Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Ähnlichkeitssuche in Multimedia-Daten

Ungarischer Algorithmus im Kontext der Ähnlichkeitsberechnung (Zuordnungsproblem)

Von:
Timon Brüning

Überblick

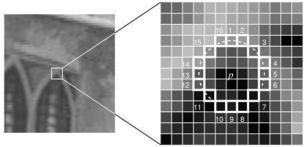
- Einführung
- Ansätze der Ähnlichkeitsberechnung
  - Assignment Problem
  - Earth Mover’s Distance (EMD)
- Ungarische Methode
- Weitere Methode (Simplex)
- Zusammenfassung

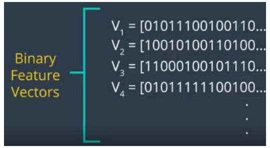
1. Einführung

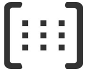
Einführung – Wo sind wir?

- **Bilderimport**
- **Signaturbildung**
- **Berechnung der Ähnlichkeit**
- **Anfragetypen**
- **Optimierung**

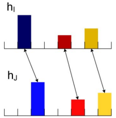
```
Mat image1 = Highgui.imread(pathImg1, Highgui.CV_LOAD_IMAGE_COLOR);
```



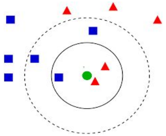




Hungarian/ Simplex-Algortihm

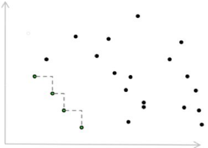


Earth mover's distance



kNN

Skyline Anfrage



Berechnungszeit

Speicherauslastung

[Abv.]

2.
Ansätze der Ähnlichkeitsberechnung

Assuming that numerical scores are available for the performance of each of n persons on each of n jobs, the "assignment problem" is the quest for an assignment of persons to jobs so that the sum of the n scores so obtained is as large as possible.

”

- H. W. Kuhn [1]

Assignment Problem: Beispiel 1

Beispiel 1 - “The simple assignment problem”

Situation: Es stehen vier Individuen für vier Jobs zur Verfügung.

Gegeben: Qualifikationen der Individuen für die Jobs. {0,1}

Aufgabe: Die Anzahl belegter Jobs soll maximiert werden, wobei jede Person nur einen Job ausüben kann.

Person i \ Job j	1	2	3	4
1	1	1	1	0
2	0	0	1	1
3	0	0	0	1
4	0	0	0	1

Assignment Problem: Beispiel 1

$$\begin{bmatrix} 1 & 1 & 1^* & 0 \\ 0 & 0 & 1 & 1^* \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{“transfer”}} \begin{bmatrix} 1^* & 1 & 1 & 0 \\ 0 & 0 & 1^* & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{“transfer”}} \begin{bmatrix} 1^* & 1 & 1 & 0 \\ 0 & 0 & 1^* & 1 \\ 0 & 0 & 0 & 1^* \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Diese Zuordnung ist “complete”

Keine Möglichkeit einen weiteren Job zuzuordnen

Diese Zuordnung ist “incomplete”

Job 4 wurde nicht zugeordnet

Diese Zuordnung ist “optimal”

Transfer ist möglich, führt jedoch wieder zu complete

Assignment Problem

“The general assignment problem” (GAP)

Situation: Es stehen i Individuen für j Jobs zur Verfügung.

Gegeben: Bewertungen r der Individuen für die Jobs. ($\mathbb{N}_{(0)}$)

Aufgabe: Finde die Zuordnung, sodass Summe aller r_{ij} in dieser maximal.

Person i \ Job j	1	2
1	5	3
2	1	2

Assignment Problem: Beispiel 2

Beispiel 2

Situation: Verkaufsleiter ist verantwortlich für drei Verkäufer an unterschiedlichen Standorten. Diese sollen den drei neuen Geschäftsstellen aushelfen.

Gegeben: Preise günstigster Tickets in Euro.

Aufgabe: Wohin soll jeder Verkäufer geschickt werden, um die Reisekosten zu minimieren.

Von \ Nach	Dortmund	Amsterdam	Frankfurt
Hamburg	100	200	150
München	200	300	150
Leipzig	50	200	100

Assignment Problem: Beispiel 2

$$\begin{bmatrix} 100 & 200 & 150 \\ 200 & 300 & 150 \\ 50 & 200 & 100 \end{bmatrix}$$

Gesamte Kosten dieser Zuordnung belaufen sich auf:

100 + 300 + 100 = <u>500€</u>

$$\begin{bmatrix} 100 & 200 & 150 \\ 200 & 300 & 150 \\ 50 & 200 & 100 \end{bmatrix}$$

Gesamte Kosten dieser Zuordnung belaufen sich auf:

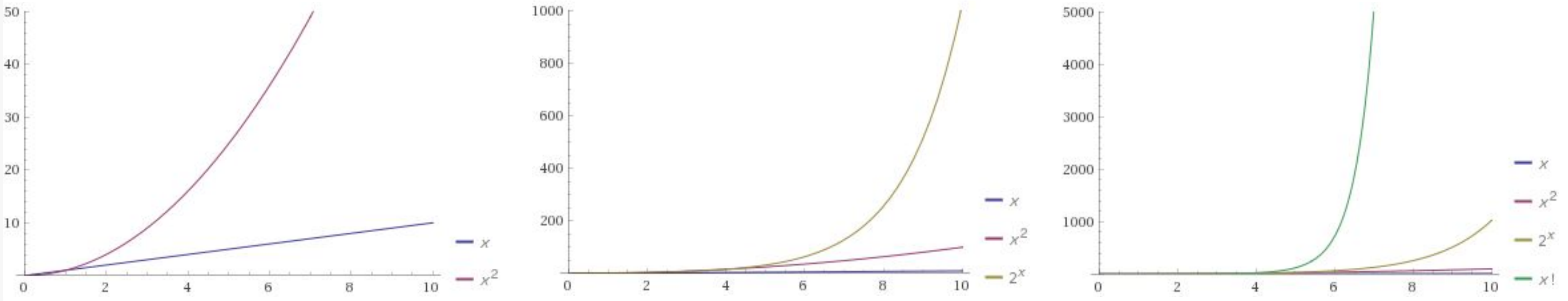
100 + 200 + 150 = <u>450€</u>

$$\begin{bmatrix} 100 & 200 & 150 \\ 200 & 300 & 150 \\ 50 & 200 & 100 \end{bmatrix}$$

Nach ausprobieren aller 6 Möglichkeiten, ergeben sich die optimalen Gesamtkosten:

50 + 200 + 150 = <u>400€</u>

Assignment Problem: Beispiel 2



[2]

Es gibt x! Möglichkeiten x Resourcen zu x Aufgaben zuzuordnen.

Bei vorherigem Beispiel und 10 Verkäufern und Städten:

3 628 800 Möglichkeiten

Assignment Problem: Dualität

Primales Problem

Maximiere Zielfunktion (r_{ij})

Person i \ Job j	1	2
1	5	3
2	1	2

Duales Problem

Weise Wert aus $\mathbb{N}_0$ zu:

- Jeder Person ← u_i
- Jedem Job ← v_j

Wobei: $u_i + v_j \geq r_{ij}$ $(i,j = 1, \dots, n)$

Das duale Problem zum (maximierenden) GAP fragt nun nach dem Minimum von $u_1 + \dots + u_n + v_1 + \dots + v_n$

Earth Mover's Distance

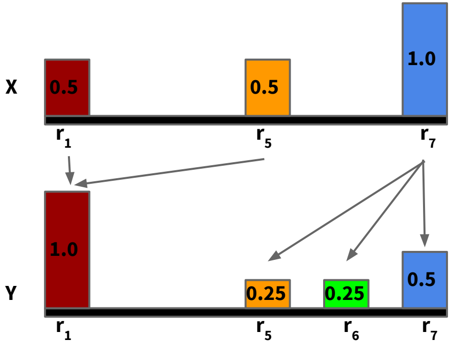
Recap (EMD)

Situation:

- Erdhaufen R_X mit Kapazitäten $X(r_i)$ für $r_i \in R_X$
- Erdlöcher R_Y mit Kapazitäten $Y(r_j)$ für $r_i \in R_Y$
- Kosten für das Bewegen einer Einheit Erde
- Alle möglichen Flüsse

Aufgabe:

Finde den maximalen Fluss von R_X nach R_Y mit den minimalsten Kosten



Mit $\delta(r_i,r_j)=|i-j|$:

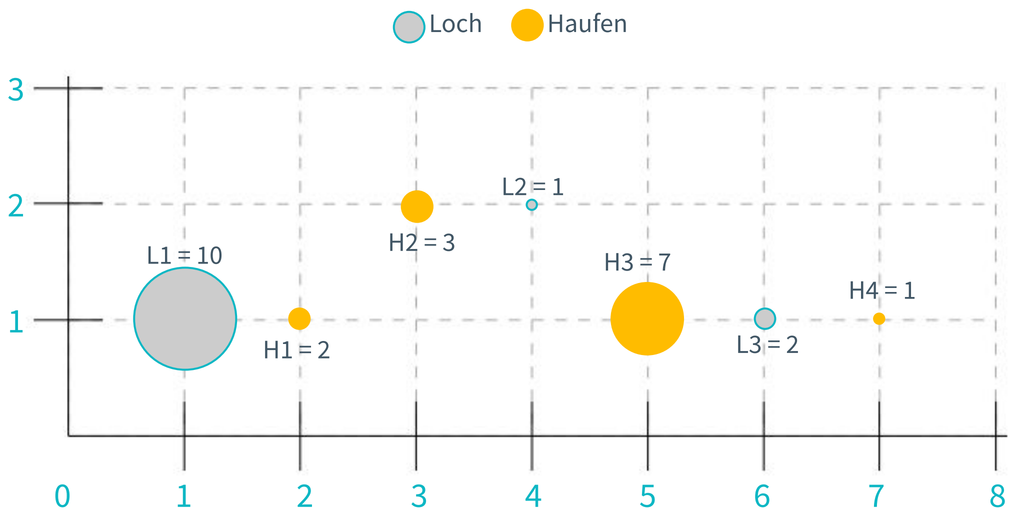
$EMD_\delta(X,Y) = f(r_5,r_1)*\delta(r_5,r_1) + f(r_7,r_5)*\delta(r_7,r_5) + f(r_7,r_6)*\delta(r_7,r_6)$

$= 0.5*4 + 0.25*2 + 0.25*1$

$= 2.75$

Earth Mover's Distance: 2D

Von	Nach	Fluss	Distanz	Σ Kosten	
H1	L1	2	1.00	2.00	
H2	L1	3	2.24	6.71	
H3	L1	5	4.00	20.00	
H3	L2	1	1.41	1.41	
H3	L3	1	1.00	1.00	
H4	L3	1	1.00	1.00	
		13		32.12	2.47/ Fluss-einheit



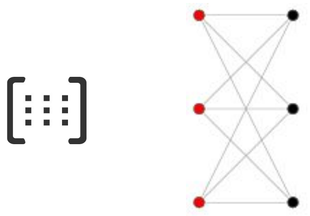
3.
Die Ungarische Methode

The Hungarian Method

Hintergrund

- Veröffentlicht von H.W. Kuhn, Januar 1955
- Baut auf Ideen der ungarischen Mathematiker D. König und E. Egerváry auf
- Ursprüngliche Laufzeit von $O(n^4)$ konnte später auf $O(n^3)$ gesenkt werden
- Zwei arten der mathematischen Formulierung:
  - Quellen und Senken als bipartiter Graph
  - Formulierung als Zuordnungsproblem in einer quadratischen Matrix

> One interesting aspect of the algorithm is the fact that it is latent in work of D. König and E. Egerváry that predates the birth of linear programming by more than 15 years (hence the name, the "Hungarian Method"). [1]



[Abv.]

The Hungarian Method: Lösung der Matrixinterpretation

Vorbereitung

- Wollen wir die totalen Kosten minimieren oder maximieren?
    - Bei max: negiere alle Elemente; Addiere den ursprünglich größten Wert auf alle drauf
- Liegt eine quadratische Matrix vor?
    - Wenn nicht: ergänze zu n*n und fülle Elemente mit Null

Ablauf

1. Für jede Zeile Z: subtrahiere das kleinste Element aus Z von jedem Element aus Z
2. Für jede Spalte S: subtrahiere das kleinste Element aus S von jedem Element aus S
3. Versuche alle Nullen mit einer minimalen Anzahl an horizontalen/vertikalen Linien L zu überdecken
    3.1. Wenn |L| = n: optimale Zuordnung existiert
    3.2. Wenn |L| < n: → Schritt 4
4. Finde das kleinste k ∉ L. Subtrahiere k von allen Elementen die nicht in L sind. Addiere k auf alle Elemente die in L doppelt vorkommen. → Schritt 3

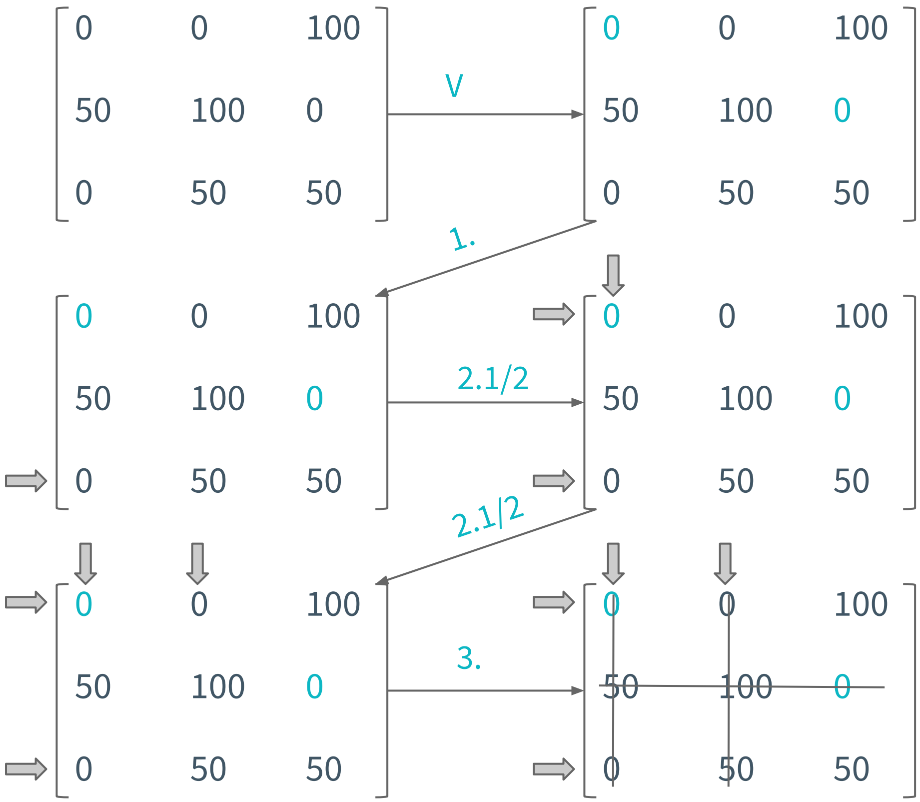
The Hungarian Method:
Schritt 3 - Minimale Linenzahl

Vorbereitung

- Weise jeder Zeile eine unzugewiesene Spalte bei entsprechender Null zu.

Ablauf

1. Sollte es unzugewiesene Zeilen geben, markiere diese. Ansonsten existiert eine optimale Zuordnung → Ende
2. Wiederhole, bis keine Änderung mehr eintritt: Markiere alle (unmarkierten)
   - 2.1. Spalten, welche Nullen in markierten Zeilen haben.
   - 2.2. Zeilen, welche zugewiesene Nullen in markierten Spalten haben.
3. Zeichne Linien durch alle markierten Spalten und alle unmarkierten Zeilen



The Hungarian Method: Beispiel “Autohändler 1x1”

Vorbereitung

1. Wollen wir die totalen Kosten minimieren oder maximieren?
   1.1. Bei max: negiere alle Elemente; Addiere den ursprünglich größten Wert auf alle drauf
2. Liegt eine quadratische Matrix vor?
   2.1. Wenn nicht: ergänze zu n*n und fülle Elemente mit Null

Auto / Käufer	1	2	3	4
1	10	16	12	13
2	6	8	6	5
3	8	9	5	6

Preise in $x*10^3$, [7]

$$\begin{bmatrix} 10 & 16 & 12 & 13 \\ 6 & 8 & 6 & 5 \\ 8 & 9 & 5 & 6 \end{bmatrix} \xrightarrow{1.1} \begin{bmatrix} 6 & 0 & 4 & 3 \\ 10 & 8 & 10 & 11 \\ 8 & 7 & 11 & 10 \end{bmatrix} \xrightarrow{2.1} \begin{bmatrix} 6 & 0 & 4 & 3 \\ 10 & 8 & 10 & 11 \\ 8 & 7 & 11 & 10 \\ 0 & 0 & 0 & 0 \end{bmatrix} \longrightarrow$$

The Hungarian Method: Beispiel "Autohändler 1x1"

Ablauf

3. Für jede Zeile Z: subtrahiere das kleinste Element aus Z von jedem Element aus Z
4. Für jede Spalte S: subtrahiere das kleinste Element aus S von jedem Element aus S
5. Versuche alle Nullen mit einer minimalen Anzahl an horizontalen/vertikalen Linien L zu überdecken
   5.1. Wenn |L| = n: optimale Zuordnung existiert
   5.2. Wenn |L| < n: → Schritt 6
6. Finde das kleinste k ∉ L. Subtrahiere k von allen Elementen die nicht in L sind. Addiere k auf alle Elemente die in L doppelt vorkommen. → Schritt 5

$$\xrightarrow{3/(4)} \begin{bmatrix} 6 & 0 & 4 & 3 \\ 2 & 0 & 2 & 3 \\ 1 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{5} \begin{bmatrix} 6 & 0 & 4 & 3 \\ 2 & 0 & 2 & 3 \\ 1 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\xrightarrow{5.2 \to 6} \begin{bmatrix} 5 & 0 & 3 & 2 \\ 1 & 0 & 1 & 2 \\ 0 & 0 & 3 & 2 \\ 0 & 1 & 0 & 0 \end{bmatrix} \xrightarrow{6 \to 5.2 \to 6} \begin{bmatrix} 4 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 3 & 2 \\ 0 & 2 & 0 & 0 \end{bmatrix} \rightarrow$$

The Hungarian Method: Beispiel "Autohändler 1x1"

$$\begin{array}{cc} 6 \to 5.1 & \begin{bmatrix} 4 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 3 & 2 \\ 0 & 2 & 0 & 0 \end{bmatrix} \end{array}$$

Auto / Käufer	1	2	3	4
1	10	16	12	13
2	6	8	6	5
3	8	9	5	6

Der optimale Wert ist also:

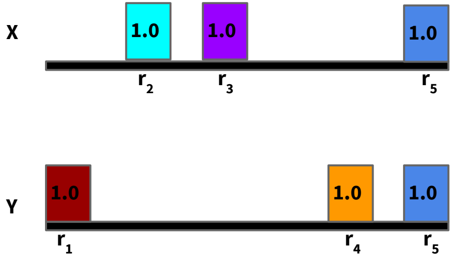
8k + 16k + 6k = 30.000€

The Hungarian Method: EMD

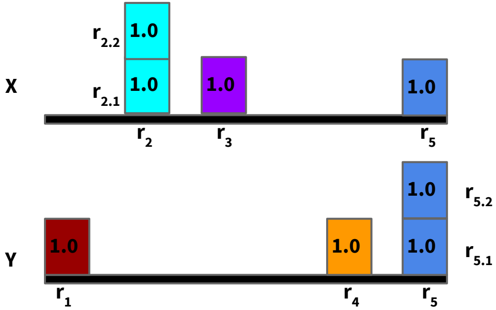
Berechnung EMD durch H.M.

- Fall 1: binäre Werte für Haufen/ Löcher → {1,0}
- Fall 2: Ganzzahlige Werte
- Fall 3: Reelzahlige Werte
    - Modellierbar durch z.B. 10 Buckets à 0.1

$$\begin{array}{c} \mathbf{Y} \rightarrow \\ \mathbf{X} \downarrow \end{array} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 4 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{bmatrix}$$



$$\begin{array}{c c} & \begin{array}{cccc} & & r_{5.1} & r_{5.2} \end{array} \\ \begin{array}{c} r_{2.1} \\ r_{2.2} \\ \\ \\ \end{array} & \begin{bmatrix} 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 3 \\ 2 & 1 & 2 & 2 \\ 4 & 1 & 0 & 0 \end{bmatrix} \end{array} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 6 & 2 & 0 & 0 \end{bmatrix}$$

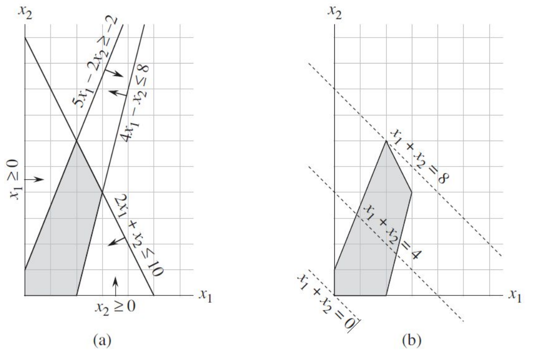


4.
Das Simplex-Verfahren

Das Simplex-Verfahren

Simplex - Grundlegendes Vorgehen

- "Lineare Programmierung"
- Eine Zielfunktion soll maximiert / minimiert werden
- Bedingungen schränken den zulässigen Bereich (Simplex, grau) an Lösungen ein
- Zulässige Lösung: das Wertepaar (X1, X2), welches alle NB erfüllt
- Schnittpunkt der Gerade mit maximalem Zielfunktionswert mit dem zulässigen Bereich → optimale Lösung



[3]

Maximiere:

Z = X1 + X2

Unter den Nebenbedingungen:

4 X1 - X2 ≤ 8

2 X1 + X2 ≤ 10

5 X1 - 2 X2 ≥ -2

X1, X2 ≥ 0

Das Simplex-Verfahren: EMD

Berechnung EMD durch Simplex

- Standard Simplex anwendbar
- Laufzeit in Spezialfällen exponentiell, in Praxis effizient
- Jeder mögliche Fluss durch Variable referenziert

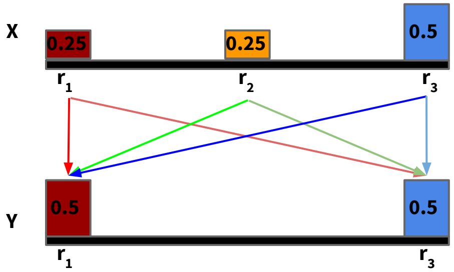
Lösung

Min:

X1 = 0.25, X2 = 0, X3 = 0.25, X4 = 0, X5 = 0, X6 = 0.5

(Max):

X1 = 0, X2 = 0.25, X3 = 0, X4 = 0.25, X5 = 0.5, X6 = 0



Minimiere:

$$Z = 0\,X1 + 0.5\,X2 + 0.25\,X3 + 0.25\,X4 + 1\,X5 + 0\,X6$$

Unter den Nebenbedingungen:

$$1\,X1 + 1\,X2 + 0\,X3 + 0\,X4 + 0\,X5 + 0\,X6 = 0.25$$

$$0\,X1 + 0\,X2 + 1\,X3 + 1\,X4 + 0\,X5 + 0\,X6 = 0.25$$

$$0\,X1 + 0\,X2 + 0\,X3 + 0\,X4 + 1\,X5 + 1\,X6 = 0.5$$

$$1\,X1 + 0\,X2 + 1\,X3 + 0\,X4 + 1\,X5 + 0\,X6 = 0.5$$

$$0\,X1 + 1\,X2 + 0\,X3 + 1\,X4 + 0\,X5 + 1\,X6 = 0.5$$

$$X1, X2, X3, X4, X5, X6 \geq 0$$

[8]

5. Zusammenfassung

Was nehmen wir mit?

Ungarischer Algorithmus → **Zuordnungsproblem**

Simplex Algorithmus → **Earth Mover’s Distance**

Quellen

[1]: https://web.eecs.umich.edu/~pettie/matching/Kuhn-hungarian-assignment.pdf
[2]: https://www.wolframalpha.com/input/?i=plot+x+and+x%5E2+and+2%5Ex+and+x!+with+x+from+0+to+10+and+y+from+0+to+5000
[3]: https://mitpress.mit.edu/books/introduction-algorithms-third-edition
[4]: https://www.hindawi.com/journals/jam/2013/749429/
[5]: http://www.mit.edu/~dimitrib/Orig_Auction.pdf
[6]: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/RUBNER/emd.htm
[7]: http://www.hungarianalgorithm.com/solve.php?c=10-16-12-13--6-8-6-5--8-9-5-6&obj=max&random=1
[8]:http://www.phpsimplex.com/simplex/page2.php?o=min&x2=0.5&x3=0.25&x4=0.25&x5=1&rt=5&v=6&l=en&r1_1=1&r1_2=1&y1=0.25&r2_3=1&r2_4=1&y2=0.25&r3_5=1&r3_6=1&y3=0.5&r4_1=1&r4_3=1&r4_5=1&y4=0.5&r5_2=1&r5_4=1&r5_6=1&y5=0.5&Submit=Continue#

Abbildungsverzeichnis (Abv.):
http://mathworld.wolfram.com/BipartiteGraph.html
https://www.researchgate.net/figure/The-Earth-Movers-Distance_fig2_220965753
https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
https://pngtree.com/free-icon/matrix_777567
http://www.dbs.ifi.lmu.de/Lehre/STMD/SS2013/Skript/Kap02-Anfragebearbeitung_Teil5.pdf

Ende

Vielen Dank für die Aufmerksamkeit!