Übung 11

Graph Analysis

11-1.1

Eine gute Partitionierung erfüllt zwei Bedingungen:

- maximiert die Anzahl der Kanten innerhalb der Gruppen
- minimiert die Anzahl der Kanten zwischen den Gruppen

11-1.1

- Partitionen sollen möglichst balanciert sein $\Rightarrow$ der Graph sollte entweder in $\{A,B,C\}$ und $\{D,E\}$ oder in $\{A,B\}$ und $\{C,D,E\}$ aufgeteilt werden
- Erste Möglichkeit hat mehr Kanten innerhalb der Gruppen und es muss nur eine Kante statt zwei durchschnitten werden
- $\Rightarrow$ Beste Partitionierung ist $\{A,B\}$ und $\{C,D,E\}$

11-1.2

Modularity of partitioning S of graph G:

$$Q \propto \sum_{s \in S} [(\#edges\ within\ group\ s) - (expected\ \#edges\ within\ group\ s)]$$

$$Q(G,S) = \underbrace{\frac{1}{2m}}_{\text{Normalizing: } -1 < Q < 1} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(a_{ij} - \frac{k_i k_j}{2m}\right)$$

11-1.2

Bereits bekannt:

- Anzahl der Knoten $|n| = 5$
- Anzahl der Kanten $|m| = 5$
- Grad der Knoten $k_A = 1, k_B = 2, k_C = 3, k_D = 2, k_E = 2$

Schritte um Q zu berechnen

- Berechne die Adjazenzmatrix
- Berechne die Modularitymatrix ($B_{ij} = A_{ij} - \frac{k_i \cdot k_j}{2m}$)
- Summiere die Einträge der einzelnen Cluster
- Summiere die Summen aller Cluster
- Normalisiere das Ergebnis

11-1.2

$$\text{Adjazenzmatrix } A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

11-1.2

$$\text{Modularitymatrix } B = \frac{1}{10}\begin{pmatrix} -1 & 8 & -3 & - & - \\ 8 & -4 & 4 & - & - \\ -3 & 4 & -9 & - & - \\ - & - & - & -4 & 6 \\ - & - & - & 6 & -4 \end{pmatrix}$$

11-1.2

$$s_1 = \{A, B, C\} \rightarrow \frac{1}{10} \cdot ((-1) + 8 - 3 + 8 - 4 + 4 - 3 + 4 - 9) = -\frac{4}{10}$$
$$s_2 = \{D, E\} \rightarrow \frac{1}{10} \cdot ((-4) + 6 + 6 - 4) = \frac{4}{10}$$
$$\sum_{s \in S} = \frac{4}{10} + \frac{4}{10} = \frac{8}{10} = 0.8$$
$$Q(G, S) = \frac{1}{10} \cdot (0.8) = (0.08)$$

11-1.3

$$\text{Adjazenzmatrix } A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

11-1.3

Modularitymatrix $B = \frac{1}{10}\begin{pmatrix} -1 & 8 & - & - & - \\ 8 & -4 & - & - & - \\ - & - & -9 & 4 & 4 \\ - & - & 4 & -4 & 6 \\ - & - & 4 & 6 & -4 \end{pmatrix}$

11-1.3

$$s_1 = \{A, B\} \rightarrow \frac{1}{10} \cdot ((-1) + 8 + 8 - 4) = \frac{11}{10}$$
$$s_2 = \{C, D, E\} \rightarrow \frac{1}{10} \cdot ((-9) + 4 + 4 + 4 - 4 + 6 + 4 + 6 + -4) = \frac{11}{10}$$
$$\sum_{s \in S} = \frac{11}{10} + \frac{11}{10} = \frac{22}{10}$$
$$Q(G, S) = \frac{1}{10} \cdot \frac{22}{10} = (0.22)$$

11-1.4

- Je höher die Modularität Q ist, desto besser ist die Partitionierung
- Das Entfernen der Kante $\{B, C\}$ ergibt ein höheres Q als das Entfernen der Kanten $\{C, E\}$ und $\{C, D\}$
- $\Rightarrow$ Die Vermutung aus 1, die auf dem Maximieren der Kanten innerhalb der Gruppen und Minimieren der Kanten zwischen den Gruppen liegt, war korrekt

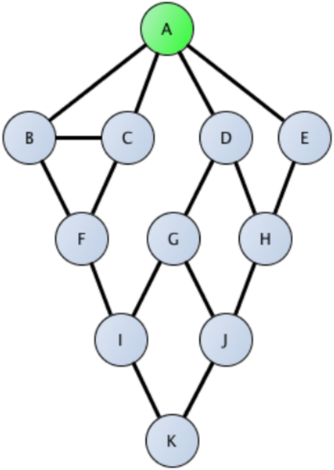
11-2

Girven-Newman Algorithmus:

1. Beginne bei Knoten A und führe eine Breitensuche durch und konstruiere einen DAG (directed, acyclic graph)
2. Zähle die Anzahl der kürzesten Pfade von A zu allen anderen Knoten
3. Berechne betweenness, indem man den Baum von unten nach oben durchgeht. Wenn es mehrere Pfade gibt, werden diese partiell gezählt
   - $\textit{node flow} = 1 + \sum \textit{childEdges}$
   - flow basierend auf den Elternwerten (kürzester Pfad) aufteilen

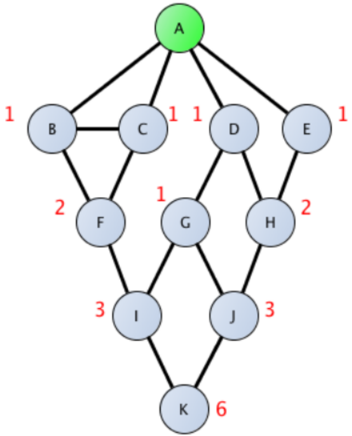
11-2

Schritt 1:



11-2

Schritt 2:



11-2

Schritt 3: