Lösungen für Aufgabenblatt 10

Einführung in die Computerlinguistik (WS19/20) Abgabe bis

Aufgabe 1) Term-Term-Ähnlichkeiten **Punkte:** –

Gegeben sind sind die folgenden Kookurrenzen aus dem BNC-Corpus:

	species	motion	area	environment	star	idea	heat
animal	616	15	230	266	53	138	41
atmosphere	8	7	84	39	19	46	93
carnivore	21	0	3	0	0	4	0

a) Kosinusähnlichkeit *Punkte: –*

Berechnen Sie die Kosinusähnlichkeit zwischen den Begriffen *animal*, *atmosphere* und *carnivore* indem Sie die Spalteneinträge als Featuredimensionen behandeln.

Lösung zu 1a)

- $\|animal\| = \sqrt{616^2 + 15^2 + 230^2 + 266^2 + 53^2 + 138^2 + 41^2} = 725.86$
- $\|atmosphere\| = \sqrt{8^2 + 7^2 + 84^2 + 39^2 + 19^2 + 46^2 + 93^2} = 140.77$
- $\|carnivore\| = \sqrt{21^2 + 0^2 + 3^2 + 0^2 + 0^2 + 4^2 + 0^2} = 21.59$
- cos(animal, atmosphere) $= \frac{616 \cdot 8 + 15 \cdot 7 + 230 \cdot 84 + 266 \cdot 39 + 53 \cdot 19 + 138 \cdot 46 + 41 \cdot 93}{\|animal\| \cdot \|atmosphere\|} = 0.45$
- cos(animal, carnivore) $= \frac{616 \cdot 21 + 15 \cdot 0 + 230 \cdot 3 + 266 \cdot 0 + 53 \cdot 0 + 138 \cdot 4 + 41 \cdot 0}{\|animal\| \cdot \|carnivore\|} = 0.9$
- cos(carnivore, atmosphere) $= \frac{21 \cdot 8 + 0 \cdot 7 + 3 \cdot 84 + 0 \cdot 39 + 0 \cdot 19 + 4 \cdot 46 + 0 \cdot 93}{\|carnivore\| \cdot \|atmosphere\|} = 0.20$

b) PPMI-Umwandlung *Punkte: x*

Beschränken wir uns nun zu Veranschaulichungszwecken auf einen kleineren Ausschnitt der Matrix:

	species	motion	area
animal	616	15	230
atmosphere	8	7	84

Wandeln Sie die Kookurrenzen in PPMI-Einträge um.

Lösung zu 1b)
Die Tabelle mit Marginalen sieht wie folgt aus:

	species	motion	area	
animal	616	15	230	861
atmosphere	8	7	84	99
	624	22	314	960

Einzelwahrscheinlichkeiten:
P(w=animal) = 0.897
P(w=atmosphere) = 0.103
P(c=species) = 0.65
P(c=motion) = 0.023
P(c=area) = 0.327

Gemeisame Wahrscheinlickeiten:
P(w=animal, c=species) = 0.64
P(w=animal, c=motion) = 0.016
P(w=animal, c=area) = 0.24
P(w=atmosphere, c=species) = 0.008
P(w=atmosphere, c=motion) = 0.007
P(w=atmosphere, c=area) = 0.0875

PPMIs:

	species	motion	area
animal	0.134	0 (-0.37)	0 (-0.29)
atmosphere	0 (-3.07)	1.63	1.38

- PPMI(animal, species) = $\max(\log_2 \frac{0.64}{0.897 \cdot 0.65}, 0) = 0.134$
- PPMI(animal, motion) = $\max(\log_2 \frac{0.016}{0.897 \cdot 0.023}, 0) = 0 \quad (-0.37)$
- PPMI(animal, area) = $\max(\log_2 \frac{0.24}{0.897 \cdot 0.327}, 0) = 0 \quad (-0.29)$
- PPMI(atmosphere, species) = $\max(\log_2 \frac{0.008}{0.103 \cdot 0.65}, 0) = 0 \quad (-3.07)$
- PPMI(atmosphere, motion) = $\max(\log_2 \frac{0.007}{0.103 \cdot 0.023}, 0) = 1.63$
- PPMI(atmosphere, area) = $\max(\log_2 \frac{0.0875}{0.103 \cdot 0.327}, 0) = 1.38$

Man sieht zum Beispiel, dass die laut Frequenzen sehr kleine Abhängigkeit (7) von *atmosphere* zu *motion* nun höher bewertet wird.

Aufgabe 2) Euklidische und Cosinus-Distanz für Einheitsvektoren **Punkte:** –

Zeigen Sie, dass für beliebige Einheitsvektoren $\overrightarrow{p}$, $\overrightarrow{v}$, $\overrightarrow{w}$ gilt:

$$d_{euklid}(\overrightarrow{p}, \overrightarrow{v}) \leq d_{euklid}(\overrightarrow{p}, \overrightarrow{w}) \Leftrightarrow d_{cos}(\overrightarrow{p}, \overrightarrow{v}) \leq d_{cos}(\overrightarrow{p}, \overrightarrow{w}).$$

Hierbei ist $d_{cos}(\overrightarrow{x}, \overrightarrow{y}) := 1 - sim_{cos}(\overrightarrow{x}, \overrightarrow{y})$ für alle Vektoren $\overrightarrow{x}, \overrightarrow{y}$ definiert.[1] Einheitsvektoren sind Vektoren der Länge 1.

[1] Dies ist im allgemeinen keine Metrik, da u.a. die Dreiecksungleichung nicht gilt. Oft wird dies stattdessen eine *dissimilarity* genannt.

Lösung zu 2)

Alle Vektoren ab jetzt sollen Einheitsvektoren sein.

$$d_{euklid}(\overrightarrow{p}, \overrightarrow{q}) = \sqrt{\sum_i (p_i - q_i)^2} \tag{1}$$

$$= \sqrt{\sum_i p_i^2 - 2\sum_i p_i q_i + \sum_i q_1^2} \tag{2}$$

$$|\overrightarrow{p}| = \sqrt{\sum_i p_i^2} = d_{euklid}(\overrightarrow{p}, 0) = 1 \tag{3}$$

$$\sqrt{\sum_i p_i^2} = 1 \Leftrightarrow \sum_i p_i^2 = 1 \tag{4}$$

$$d_{euklid}(\overrightarrow{p}, \overrightarrow{q}) = \sqrt{1 - 2\sum_i p_i q_i + 1} \tag{5}$$

$$= \sqrt{2 - 2\sum_i p_i q_i} = \sqrt{2(1 - \sum_i p_i q_i)} \tag{6}$$

$$s_{cos}(\overrightarrow{p}, \overrightarrow{q}) = \frac{\overrightarrow{p} \cdot \overrightarrow{q}}{|\overrightarrow{p}||\overrightarrow{q}|} = \frac{\sum_i p_i q_i}{|\overrightarrow{p}||\overrightarrow{q}|} \tag{7}$$

$$= \frac{\sum_i p_i q_i}{1 \cdot 1} = \sum_i p_i q_i \tag{8}$$

$$d_{cos}(\overrightarrow{p}, \overrightarrow{q}) = 1 - s_{cos}(\overrightarrow{p}, \overrightarrow{q}) = 1 - \sum_i p_i q_i \tag{9}$$

$$\text{Ang. } d_{euklid}(\overrightarrow{p}, \overrightarrow{v}) \leq d_{euklid}(\overrightarrow{p}, \overrightarrow{w}) \tag{10}$$

$$\Leftrightarrow \sqrt{2(1 - \sum_i p_i v_i)} \leq \sqrt{2(1 - \sum_i p_i w_i)} \tag{11}$$

$$\Leftrightarrow 2(1 - \sum_i p_i v_i) \leq 2(1 - \sum_i p_i w_i) \tag{12}$$

$$\Leftrightarrow 1 - \sum_i p_i v_i \leq 1 - \sum_i p_i w_i \tag{13}$$

$$\Leftrightarrow d_{cos}(\overrightarrow{p}, \overrightarrow{v}) \leq d_{cos}(\overrightarrow{p}, \overrightarrow{w}) \tag{14}$$

1. Definition Euklidische Distanz
2. 2. Binomische Formel
3. Definition Einheitsvektor
4. $\sqrt{1} = 1$

5. Einsetzen (4) in (2)
6. Vereinfachen
7. Definition Cosinus-Ähnlichkeit
8. Siehe (3)
9. Umformung von Ähnlichkeits- zu Distanz-Maß
10. (10-14) Abschluss

Daraus ergibt sich, dass ein Ähnlichkeitsranking für Wortpaare iudentisch zwischen Kosinusdistanz und Euklidischer Distanz ist, solange die Vektoren vorher normiert werden.

Aufgabe 3) Pointwise Mutual Information **Punkte:** –

Aus Suchanfragen von Unigrammen und Bigrammen einer bekannten Online-Suchmaschine lässt sich folgende Matrix erstellen:

	Emily	Charlotte	...	
Brontë	2.620	3.120	...	25.100
Dickinson	12.000	18	...	92.100
...	...	...	...	...
	604.000	733.000	...	100.000.000

Emily liefert also 604.000 Ergebnisse, während das Bigram *Emily Dickinson* 12.000 mal gefunden wurde. Insgesamt wurden 100 Millionen Suchergebnisse ausgewertet.

1. Berechnen Sie Pointwise Mutual Information für *Emily Dickinson*, *Emily Brontë* und *Charlotte Dickinson*.

2. Deckt sich das Ergebnis mit ihrer Intuition?

3. Angenommen der Gebirgszug *Ephel Dúath* aus Herr der Ringe taucht in unseren Suchergebnissen einmal auf und insbesondere treten *Ephel* und *Dúath* jeweils nur einmal und nur zusammen auf. Berechnen Sie PMI für *Ephel Dúath*. Wie verändert sich der Wert für ein größeres N, d.h wenn wir mehr Ergebnisse auswerten, aber keine weiteren Vorkommen der beiden Wörter finden? Halten Sie diese Tendenz für gerechtfertigt?

Lösung zu 3)

1.

$$\begin{aligned} \mathrm{pmi}(E,D) &= \log \frac{n_{E,D} \cdot N}{(n_E) \cdot (n_D)} = \log 21.57 \\ &\approx 4.43 \\ \mathrm{pmi}(E,B) &= \log 17.281 && \approx 4.111 \\ \mathrm{pmi}(C,D) &= \log 0.026 && \approx -5.26 \end{aligned}$$

2. Charlotte Dickinson sollte einen negativen PMI-Wert haben, da es sich hierbei nicht um eine bekannte Schriftstellerin (oder generell um eine bekannte Person) handelt. Dies ist, wenn richtig berechnet, auch der Fall. Des weiteren ist *Emily* mit *Bronte* fast genauso hoch assoziiert wie mit *Dickinson*, obwohl die Bigrammfrequenz der ersteren Kombination deutlich niedriger ist als die der zweiten. (Grund: seltener Nachname *Bronte*).

3. $\mathrm{pmi}(Ephel, Duath) = log\frac{N}{1}$

   Mit einem größeren N steigt auch der PMI für solche Singletons. Da ein größeres Korpus nicht automatisch zu höheren Werten für solche Singletons führt (zipf'sche Verteilung),

wird hier insignifikanten Werten eine sehr große Assoziation zugesprochen, die nicht unbedingt durch die Korpusgröße zu verantworten ist.

Aufgabe 4) Agglomeratives Clustering **Punkte:** –

Gegeben seien die folgenden Daten[2]:

ID	Stern	scheinbare Helligkeit	absolute Helligkeit
0	Proxima Centauri	11	16
1	α Centauri A	0	4
2	α Centauri B	1	6
3	Barnards Pfeilstern	10	13
4	Wolf 359	14	17
5	Lalande 21185	7	10
6	α Canis Majoris A	-1	1
7	α Canis Majoris B	8	11

Wir nutzen für diese Aufgabe die Euklidische Distanz.

Um den Rechenaufwand zu minimieren bzw. Programmierung zu vermeiden, ist Ihnen die symmetrische Distanzmatrix vorgegeben.

	0	1	2	3	4	5	6	7
0	0.00	16.28	14.14	3.16	3.16	7.21	19.21	5.83
1	16.28	0.00	2.24	13.45	19.10	9.22	3.16	10.63
2	14.14	2.24	0.00	11.40	17.03	7.21	5.39	8.60
3	3.16	13.45	11.40	0.00	5.66	4.24	16.28	2.83
4	3.16	19.10	17.03	5.66	0.00	9.90	21.93	8.49
5	7.21	9.22	7.21	4.24	9.90	0.00	12.04	1.41
6	19.21	3.16	5.39	16.28	21.93	12.04	0.00	13.45
7	5.83	10.63	8.60	2.83	8.49	1.41	13.45	0.00

Geben Sie die Reihenfolge der Zusammenführung der einzelnen Cluster, sowie die resultierenden Dendrogramme für folgende Gruppenähnlichkeitsmaße an:

1. Single Link
2. Complete Link
3. Average Link

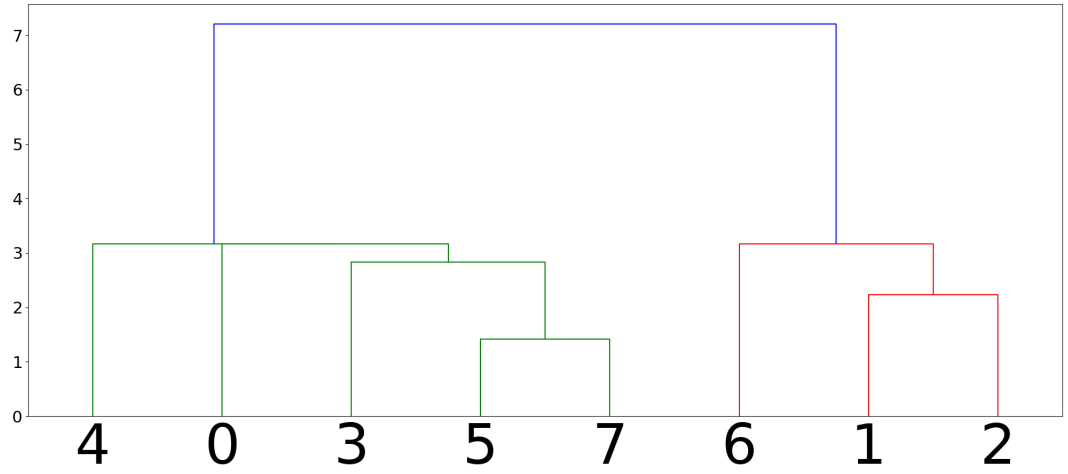
Welche typischen Eigenschaften von complete link vs. single link clustering-Methoden kann man erkennen?

Wenn man nur eine Dimension (zum Beispiel die scheinbare Helligkeit) zum Clustern verwenden würde, würde sich dann in diesem Beispiel viel am Clustering ändern?

Lösung zu 4)

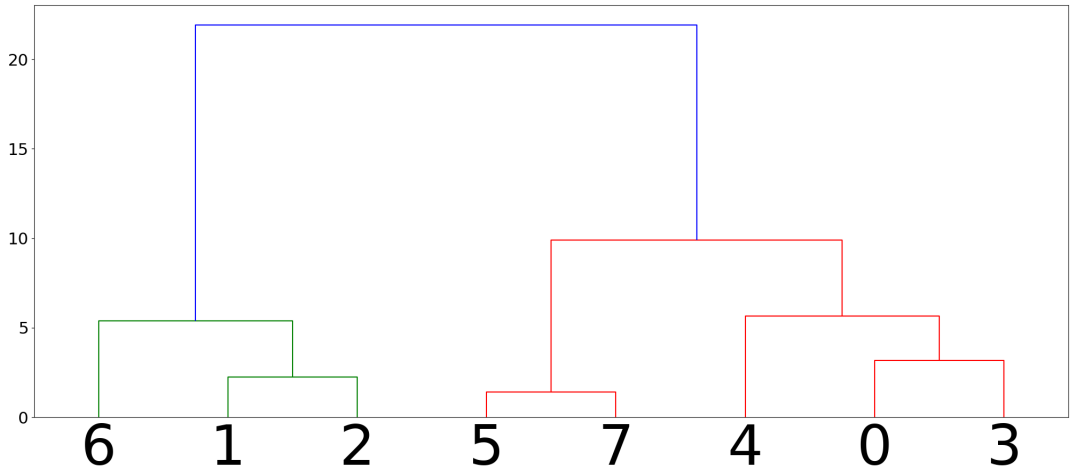
[2]Entnommen der Liste der nächsten extrasolaren Sternensysteme

1. Lösung zum single-link clustering:



Hier ist ein gleichzeitiges Mergen mehrerer Punkte/Cluster (wie 0,3,4) möglich, da sich die besten Kandidatencluster für einen Punkt beim single link clustering nicht ändern. Sprich, der beste Kandidat in Schritt 3 für 4 ist 0. Wenn man nun 0 und 3 zu $\{0, 3\}$ merged, ist der beste Kandidate für 4 dann $\{0, 3\}$, da man das Minimum der Distanzen (vorheriger bester Kandidat 0) wählt und somit sich sowohl die Distanz als auch der beste Kandidat nicht ändert. Man kann damit also auch gleichzeitig mergen.

2. Lösung zum complete link clustering:

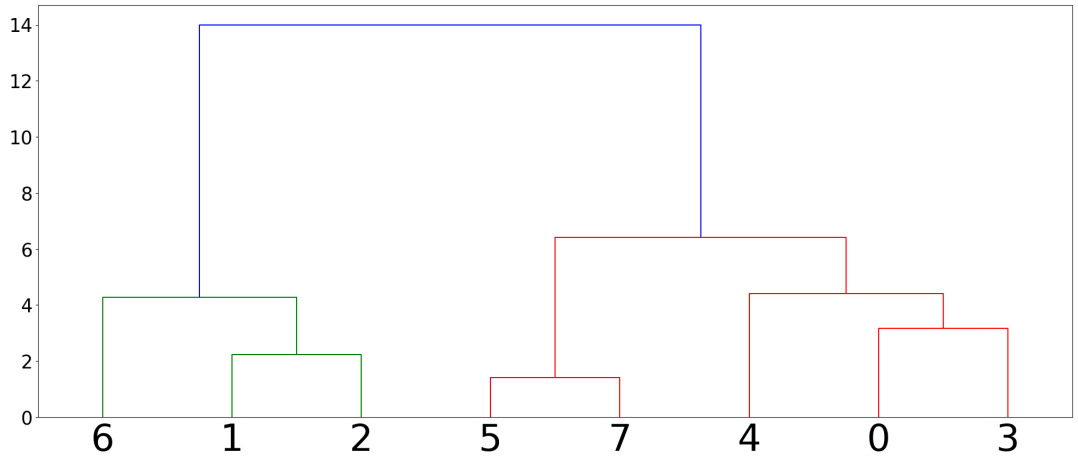


Korrektur im Gegensatz zur Vorlesung:

Fälschlicherweise wurden in der Vorlesung gleich mehrere Cluster bzw Punkte gemerged,

falls der Abstand gleich war. Zum Beispiel wären nach dieser Vorgehensweise im dritten Schritt 0,3 und 4 zusammen gemerged worden. Dies ist beim complete und average link clustering nicht richtig! Es dürfen immer nur zwei Cluster gemerged werden: bei gleichem Abstand wird zufällig gewählt. Der Grund ist, dass sich beim complete sowie beim average link clustering der beste Kandidat sowie der Abstand nach einem merge ändern kann (siehe Unterschied zur vorherigen Erklärung). So ist der Abstand von 4 zu $\{0, 3\}$ nun beim complete clustering größer geworden (5.66) als der vorherige Abstand von 0 zu 4. Dies kann zu Änderungen im späterem Clustering führen. Also bitte immer nur zwei Clusters mergen pro Schritt. Entschuldigung!

3.



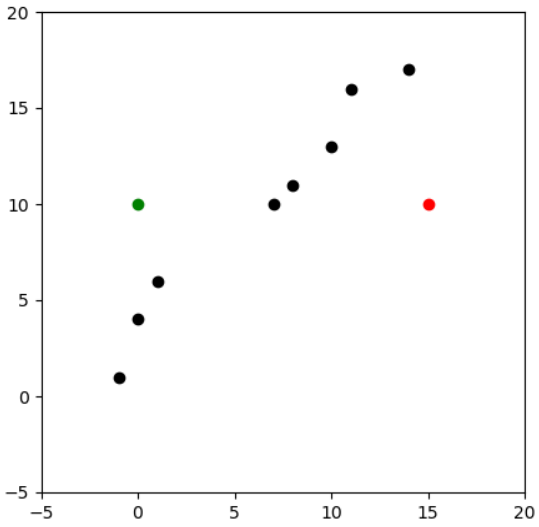
Man kann die Tendenz zu langen Clustern bei single link erkennen.

In dem Beispiel korrelieren die beiden Dimensionen sehr stark miteinander, d.h. wenn die scheinbare Helligkeit groß ist, ist auch die absolute Helligkeit hoch und umgekehrt. Das heißt, dass auch wenn man nur nach einer Dimension clustern würde, im Endeffekt die gleichen Cluster entstehen. In der Praxis sollte man solche Abhängigkeiten zwischen Merkmalen als Vektordimensionen vermeiden, da diese dann sozusagen "doppelt" gewertet werden.

Aufgabe 5) k-means Clustering zum Rechnen **Punkte:** –

Berechnen Sie nun 2-means Clustering bis zur Konvergenz für die Tabelle der Sternensysteme aus der vorherigen Aufgabe 4. Die Zentroiden sind wie folgt initiiert:

Zentroid	Farbe	x	y
c1	rot	15	10
c2	grün	0	10



Geben Sie für jeden Zuweisungsschritt die Koordinaten der zwei Zentroiden, sowie die Menge der Punkte für jeden Cluster an.

Inwieweit hängt die Lösung mit den Lösungen beim agglomerativem Clustering der vorherigen Aufgabe zusammen?

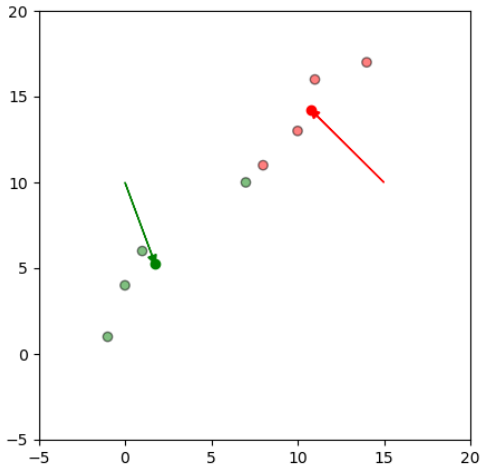
Lösung zu 5)
Folgend sind die Iterationen aufgelistet:

1. Zuweisung:

	x	y	distance_from_c1	distance_from_c2	closest	color
0	11	16	7.211103	12.529964	c1	r
1	0	4	16.155494	6.000000	c2	g
2	1	6	14.560220	4.123106	c2	g
3	10	13	5.830952	10.440307	c1	r
4	14	17	7.071068	15.652476	c1	r
5	7	10	8.000000	7.000000	c2	g
6	-1	1	18.357560	9.055385	c2	g
7	8	11	7.071068	8.062258	c1	r

Update:

Zentroid	Farbe	x	y
c1	rot	10.75	14.25
c2	grün	1.75	5.25

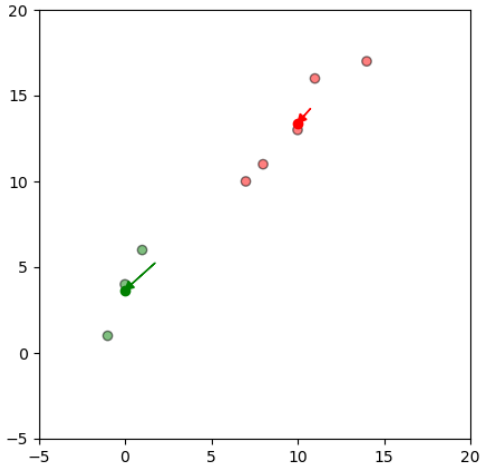


2. Zuweisung:

	x	y	distance_from_c1	distance_from_c2	closest	color
0	11	16	1.767767	14.181855	c1	r
1	0	4	14.853451	2.150581	c2	g
2	1	6	12.772040	1.060660	c2	g
3	10	13	1.457738	11.319231	c1	r
4	14	17	4.257347	16.974245	c1	r
5	7	10	5.667892	7.079901	c1	r
6	-1	1	17.709461	5.062114	c2	g
7	8	11	4.257347	8.492644	c1	r

Update:

(neuer) Zentroid	Farbe	x	y
c1	rot	10	13.4
c2	grün	0	3.7



3. Zuweisung (keine Veränderung mehr):

	x	y	distance_from_1	distance_from_2	closest	color
0	11	16	1.767767	14.181855	1	r
1	0	4	14.853451	2.150581	2	g
2	1	6	12.772040	1.060660	2	g
3	10	13	1.457738	11.319231	1	r
4	14	17	4.257347	16.974245	1	r
5	7	10	5.667892	7.079901	1	r
6	-1	1	17.709461	5.062114	2	g
7	8	11	4.257347	8.492644	1	r

Aufgabe 6) Clustering zum Denken **Punkte:** –

- Gegeben seien Goldcluster $G_1 \dots G_n$ und ein erstelltes Clustering $C_1 \dots C_n$. Das Clustering sei perfekt, d.h. wir haben es geschafft, die Goldcluster genau zu reproduzieren. Wie hoch ist die Reinheit aller Cluster? Eine zweite Methode erstellt $2n$ Cluster, wobei jedes Cluster C_i zufällig in zwei geteilt wurde. Wie hoch ist die Reinheit dieser Cluster? Leiten Sie daraus ein Problem der Benutzung von Reinheit als Evaluationsmaß ab. Wie vermeiden NMI und Rand-Index dieses Problem?

- Überlegen Sie sich ein Beispiel für 3-means clustering (Punkte und Anfangszentroiden), bei dem der Algorithmus einen leeren Cluster generiert.

- Gehen Sie auf die Demo bei `http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html` und versuchen Sie den k-means-Algorithmus zu überlisten, d.h. konstruieren Sie ein Beispiel, so dass keine Konvergenz zum globalen Optimum erfolgen kann.

Lösung zu 6)

- Bei einem perfekten Clustering ist die Reinheit aller Cluster 1. Wenn man die Cluster einfach noch einmal teilt, ändert sich an der Reinheit nichts (die Reinheit misst ja nur eine Art precision mit dem am besten passenden Goldcluster). Reinheit bevorzugt also kleine Cluster. Mit sehr vielen Clustern kann also eventuell einfach hohe Reinheit erzielt werden (z.B. wenn jeder Punkt sein eigenes Cluster erhält).

  Eine Möglichkeit ist, stattdessen linkbasierte Maße zu nehmen (siehe auch Koreferenzmaße, die ja im Endeffekt auch zwei Clusterings vergleichen). Der Randindex ist linkbasiert und bestraft auch für False Negatives, also Links, die nicht erkannt werden.

- Viele Antworten möglich. Einfachste Möglichkeit ist ein Anfangszentroid, der so weit von den Punkten entfernt ist, dass er keine Punkte zugewiesen bekommt.

- Eine Möglichkeit sind nicht-konvexe Cluster.