätzer' sein, wähle daher

$$b^* = 2 \cdot \max_{x \in \{x_i\}} (x)$$