A1	A2	A3	A4	Σ
5.75P	8P.	–	–	13.7

SMD-Übungsblatt 10

Abgabe: 10.01.19

Yvonne Kasper yvonne.kasper@udo.edu ,
Robert Appel robert.appel@udo.edu ,
Julian Schröer julian.schroeer@udo.edu

1 Aufgabe1

Aufgabe 1 wurde handschriftlich abgegeben.

2 Aufgabe 2

a A beschreibt Messprozess der die verschmierung in Nachbar-Bin mit Wahrscheinlichkeit ϵ beschreibt, z.B. Klassifizierung von Ereignissen in mehere Klassen. Ergebnis der geschriebenen Methode für $n = 4$ und $\epsilon = 0.23$:

$$\begin{pmatrix} 0.77 & 0.23 & 0. & 0. \\ 0.23 & 0.54 & 0.23 & 0. \\ 0. & 0.23 & 0.54 & 0.23 \\ 0. & 0. & 0.23 & 0.77 \end{pmatrix} \tag{1}$$

2P.

b gmess er gibt sich zu:

$$g_{mess} = [254, 474, 616, 758, 826, 770, 759, 729, 691, 610, 563, 487, 459, 407, 341, 318, 247, 223, 194, 181] \tag{2}$$

1P.

c Mit Transformation von $A = UDU^{-1}$ und U transformationsmatrix und D Diagonalmatrix folgt

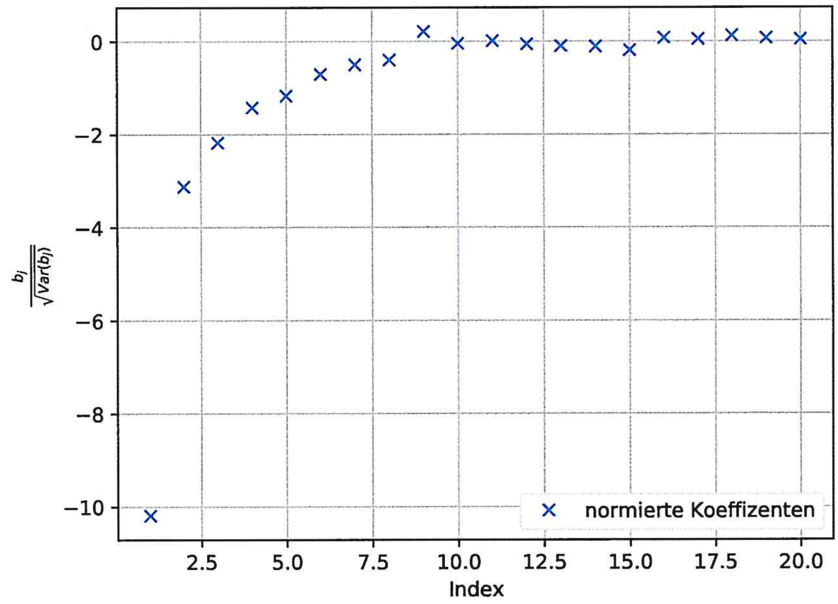
$$g = A \cdot f = UDU^{-1} \cdot f \tag{3}$$
$$\Longleftrightarrow \underbrace{U^{-1}g}_{=c} = D\underbrace{U^{-1} \cdot f}_{=b} \tag{4}$$
$$\Longleftrightarrow c = D \cdot b\,. \tag{5}$$ ✓

Vorteil: Die Einträge b_j und c_j können unabhängig transfomiert werden, da D Diagonalmatrix. ✓ 2P.

d Aus den Gleichungen aus **b** kann man dann f und b leicht durch Matrixmultiplikation brechnen. Die Kovarianz $V[b]$ von $b = D^{-1}U^{-1}g_{mess}$ lässt sich berechnen mit $B = D^{-1}U^{-1}$ und $V[b] = BV[g_{mess}]B^T$. Die Koeffizienten normiert auf ihre Standardabweichung sind in der Abbildung 1 dargestellt. Alle Werte die um 1 liegen haben eine große Abweichung in der Größenordnung des Wertes. Folglich sind dies die Werte, die verstärkt von statistischen Fluktuationen betroffen sind und somit kommt es dann zu Oszillationen. ✓

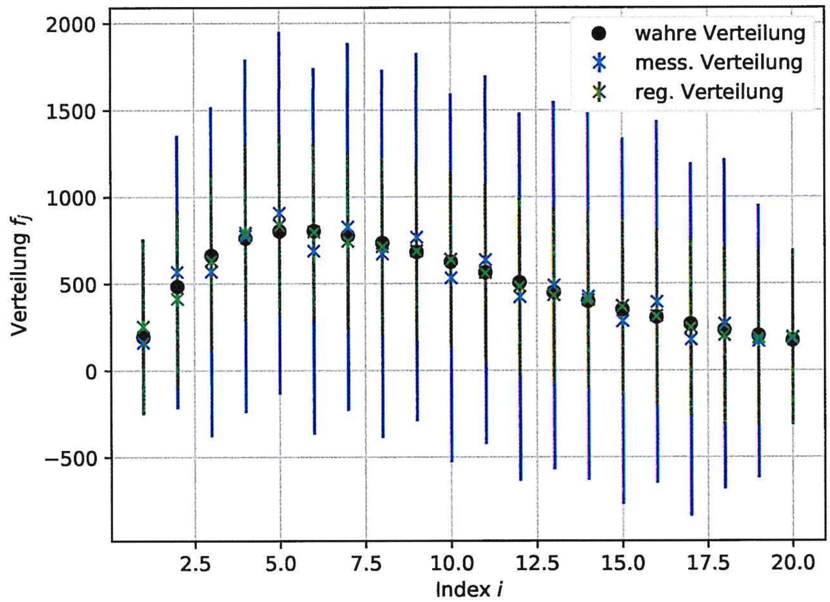
e Die Werte mit Regularisierung sind nähr an der wahren Verteilung dran und zeigen auch kleine Fehler. Siehe dazu Abbildung 2.

siehe code...



1.5P.

Abbildung 1: Einträge von b normiert auf ihre Standardabweichung gegen ihren Index aufgetragen.



Fehler falsch. bei unreg. verteilung ist es der Folgefehler

Abbildung 2: Wahre, gemessene und die regularisierte Verteilung gegen den Index Aufgetragen.

1.5P

8/9P.

SMD Blatt 10 Appel, Kasper, Schröer

für Alicia und Simone

Aufgabe 28

a) $$\underline{\underline{A}} = 0{,}8 \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$ ✓

Faktor 0,8, da nur 80% der Messungen überleben

$A_{11} = A_{22} = 1-\epsilon$, da dies die Wahrscheinlichkeit ist, die Ereignisse richtig einzuordnen.

$A_{12} = A_{21} = \epsilon$, da diese Elemente für die falsche Einordnung stehen

Zusammenhang: $\vec{g} = \underline{\underline{A}}\,\vec{f}$

Messung, Antwort-Matrix, wahre Ereigniszahl

$$\Rightarrow \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \begin{pmatrix} 0{,}8(1-\epsilon) & 0{,}8\,\epsilon \\ 0{,}8\,\epsilon & 0{,}8(1-\epsilon) \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

$$= \begin{pmatrix} 0{,}8\left((1-\epsilon)f_1 + \epsilon f_2\right) \\ 0{,}8\left(\epsilon f_1 + (1-\epsilon)f_2\right) \end{pmatrix}$$ ✓

1,5 P.

b) Lösung durch Inversenbildung

$\vec{g} = \underline{\underline{A}}\vec{f} \Rightarrow \vec{f} = \underline{\underline{A}}^{-1}\vec{g}$

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{pmatrix}$$

$$= \frac{0{,}8}{0.8^2\left((1-\epsilon)^2 - \epsilon^2\right)} \begin{pmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{pmatrix}$$

$$= \frac{1}{0.8\left(1-2\epsilon+\epsilon^2-\epsilon^2\right)} \begin{pmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{pmatrix}$$

$$= \frac{10}{8}\,\frac{1}{(1-2\epsilon)} \begin{pmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{pmatrix}$$

$\vec{f} = \ldots ?$

0.5 P.

c) Kovarianzmatrix

$$V[\vec{f}] = \underline{\underline{A}}^{-1}\, V[\vec{g}]\, (\underline{\underline{A}}^{-1})^T$$

$V[\vec{g}]$ ist diagonal, da die gemessenen Ereigniszahlen unkorreliert sind.

$$V[\vec{g}] = \begin{pmatrix} \sigma_{g_1}^2 & 0 \\ 0 & \sigma_{g_2}^2 \end{pmatrix} \checkmark$$

Da $\vec{g}$ Poisson-Verteilt sind entspricht die Varianz dem Erwartungswert.

$$V[\vec{g}] = \begin{pmatrix} g_1 & 0 \\ 0 & g_2 \end{pmatrix} \checkmark$$

$$V[\vec{f}] = \frac{1}{(0.8(1-2\epsilon))^2} \begin{pmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{pmatrix} \begin{pmatrix} g_1 & 0 \\ 0 & g_2 \end{pmatrix} \begin{pmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{pmatrix}$$

$$= \frac{1}{0{,}64(1-2\epsilon)^2} \begin{pmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{pmatrix} \begin{pmatrix} (1-\epsilon)g_1 & (-\epsilon)g_1 \\ (-\epsilon)g_2 & (1-\epsilon)g_2 \end{pmatrix}$$

$$= \frac{1}{0{,}64(1-2\epsilon)^2} \begin{pmatrix} (1-\epsilon)^2 g_1 + \epsilon^2 g_2 & (1-\epsilon)(-\epsilon)(g_1+g_2) \\ (-\epsilon)(1-\epsilon)(g_1+g_2) & \epsilon^2 g_1 + (1-\epsilon)^2 g_2 \end{pmatrix} \checkmark$$

1P.

d) $\vec{g} = \begin{pmatrix} 200 \\ 169 \end{pmatrix}$ $\quad \epsilon = 0{,}1 \quad \underline{A}^{-1} = \frac{10}{8} \frac{1}{(1-0.2)} \begin{pmatrix} 1-0.1 & -0.1 \\ -0.1 & 1-0.1 \end{pmatrix}$

$$\vec{f} = \frac{25}{16} \begin{pmatrix} 0.9 & -0.1 \\ -0.1 & 0.9 \end{pmatrix} \begin{pmatrix} 200 \\ 169 \end{pmatrix} = \frac{25}{16} \begin{pmatrix} 1631/10 \\ 1321/10 \end{pmatrix} = \begin{pmatrix} \frac{8155}{32} \\ \frac{6605}{32} \end{pmatrix} \approx \begin{pmatrix} 254{,}8 \\ 206{,}4 \end{pmatrix} \checkmark$$

$$V[\vec{f}] = \frac{625}{256} \begin{pmatrix} 163{,}69 & -33{,}21 \\ -33{,}21 & 138{,}89 \end{pmatrix} = \begin{pmatrix} 399.63 & -81.08 \\ -81.08 & 339.09 \end{pmatrix} (\checkmark)$$

$$\sigma_{f_1} = \sqrt{\frac{625}{256} \cdot 163{,}69} \approx 20{,}00 \checkmark$$

$$\sigma_{f_2} = \sqrt{\frac{625}{256} \cdot 138{,}89} \approx 18{,}41 \checkmark$$

$$\rho(f_1, f_2) = \frac{\mathrm{cov}(f_1, f_2)}{\sigma_{f_1} \sigma_{f_2}} \approx \frac{\frac{256}{625} \cdot (-33{,}21)}{8{,}19 \cdot 7{,}54} \approx -0{,}22 \checkmark$$

2P.

e) $\epsilon = 0{,}4 \quad \vec{f} = \begin{pmatrix} 109{,}16 \\ 44{,}58 \end{pmatrix}$, $\quad V[\vec{f}] = \frac{625}{16} \begin{pmatrix} 99{,}04 & -88{,}56 \\ -88{,}56 & 91{,}84 \end{pmatrix} = \ldots$

$\sigma_{f_1} \approx 62{,}20$ ✓, $\sigma_{f_2} \approx 60{,}22$ ✓, $\rho(f_1, f_2) \approx -0{,}92$ ✓

→ Die Werte von $\vec{f}$ haben größer Fehler und sind stärker korreliert, da die Wahrscheinlichkeit der falschen Zuordnung größer geworden ist.

0.5P.

f) $\epsilon = 0{,}5$, Hier lässt sich die Inversionsmethode nicht mehr Anwenden, da die Determinante von A Null wird, und sich die Inverse nicht bilden lässt.
A ist also singulär (?)

0.25P.

Code fuer Blatt10

Kasper, Appel, Schroeer

11. Januar 2019

../B/1/Blatt10_Kasper_Appel_Schroeer/Blatt10.py

```
def Aufgabe2():
    f = np.array([193, 485, 664, 763, 804, 805, 779, 736, 684, 626,566, 508, 452, 400, 351, 308,
    268, 233, 202, 173])
    def antwortmatrix(n,epsi):
        A = np.matrix(np.zeros((n,n)))
        np.fill_diagonal(A,1-2*epsi)
        A[(0,0)]+=epsi
        A[(n-1,n-1)]+=epsi
        for i in range(0,n):
            if i-1 >= 0:
                A[i,i-1] = epsi
            if i+1 <= n-1:
                A[i,i+1] = epsi
        return A

    def diago(Ant):
        ew,ev = np.linalg.eig(Ant)
        ewindis = np.argsort(ew)
        ewindis = ewindis[::-1]
        ew = [ew[i] for i in ewindis]
        tmatrix = ev.T[ewindis]
        tmatrix = tmatrix.T
        diago = np.matrix(np.zeros((Ant.shape)))
        np.fill_diagonal(diago,ew)
        return tmatrix, diago
    #print('A mit n=4 und e=0.23:',antwortmatrix(4,0.23))
    A = antwortmatrix(20,0.23)
    g = np.dot(A,f)
    np.random.seed(42)
    gmess = [np.random.poisson(i) for i in g]
    gmess = np.squeeze(np.array(gmess))
    #print('gmess:', gmess)
    Atrans, Adiag = diago(A)
    c = np.dot(Atrans.T,g.T)
    b = np.dot(Atrans.T,f.T)
    #print('bwahr:', b)
    #print('c:', c)
    ##Kovm von b  mit V[b] = B V[f] B.T mit B = Atrans.I
    Vb = np.dot(np.dot(Atrans.T,np.cov(f)), (Atrans.I).T)
    ###
    bmess = np.dot(np.dot(Adiag.I,Atrans.T),gmess)      -> diag (gmess)
    B = np.dot(Adiag.I,Atrans.T)
    Vbmess = np.dot(np.dot(B,np.cov(gmess)), B.T)
    stdbmess = np.sqrt(np.diagonal(Vbmess))
    nbmess = np.array(bmess/stdbmess)      -> abs( )
    nbmess = np.squeeze(nbmess)

    plt.plot(range(1,21),nbmess, 'bx', label = 'normierte Koeffizenten')
    plt.xlabel(r'Index')
    plt.ylabel(r'$\frac{b_j}{\sqrt{Var(b_j )}}$')      logscale
    plt.grid()
    plt.legend(loc='best')
    plt.savefig('bjggIndexplot.pdf')
    plt.clf()

    fmess = np.dot(Atrans,bmess.T)
    breg = np.array(bmess)
```

```
    np.put(breg,range(8,20),np.zeros(12))
    #print('breg:',breg)
    freg = np.dot(Atrans,breg.T)
    Vfmess = np.dot(np.dot(Atrans,Vbmess) , Atrans.T)
    Vfreg = np.dot(np.dot(Atrans,np.cov(breg)) , Atrans.T)   f.
    stdfmess = np.sqrt(np.diagonal(Vfmess))
    stdfreg = np.sqrt(np.diagonal(Vfreg)) → ff.

    plt.plot(range(1,21),f,'ko', label='wahre Verteilung')
    plt.errorbar(range(1,21),fmess,yerr = stdfmess,fmt='bx', label='mess. Verteilung')
    plt.errorbar(range(1,21),freg,yerr = stdfreg, fmt='gx', label='reg. Verteilung')
    plt.xlabel(r'Index $i$')
    plt.ylabel(r'Verteilung $f_j$')
    plt.legend(loc='best')
    plt.grid()
    plt.savefig('regplot.pdf')


if __name__ == '__main__':
    import matplotlib.pyplot as plt
    import numpy as np
    import uncertainties.unumpy as unp
    import scipy.constants as const
    from scipy.optimize import curve_fit
    from uncertainties import correlated_values, correlation_matrix
    from uncertainties import ufloat
    from uncertainties.unumpy import (nominal_values as noms, std_devs as stds)

    Aufgabe2()
```