Kasper; Appel; Schröer   Die Gewächse

A1	A2	A3	Σ
4	3	5.5	12.5

A19|

a) $c_i^{(t+1)} = \sum_{x_j \in S_i^{(t)}} (x_j) \cdot \frac{1}{|S_i^{(t)}|}$

$t = 1$

$S_1^1 = \{(1;6)\} \qquad |S_1^1| = 1$

$S_2^1 = \{(1;5), (1,4), (3;3), (3;2), (4;1)\} \qquad |S_2^1| = 5$

$S_3^1 = \{(5;1), (6;2), (6;3), (8;4), (8;5), (8;6)\} \qquad |S_3^1| = 6$

$c_1^2 = \frac{1}{1} \cdot (1;6) = (1;6)$

$c_2^1 = \frac{1}{5} \sum_{j=1}^{5} x_j \qquad x_j \in S_2^1$

$= \frac{1}{5} [(1;5) + (1;4) + (3;3) + (3;2) + (4;1)]$

$= (\frac{12}{5} ; 3)$

$c_3^2 = \frac{1}{6} (41;21) = (\frac{41}{6} ; \frac{21}{6}) = (\frac{41}{6} ; \frac{7}{2})$

10P.

b) $t = 2$

$S_1^2 = \{(1;6), (1;5)\} \qquad |S_1^2| = 2$

$S_2^2 = \{(1;4), (3;3), (3;2), (4;1)\} \qquad S_2^2 = 4$

$S_3^2 = \{(5;1), (6;2), (6;3), (8;4), (8;5), (8;6)\} \qquad S_3^2 = 6$

Abstände

$c_3^2 \ (5;1) = \| (5;1) - (\frac{41}{6} ; \frac{7}{2}) \|^2$

$= \| (-\frac{11}{6} ; -\frac{5}{2}) \|^2$

$= \sqrt{\frac{121}{36} + \frac{25}{4}} \simeq \frac{\sqrt{346}}{6} \approx 3.1$

$c_2^2 (5;1) = \| (5;1) - (\frac{12}{5} ; 3) \|^2 = \sqrt{(\frac{13}{5})^2 + (2)^2} = \frac{\sqrt{269}}{5} \approx 3.2$

$c_1^3 = \frac{1}{2} \cdot [(1;6)+(1;5)] = (1;\frac{11}{2})$

$c_2^3 = \frac{1}{4} \cdot [(1;4)+(3;3)+(3;2)+(4;1)]$

$= (\frac{11}{4};\frac{5}{2})$

$c_3^3 = \frac{1}{6}[(6;2)+(6;3)+(8;4)+(8;5)+(8;6)+(5;1)]$

$= \frac{1}{6}(41;21) = (\frac{41}{6};\frac{7}{2})$

$t=3$

$S_1^3 = \{(1;6),(1;5),(1;4)\}$ $\quad |S_1^3| = 3$

$S_2^3 = \{(3;3),(3;2),(4;1),(5;1)\}$ $\quad |S_2^3| = 4$

$S_3^3 = \{(6;2),(6;3),(8;4),(8;5),(8;6)\}$ $\quad |S_3^3| = 5$

$c_1^4 = \frac{1}{3}[(1;6),(1;5),(1;4)] = (1;5)$

$c_2^4 = \frac{1}{4}[(15;7)] = (\frac{15}{4};\frac{7}{4})$

3,75 $\quad$ 1,75

$c_3^4 = \frac{1}{5}(36;20) = (\frac{36}{5};4)$

7,2

$t=4$

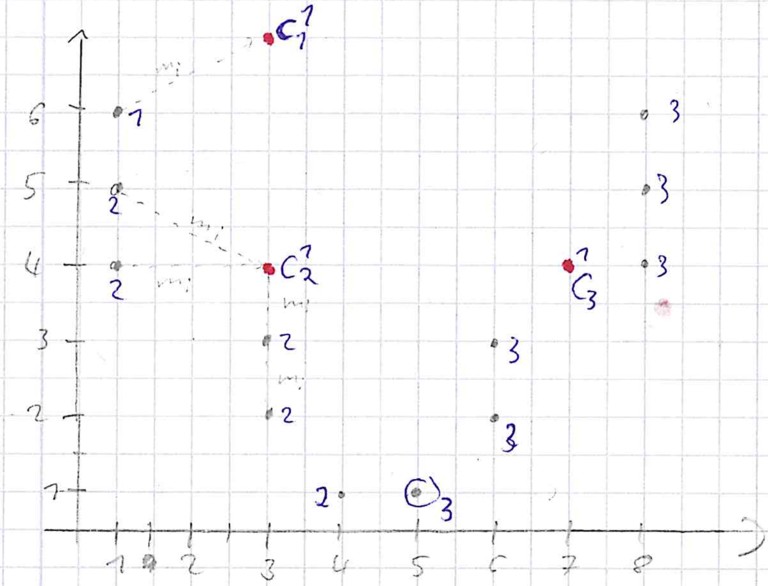
$S_1^4 = \{(1;6),(1;5),(1;4)\}$ $\quad |S_1^4| = 3$

$S_2^4 = \{(3;3),(3;2),(4;1),(5;1),(6;2)\}$ $\quad |S_2^4| = 5$

$S_3^4 = \{(6;3),(8;4),(8;5),(8;6)\}$ $\quad |S_3^4| = 4$

$c_1^5 = (1;5)$ $\quad c_2^5 = \frac{1}{5}(21;9)$ $\quad c_3^5 = \frac{1}{4}(30;18) = (\frac{15}{2};\frac{9}{2})$

A19 a)



● Clusterzentren

● Population

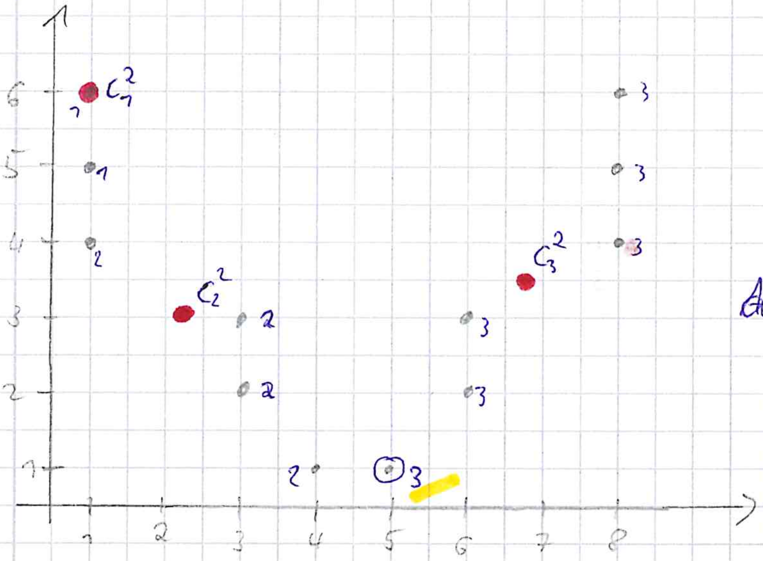
○ nicht offensichtlich

$(5;1)\ C_2 = \sqrt{13}$

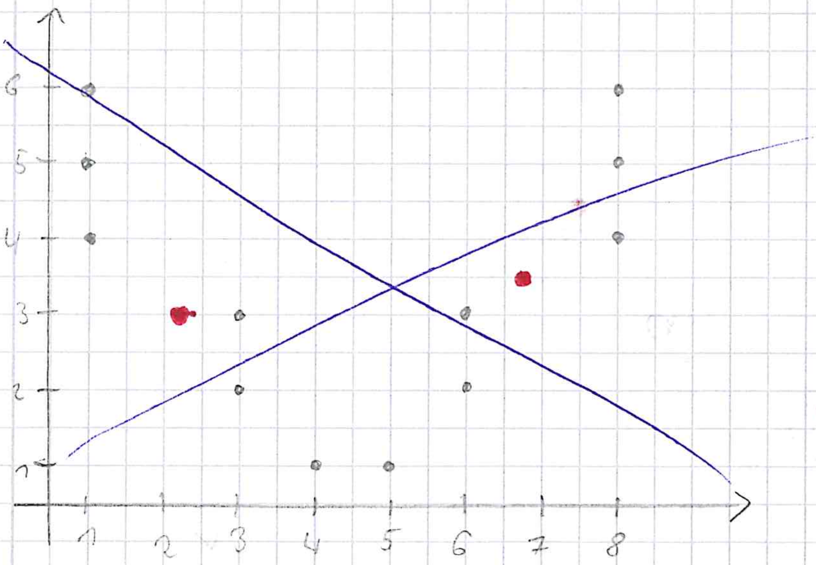
Pythagoras

$(5;1)\ C_3 = \sqrt{13}$

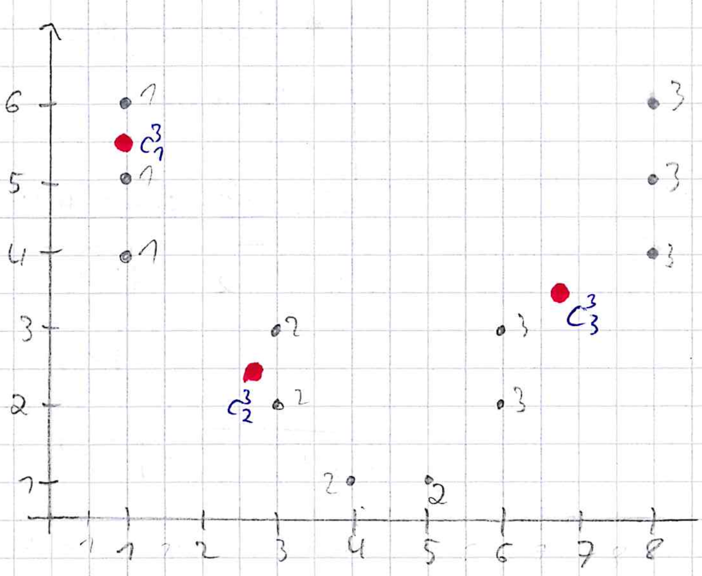
ordne $(5;1)$ zu C_3 zu

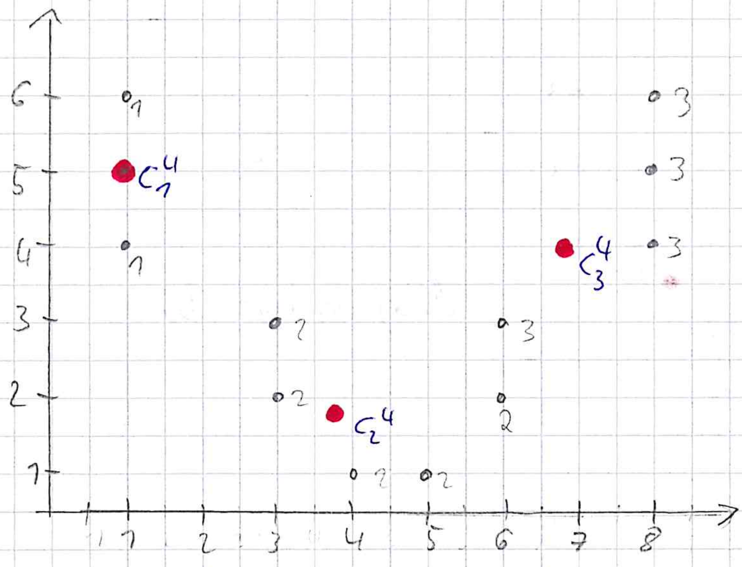


Ausgangslage b)

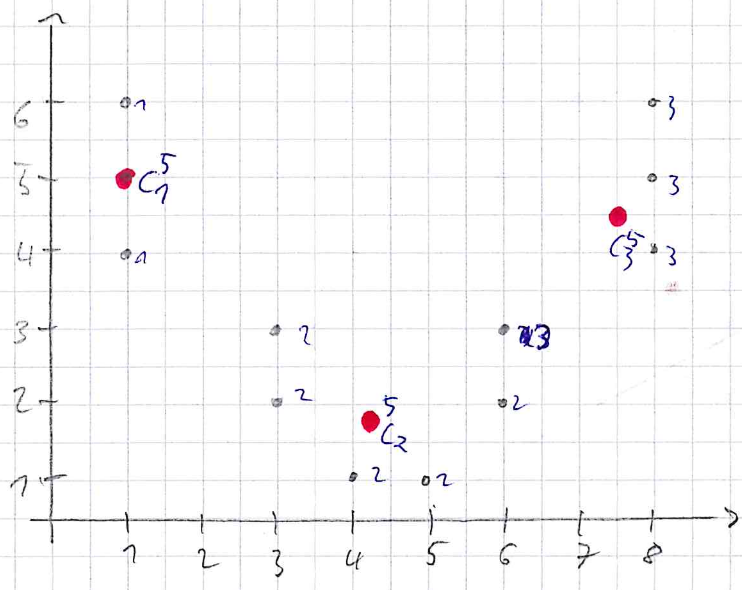


b)





!! Feels Inga Hödman



4P./5P.

$t = 5$

$S_1^5 = S_1^4$ $|S_1^5| = 3$

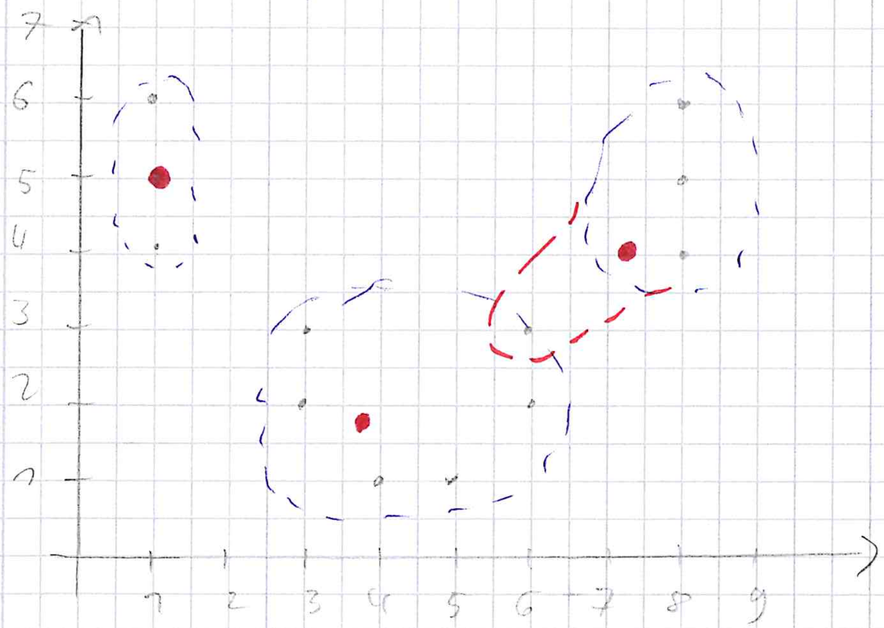
$S_2^5 = \{(3;3), (3;2), (4;1), (5;1)\}$ $|S_2^5| = \cancel{4}$ 5

$S_3^5 = \{\cancel{(6;2)}, (6;3)(8;4), (8;5), (8,6)\}$ $|S_3^5| = \cancel{5}$ 4

$C_1^6 = C_1^5$

$C_2^6 = \frac{1}{4}(15;7)$

$C_3^6 = \frac{1}{5}(36;20) = \left(\frac{36}{5}; 4\right)$



1.5P.

c) nach 6 Iterationen müsste das ganze konvergieren sodass (5)

ff.

konvergiert ⇒ $S_1^k = \{(1;6); (1,5); (1;4)\}$

$S_2^k = \{(3;3); (3,2); (4;1); (5;1); (6;2), (6;3)\}$

$S_3^k = \{(8;4), (8;5), (8;6)$

ja da es sich so wie in Skizze oben getrennt hat. (nach konvergieren

Nein, weil falsch eingezeichnet. 1.5P.

SMD-Nikolaus-Übungsblatt (7)

Abgabe: 06.12.18

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

1 Aufgabe 2

1.1 a)

Die Loss Funktion beschreibt den Schaden, der bei einem Entscheidungsproblem entsteht. Dabei ordnet sie jeder Entscheidung einen Schaden zu, welcher durch die Abweichung des geschätzten Parameters von dem wahren Parameter entsteht. Eine Loss Funktion

$$L[Y, f(x)] \tag{1}$$

wird durch die wahren Outputparameter Y und die aus den Startparametern x gefundene Funktion $f(x)$ beschrieben. Die Funktion $f(x)$ gibt dabei eine Approximation für die Outputparameter an. 0.5P

1.2 b)

Least Square Method, Support Vector Machines, Gradientenverfahren, uvm. 1P.

Nicht realisierbar

1.3 c)

Aktivierungsfunktionen sind nichtlinear und dienen dazu den Raum bei der Transformation so zu verzerren, dass Daten im verzerrten Raum mit geraden Schnitten getrennt werden können. Ohne die Aktivierungsfunktion wäre ein Neuronales Netz unsinnvoll, weil weiterhin einfach eine Reihe von linearen Transformationen durchgeführt wird.
Gängige Aktivierungsfunktionen sind z.B. der Tangens-Hyperbolicus-Funktion, die ReLU(Rectified-Linear-Unit)-Aktivierungsfunktion

$$f(x) = max(0, x) \tag{2}$$

und die Leaky-ReLU-Funktion

$$f(x) = \mathbb{1}(x < 0)(\alpha x) + \mathbb{1}(x \geq 0)(x) \tag{3}$$

1P.

1.4 d)

Die Zahl der Neuronen ist die Summe der Hidden-Layer und Output-Layer also alle Layer zu denen Informationen hingehen und verarbeitet werden. Ne. 0P.

1.5 e)

Bildsignalverarbeitung, Trennen von Kreisförmig angeordneten Daten ? 0.5P.

3/5P.

2 Aufgabe 3

a)
x_i ist M dimensionaler Spaltenvektor. ✓
$f(x_i, W, b)$ ist eine vektorwertige Funktion →K dim. Spaltenvektor.
W ist K×M-Matrix. ✓
C ist ein Skalar. ✓
$\nabla_W \hat{C}$ ist eine K×M-Matrix. ✓
$\nabla_{f_i} \hat{C}$ ist ein K dim. Spaltenvektor. ✓
$\frac{\partial f_{k,i}}{\partial W}$ ist K×M-Matrix. ✓
$\frac{\partial f_{k,i}}{\partial b}$ K dim Spaltenvektor. ✓
b?

2P.

b) Für Klasse a folgt:

$$\nabla_{f_a} C(f) = \frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{1}(y_i = a)\nabla \log \frac{\exp(f_{a,i})}{\sum_j \exp(f_{j,i})}\right) \tag{4}$$

Mit

$$\nabla \log \frac{\exp(f_{a,i})}{\sum_j \exp f_{j,i}} = \nabla\left(\log\exp(f_{a,i}) - \log\sum_j \exp(f_{j,i})\right) \tag{5}$$

$$= \exp(-f_{a,i}) \cdot \nabla \exp(f_{a,i}) - \frac{1}{\sum_j \exp(f_{j,i})} \cdot \nabla \sum_j \exp(f_{j,i}) \tag{6}$$

$$\text{mit} \qquad \exp(-f_{a,i}) \cdot \nabla \exp(f_{a,i}) = \mathbf{1}(y_i = a) \tag{7}$$

$$\Longrightarrow \mathbf{1}(y_i = a) - \frac{\exp(f_{j,i})}{\sum_j \exp(f_{j,i})} \tag{8}$$

$$\Longrightarrow \nabla_{f_a} C(f) = \frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{1}(y_i = a) - \frac{\exp(f_{j,i})}{\sum_j \exp(f_{j,i})}\right) \cdot (-1) \tag{9}$$

2P.

c)

$$\frac{\partial f_{k,i}}{\partial W} = \frac{\partial W_k \cdot x_i}{\partial W} + \frac{\partial b_k}{\partial W} = \begin{pmatrix} \vdots \\ \delta_{a,k} x_i^T \\ \vdots \end{pmatrix} \quad K \times M - \text{Matrix} \tag{10}$$

$\delta_{1,k}$

$$\frac{\partial f_{k,i}}{\partial b} = \frac{\partial W_k \cdot x_i}{\partial b} + \frac{\partial b_k}{\partial b} = \begin{pmatrix} \vdots \\ \delta_{a,k} \\ \vdots \end{pmatrix} \quad K\text{dim. Vektor} \tag{11}$$

$\delta_{1,k}$

1P.

Ideen zum Programmierteil Leider keine Zeit mehr gehabt das komplett zu implementieren. :(Deshalb nur Code-Schnippsel.

```
P_0 = pd.read_hdf('populationen.hdf5', key ='P_0')
P_1 = pd.read_hdf('populationen.hdf5', key ='P_1')
Pges = pd.DataFrame({
    'x': np.append(P_0['x'],P_1['x']),
    'y': np.append(P_0['y'],P_1['y']),
```

```
  #'X': np.append(Pges['x'],Pges['y']),
  #'classmask' : np.append(np.zeros(len(Pges['x'])),np.ones(len(Pges['y']))),
  'label' : np.append(np.zeros(len(P_0)),np.ones(len(P_1))), })
Wone = np.ones((2,len(Pges['X'])))
bone = np.ones(2)
def ffunk(k,W,b):
  if k == 0:
    return np.dot(W.T[k],Pges['x']) + b[0]
  else:
    return np.dot(W.T[k],Pges['y']) + b[1]
def softmax(k,i,W,b):
  return np.exp(ffunk(W,x,b)) /(np.exp(W[ ...
```

plots/Aufgabe3.py

0.5P

5.5/10P.

c)

$$\begin{pmatrix} w_{00} & w_{10} \\ w_{01} & w_{11} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \quad \text{mit } f_1 = f_2$$

$$w_{00} x_1 + w_{10} x_2 + b_0 = w_{01} x_1 + w_{11} x_2 + b_1$$

$$\Leftrightarrow \quad x_2 (w_{10} - w_{11}) = (w_{01} - w_{00}) x_1 + b_1 - b_0$$

$$\Leftrightarrow \quad x_2 = \frac{1}{(w_{10} - w_{11})} \left((w_{01} - w_{00}) x_1 + b_1 - b_0 \right)$$