

Teil 5: Algorithmen auf Graphen

Transitive Hülle

Die transitive Hülle

In einigen Anwendungen genügt es für einen gerichteten Graphen zu wissen, ob ein Knoten von einem anderen aus erreichbar ist.

Diese Fragen können leicht mittels der **transitiven Hülle** G^* eines gerichteten Graphen G beantwortet werden. Dabei ist G^* der folgende Graph:

$$G^* := (V, E^*) \text{ mit } E^* := \{(u, v) \mid \text{es existiert ein Weg von } u \text{ nach } v\}.$$

Im Folgenden wollen wir nun beschreiben, wie die transitive Hülle eines Graphen berechnet werden kann.

Die Adjazenzmatrix und die transitive Hülle

$G = (V, E)$ sei ein gerichteter Graph mit $V = \{1, \dots, n\}$ und A_G seine Adjazenzmatrix, d.h.

$$A_G = (a_{ij}) \text{ mit } a_{ij} = \begin{cases} 1 & \text{falls } (i, j) \in E \\ 0 & \text{sonst.} \end{cases}$$

A_G^k sei die k -te Potenz von A_G , d.h.

$$A_G^0 := I_n \text{ und } A_G^k = A_G \cdot A_G^{k-1}.$$

Dabei ist I_n die **n -te Einheitsmatrix**, d.h.

$$I_n = (\delta_{ij}) \text{ mit } \delta_{ij} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j. \end{cases}$$

Lemma

Für $k \geq 0$ sei $A_G^k = (a_{ij}^{(k)})$. Dann ist $a_{ij}^{(k)}$ die Anzahl der Wege der Länge k von i nach j .

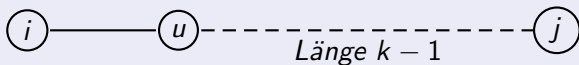
Beweis

Da es offensichtlich nur genau einen Weg der Länge 0 von einem Knoten zu sich selbst gibt, ist die Behauptung für $k = 0$ korrekt.

Da Wege der Länge 1 genau Kanten entsprechen, ist die Behauptung auch für $k = 1$ korrekt.

Sei nun $k > 1$ und die Behauptung sei für $k - 1$ erfüllt, d.h. $a_{ij}^{(k-1)}$ ist die Anzahl aller Wege der Länge $k - 1$ von i nach j .

Hat man einen Weg von u nach j der Länge l , so ergibt sich durch eine Kante (i, u) ein Weg der Länge $l + 1$ von i nach j .



Beweis (Die Potenzen der Adjazenzmatrix - Fortsetzung)

Damit ist

$$\sum_{u=1}^n a_{iu} \cdot a_{uj}^{(k-1)}.$$

Die Anzahl aller Wege von i nach j mit Länge k .

Weiter ergibt sich A_G^k als

$$A_G^k = A_G \cdot A_G^{k-1} = \left(a_{ij}^{(k)} \right)$$

mit

$$a_{ij}^{(k)} = \sum_{u=1}^n a_{iu} \cdot a_{uj}^{(k-1)}.$$



Die transitive Hülle und die Potenzen der Adjazenzmatrix

Um die transitive Hülle G^* zu berechnen müssen wir somit alle Einträge der Summe

$$A_G^* := A_G^0 + A_G^1 + A_G^2 \cdots = \sum_{k=0}^{\infty} A_G^k$$

die nicht Null sind, bestimmen.

Allerdings reicht bereits die Summe bis $k = n$ aus, denn wenn es einen Weg von i nach j in A gibt, dann gibt es auch einen **einfachen** Weg. Da G aber nur n Knoten hat, hat ein einfacher Weg höchstens die Länge n .

Somit gilt

$$a_{ij}^* \neq 0 \quad \Leftrightarrow \quad a_{ij}^{n-1} \neq 0$$

und wir müssen „nur“ $A^0 + \cdots + A^n$ berechnen.

Die erweiterte Adjazenzmatrix

Wir können uns das Leben aber noch weiter vereinfachen, indem wir die **erweiterte Adjazenzmatrix** $\hat{A} := A_G + I_n$ verwenden.

Lemma

Sei $k \geq 1$ und $\hat{A}^k := (\hat{a}_{ij}^{(k)})$. Dann gilt

$$\hat{a}_{ij}^{(k)} \neq 0 \quad \Leftrightarrow \quad a_{ij}^{(l)} \neq 0 \text{ für ein } l \leq k.$$

Beweis

Offensichtlich gilt $\hat{A} = A_G^0 + A_G^1$, was die Behauptung für $k = 1$ beweist.

Die Behauptung gelte nun für $k \geq 1$. Da alle Einträge von \hat{A} und \hat{A}^k nicht negativ sind, gilt $\hat{a}_{ij}^{(k+1)} = \sum_{u=1}^n \hat{a}_{iu} \hat{a}_{uj}^{(k)} \neq 0$ genau dann, wenn ein u existiert mit $\hat{a}_{iu} \neq 0$ und $\hat{a}_{uj}^{(k)} \neq 0$.

Beweis (Fortsetzung)

Dies ist nach Definition von \hat{A} und Induktionsvoraussetzung genau dann erfüllt, wenn $i = u$ oder $(i, u) \in E$ und ein $l \leq k$ existiert mit $a_{uj}^{(l)} \neq 0$.

Dies ist wiederum genau dann erfüllt, wenn $a_{uj}^{(l)} \neq 0$ oder $a_{uj}^{(l+1)} \neq 0$ für ein $l \leq k$. □

Das Lemma zeigt, dass es genügt die n -te Potenz \hat{A}^n der erweiterten Adjazenzmatrix \hat{A} zu betrachten um die transitive Hülle zu erhalten.

Weiterhin ist für uns nur von Interesse, ob der entsprechende Eintrag von \hat{A}^n gleich 0 ist oder nicht.

Damit stellt sich die Frage, ob sich die Berechnung von \hat{A}^n vereinfachen lässt.

Eine „andere“ Matrizenmultiplikation

Wie wir gesehen haben, müssen wir Ausdrücke der Form

$$\hat{a}_{ij}^{(k+1)} = \sum_{u=1}^n \hat{a}_{iu}^{(k)} \cdot \hat{a}_{uj}$$

berechnen. Dabei interessiert uns nur, ob dieser Ausdruck gleich 0 ist oder nicht.

Setzen wir

$$\bar{a}_{ij}^{(k)} := \begin{cases} 1 & \text{falls } \hat{a}_{ij}^{(k)} \neq 0 \\ 0 & \text{falls } \hat{a}_{ij}^{(k)} = 0, \end{cases}$$

müssen wir lediglich erreichen, dass folgendes gilt:

$$\bar{a}_{ij}^{(k+1)} = \begin{cases} 1 & \text{falls ein } u \text{ existiert mit } \bar{a}_{iu}^{(k)} = 1 \text{ und } \bar{a}_{uj} = 1 \\ 0 & \text{sonst} \end{cases}$$

Eine „andere“ Matrizenmultiplikation

Dies ist durch die Verwendung des **Booleschen Halbringes** möglich. Konkret heißt dies, dass wir statt der Addition das **logische Oder** \vee und statt der Multiplikation das **logische Und** \wedge verwenden. D.h.

$$\bar{a}_{ij}^{(k+1)} = \bigvee_{u=1}^n \bar{a}_{iu}^{(k)} \wedge \bar{a}_{uj}.$$

Damit ergibt sich entsprechend dem Beweis des obigen Lemma:

$$\bar{a}_{ij}^{(k)} = \begin{cases} 1 & \text{falls ein Weg der Länge } \leq k \text{ von } i \text{ nach } j \text{ existiert} \\ 0 & \text{sonst.} \end{cases}$$

Den Kostenaufwand für die Multiplikation zweier $n \times n$ -Matrizen können wir leicht als $O(n^3)$ abschätzen, da die Berechnung jedes Eintrags $O(n)$ benötigt.

Beschleunigung der Berechnung

Wir haben bewiesen, dass die Adjazenzmatrix $A_G^* := A_{G^*}$ der transitiven Hülle G^* eines gerichteten Graphen $G = (V, E)$ gleich der n -ten Potenz \hat{A}^n der erweiterten Adjazenzmatrix im Booleschen Ring ist (d.h. Addition = Oder, Multiplikation = Und).

Mit dem Aufwand $O(n^3)$ für eine Multiplikation, erhalten wir den Gesamtaufwand von $O(n^4)$ für die Berechnung von \bar{A}^n .

Können wir zur weiteren Beschleunigung ausnutzen, dass wir nicht das Produkt zweier beliebiger Matrizen berechnen müssen, sondern eine Matrix immer wieder mit ihren eigenen Potenzen multiplizieren?

Die Antwort lautet: JA!

Beschleunigung der Berechnung

Um dies zu sehen, benötigen wir zwei Beobachtungen:

- 1 $\bar{A}^k \cdot \bar{A}^k = \bar{A}^{2k}$
- 2 $\bar{A}^k = \bar{A}^n$ für $k \geq n$, denn die einfachen Wege haben höchstens Länge n .

Damit ergibt sich insbesondere

$$\bar{A}^n = \bar{A}^{2^{\lceil \log n \rceil}} \quad \text{und} \quad \bar{A}^{2^k} \cdot \bar{A}^{2^k} = \bar{A}^{2^{k+1}}.$$

Damit kann \bar{A}^n mit $\lceil \log 2 \rceil$ Matrizenmultiplikationen berechnet werden, was zu einer Laufzeit von $O(n^3 \log n)$ führt.

Teil 5: Algorithmen auf Graphen
Transitive Hülle

Der Algorithmus von Warshall

Das Verfahren von Warshall

Die Berechnung der transitiven Hülle kann noch weiter beschleunigt werden, in dem man statt aller möglichen Wege nur Wege betrachtet, die auf eine bestimmte Art zusammen gesetzt sind.

Wir betrachten die Matrizen $A^{(k)} = \left(a_{ij}^{(k)} \right)$ mit

$$a_{ij}^{(k)} := \begin{cases} 1 & \text{falls es einen Weg von } i \text{ nach } j \text{ gibt, der nur durch Knoten} \\ & \text{aus } \{1, \dots, k\} \text{ verläuft, } i \text{ und } j \text{ ausgenommen,} \\ 0 & \text{sonst.} \end{cases}$$

Also gilt $a_{ij}^{(k)} = 1$ genau dann, wenn ein Weg (i, v_1, \dots, v_l, j) existiert mit $v_1, \dots, v_l \in \{1, \dots, k\}$.

Insbesondere ergibt sich

$$a_{ij}^{(n)} = 1 \quad \Leftrightarrow \quad \text{Es existiert ein Weg von } i \text{ nach } j$$

Das Verfahren von Warshall

Offensichtlich gilt $A^{(0)} = A_G + I_n$, denn für $k = 0$ sind nur Wege ohne Zwischenknoten erlaubt.

Für $k > 0$ ergibt sich außerdem, dass genau dann ein Weg von i nach j durch die Knoten $\{1, \dots, k\}$ existiert, wenn

- ein Weg von i nach j durch die Knoten $\{1, \dots, k - 1\}$ existiert,
- oder ein Weg von i nach j existiert, der einmal den Knoten k durchläuft und sonst nur die Knoten $\{1, \dots, k - 1\}$ (betrachte nur einfache Wege).

Damit ergibt sich

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} \vee \left(a_{ik}^{(k-1)} \wedge a_{kj}^{(k-1)} \right).$$

Diese Gleichung kann für den folgenden Algorithmus genutzt werden.

Algorithmus (Verfahren von Warshall)

Eingabe: Eine Adjazenzmatrix A_G eines gerichteten Graphen $G = (V, E)$.

Ausgabe: Die Adjazenzmatrix A der transitiven Hülle G^* .

$A := A_G$

für $i := 1, \dots, n$ **tue** $A[i, i] := 1$

// $A = A_G + I_n$

für $k := 1, \dots, n$ **tue**

// n Durchläufe

für $i := 1, \dots, n$ **tue**

// n Durchläufe

wenn $A[i, k] = 1$ **dann**

für $j := 1, \dots, n$ **tue**

// max. n Durchläufe

wenn $A[k, j] = 1$ **dann** $A[i, j] := 1$

Ende

Ende

Ende

Ende

Korrektheit des Verfahrens von Warshall

Der Algorithmus implementiert nicht ganz die oben beschriebene Methode um $A^{(k)}$ zu berechnen.

Tatsächlich hat A in der k -ten Runde vor der Behandlung der i -ten Zeile die folgende Form:

Korrektheit des Verfahrens von Warshall

Offensichtlich gilt vor dem Eintritt in die äußerste Schleife $A = A^{(1)}$.
Wird jetzt in der k -ten Runde die i -te Zeile bearbeitet, ergibt sich anschließend

$$A[i, j] = \begin{cases} a_{ij}^{(k-1)} \vee \left(a_{ik}^{(k)} \wedge a_{kj}^{(k)} \right) & \text{falls } j < i \text{ und } i > k \\ a_{ij}^{(k-1)} \vee \left(a_{ik}^{(k)} \wedge a_{kj}^{(k-1)} \right) & \text{falls } j \geq i \text{ und } i > k \\ a_{ij}^{(k-1)} \vee \left(a_{ik}^{(k-1)} \wedge a_{kj}^{(k)} \right) & \text{falls } j < i \text{ und } i \leq k \\ a_{ij}^{(k-1)} \vee \left(a_{ik}^{(k-1)} \wedge a_{kj}^{(k-1)} \right) & \text{falls } j \geq i \text{ und } i \leq k \end{cases}$$

Weiterhin gilt:

$$a_{ik}^{(k)} = a_{ik}^{(k-1)} \vee \left(a_{ik}^{(k-1)} \wedge a_{kk}^{(k-1)} \right) = a_{ik}^{(k-1)}$$

$$a_{kj}^{(k)} = a_{kj}^{(k-1)} \vee \left(a_{kk}^{(k-1)} \wedge a_{kj}^{(k-1)} \right) = a_{kj}^{(k-1)}$$

Damit ergibt sich nach der Behandlung der i -ten Zeile in der k -ten Runde:

$$A[i, j] = a_{ij}^{(k-1)} \vee \left(a_{ik}^{(k-1)} \wedge a_{kj}^{(k-1)} \right) = a_{ij}^{(k)}.$$

Laufzeit

Da wir drei ineinander geschachtelte Schleifen mit jeweils maximal n Durchläufen haben, ergibt sich sofort eine obere Laufzeitschranke von $O(n^3)$.

Allerdings wird die innerste Schleife nicht jedesmal durchgeführt, sondern nur, wenn $A[i, k] = 1$ gilt, d.h. wenn (i, k) als Kante in der transitiven Hülle auftritt.

Damit wird die innerste Schleife höchstens $n^* := |E^*|$ -mal ausgeführt.

Die Gesamtlaufzeit ergibt sich somit als $O(n^2 + n \cdot n^*)$.

Teil 5: Algorithmen auf Graphen
Transitive Hülle

Azyklische Graphen

Azyklische Graphen

Für azyklische Graphen kann die transitive Hülle noch schneller ermittelt werden. Sind die Knoten topologisch geordnet, d.h.

$$(i, j) \in E \Rightarrow i \leq j,$$

kann ein Knoten j höchstens dann von i aus erreicht werden, wenn $i \leq j$ gilt.

Diese Beobachtung kann zu einer Beschleunigung des Verfahrens von Warshall genutzt werden.

Algorithmus (Transitive Hülle azyklischer Graphen)

Eingabe: Eine Adjazenzmatrix A_G eines azyklischen, gerichteten Graphen $G = (V, E)$.

Ausgabe: Die Adjazenzmatrix A der transitiven Hülle G^* .

Voraussetzung: Die Knoten von G sind topologisch sortiert.

$A := A_G$

für $i := 1, \dots, n$ **tue** $A[i, i] := 1$ // $A = A_G + I_n$

für $i := 1, \dots, n$ **tue** // n Durchläufe

für $k := 1, \dots, i - 1$ **tue** // $i - 1 \leq n$ Durchläufe

wenn $(k, i) \in E$ **dann**

für $j := 1, \dots, k - 1$ **tue** // max. n Durchläufe

wenn $A[j, k] = 1$ **dann** $A[j, i] := 1$

Ende

Ende

Ende

Ende

Laufzeit

Die innerste Schleife wird für jede Kante einmal ausgeführt. Jede Ausführung besteht aus maximal n Durchläufen, so dass die innere Schleife insgesamt höchstens $O(|V| \cdot |E|)$ Zeit benötigt.

Hinzu kommen die Durchläufe der beiden äußeren Schleifen (ohne die Innere Schleife). Dort ergeben sich insgesamt

$$\sum_{i=1}^n i - 1 = \frac{i(i-1)}{2} = O(|V|^2)$$

Durchläufe.

Insgesamt lässt sich somit die transitive Hülle eines azyklischen, ungerichteten Graphen in Zeit $O(|V|^2 + |V| \cdot |E|)$ berechnen.