1 Layout von Graphen

Force-directed Methoden benutzen die Analogie zur Physik, um die Knoten eines Graphen zu positionieren. Wir betrachten einen ungerichteten Graphen $G = (V, E)$ als ein System von Körpern (V), auf die Kräfte (E) einwirken. Der Layout-Algorithmus sucht nach einer Konfiguration, in der die Gesamtsumme der Kräfte minimiert wird, d.h. er ordnet jedem Knoten eine Position zu, so dass die Summe der Kräfte, die auf jeden Knoten einwirken, gleich 0 ist.

Im Allgemeinen bestehen force-directed Layout-Algorithmen aus zwei Teilen: einem *Modell* und dem *Algorithmus*.

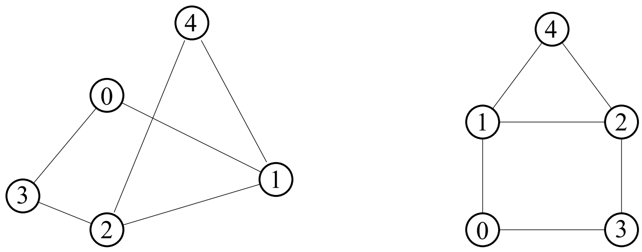

Abbildung 1: Beispiel

2 Modell: Spring-Layout

Beim Spring-Layout verwendet man als Model eine Kombination aus Federn und elektrischen Kräften. Die Kanten werden als Federn und die Knoten als sich abstoßend geladene Körper modelliert.

Im Folgenden bezeichnen wir mit $u, v \in V$ die Knoten und mit $\vec{u}, \vec{v} \in \mathbb{R}^2$ deren Ortsvektoren in der zweidimensionalen Zeichnung. Mit

$$||\vec{v} - \vec{u}|| = \sqrt{(x_v - x_u)^2 + (y_v - y_u)^2}$$

bezeichnen wir die Länge des Differenzvektor $\vec{v} - \vec{u}$, d.h. der euklidische Abstand der Punkte (x_u, y_u) und (x_v, y_v). Desweiteren bezeichnet $\vec{e}_{uv}$ den Einheitsvektor

$$\vec{e}_{uv} = \frac{\vec{v} - \vec{u}}{||\vec{v} - \vec{u}||},$$

der von $\vec{u}$ nach $\vec{v}$ zeigt.

Die abstoßende Kraft die von u auf v wirkt, folgt der Gleichung[1]

$$\vec{F_0}(u, v) = \frac{c_0}{||\vec{v} - \vec{u}||^2} \cdot \vec{e}_{uv},$$

[1]Coulombsches Gesetz: $\vec{F} = \frac{1}{4\pi\epsilon_0} \cdot \frac{q_1 \cdot q_2}{r^2} \cdot \vec{e_r}$

wobei $c_0 \in \mathbb{R}$ eine Konstante ist, die angibt, wie stark sich zwei Knoten abstoßen.
Die Federkraft, die durch u auf v wirkt, lässt sich durch die Gleichung[2]

$$\vec{F}_1(u,v) = -c_1(||\vec{v} - \vec{u}|| - l) \cdot \vec{e}_{uv}$$

berechnen, wobei die Konstante $c_1 \in \mathbb{R}$ die Federkraft und $l \in \mathbb{R}$ die natürliche Länge der Federn angibt.
Die Gesamt-Kraft, die auf einen Knoten $v \in V$ einwirkt, lässt sich folgendermaßen bestimmen:

$$\vec{F}(v) = \sum_{u \in V} \vec{F}_0(u,v) + \sum_{(u,v) \in E} \vec{F}_1(u,v).$$

3 Algorithmus

Auf Knoten, die noch nicht im Kräftegleichgewicht sind, wirkt eine Kraft. Um das System zu entspannen, werden die Knoten iterativ in Richtung der auf sie einwirkenden Kraft verschoben. Zum Zeitpunkt t wirkt auf einen Knoten v die Kraft $\vec{F}_t(v)$. Nach Berechnung aller Kräfte $\vec{F}_t(v)$ für alle $v \in V$ zum Zeitpunkt t, wird jeder Knoten v um $\delta \cdot \vec{F}_t(v)$ verschoben. Die Konstante δ, $0 < \delta < 1$, verhindert, dass Knoten übermäßig weit verschoben werden.

Algorithmus: Spring-Embedder

`WHILE (...) DO`
    `FOR` $u \in V$ `DO`
        $\vec{F}(u) = \sum_{v \in V} \vec{F}_0(u,v) + \sum_{(u,v) \in E} \vec{F}_1(u,v)$
    `FOR` $u \in V$ `DO`
        $\vec{u} \leftarrow \vec{u} + \delta \cdot \vec{F}(u)$
`OD`

3.1 Abbruchbedingung

Die einfachste Möglichkeit ist es, nach einer bestimmten Anzahl von Iterationen abzubrechen. Eine weitere Möglichkeit ist es, die globale Kraft zu einem Zeitpunkt t im System zu bestimmen. Diese kann folgendermaßen berechnet werden:

$$F_{\text{global}} = \sum_{v \in V} |\vec{F}_t(v)|.$$

Falls F_{global} einen bestimmten Wert erreicht/unterschreitet (im Idealfall $F_{\text{global}} = 0$), kann mit der Berechnung abgebrochen werden.

[2]Hookesches Gesetz: $\vec{F} = -D \cdot \vec{s}$

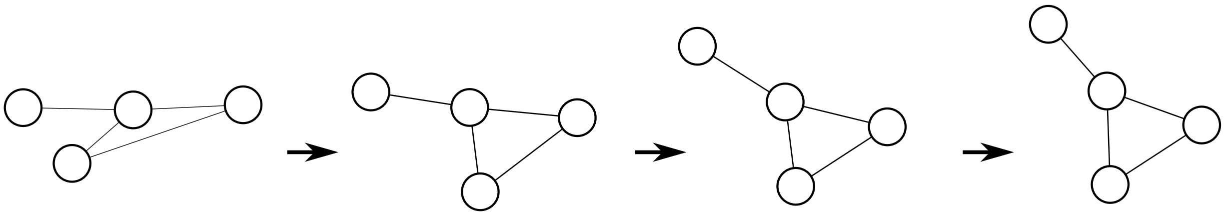

Abbildung 2: Ablauf des Spring-Embedders

4 Beschleunigung der Berechnungen: Clipping

Zur Berechnung der Kraft $\vec{F}(u)$ eines Knotens $u \in V$ wird $\sum_{v \in V} \vec{F_0}(u, v)$ bestimmt. Diese Berechnung wird pro Iteration $n(n-1) = O(n^2)$-mal durchgeführt. Knoten, die weit von einander entfernt sind, stoßen sich allerdings nur sehr gering ab. Aus diesem Grund kann man bei der Berechnung der abstoßenden Kraft diese weit entfernten Knoten vernachlässigen. Die Kraft $\vec{F_0}(u, v)$ wird nur für Knoten v berechnet, für deren Abstand zu u gilt, dass $||\vec{v} - \vec{u}||^2 < d^2$ (für eine geeignete Konstante $d \in \mathbb{R}$).

5 Nicht zusammenhängende Graphen

Der gegebene Algorithmus funktioniert nur auf zusammenhängenden Graphen G. Besteht der Graph G aus mehreren Zusammenhangs-Komponenten $Z_1, ..., Z_n$, so driften die einzelnen Zusammenhangs-Komponenten Z_i in jedem Iterationsschritt auseinander und es stellt sich kein Kräftegleichgewicht ein. Dieses Problem lässt sich auf zwei unterschiedliche Arten lösen:

- Man fügt einen weiteren Knoten w ein, und verbindet diesen entweder mit *jedem Knoten* $v \in G$ oder mit *je einem Knoten* v_i *in jeder Zusammenhangs-Komponente* Z_i. Nach der Berechnung des Layouts entfernt man wieder diesen Knoten w und die entsprechenden Kanten.

- Die Ränder des Zeichenbereiches erhalten eine Ladung, so dass die Knoten abgestoßen werden.

6 Berechnung

Anstatt in der Implementierung mit Vektoren zu rechnen, ist es gegebenenfalls einfacher, die Kräfte jeweils in die x und y-Komponente aufzuteilen. Diese kann man dann separat berechnen. Die Kraft F_0 in x-Richtung lautet beispielsweise:

$$F_{0,x}(u, v) = \frac{c_0}{(x_v - x_u)^2 + (y_v - y_u)^2} \cdot \frac{x_v - x_u}{\sqrt{(x_v - x_u)^2 + (y_v - y_u)^2}}$$

Analog lassen sich auch die Werte für $F_{0,y}$, $F_{1,x}$ und $F_{1,y}$ berechnen.

Literatur

[1] Ulrik Brandes. Drawing on Physical Analogies. Drawing Graphs. LNCS 2025: 71–86, 2001.

[2] Guiseppe di Battista and Peter Eades and Roberto Tamassia and Ioannis G. Tollis. Graph Drawing. Algorithms for the Visualizaition of Graphs. 303–325.