# gewichteter Graph

## Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

## Zusammenfassung

Ein gewichteter Graph ist ein Graph, dessen Kanten nichtnegative Zahlen, die Kantengewichte, tragen, die messen, wie stark zwei Knoten miteinander verbunden sind. Er besteht aus einer endlichen Menge von Knoten, einer Menge ungerichteter Kanten, die Paare verschiedener Knoten verbinden, und einem positiven Gewicht für jede Kante; ein Knotenpaar, das keine Kante bildet, hat das Gewicht null. Die Gewichte bilden die Gewichtsmatrix $\mathbf{A}$, eine symmetrische Matrix mit Nulldiagonale, die zugleich die Adjazenzmatrix des Graphen ist, und die Summe einer Zeile ist der gewichtete Knotengrad ihres Knotens. Im maschinelles Lernen (ML) sind die Knoten häufig die Datenpunkte eines Datensatzs und das Gewicht einer Kante ist die Ähnlichkeit zweier Datenpunkte, z. B. eine Gauß-Funktion der euklidischer Abstand zwischen ihren Merkmalsvektoren, beibehalten für die $k$ nächsten Nachbarn jedes Datenpunkts. Das Graph-Clustering sucht Teilmengen von Knoten, deren ausgehende Kanten wenig Gewicht tragen. Das Transfer-Learning und das föderiertes Lernen (FL) nutzen die Gewichte eines Graphen von Lernaufgaben als Koeffizienten eines Strafterms, der die Hypothesen benachbarter Aufgaben koppelt.

## Definition

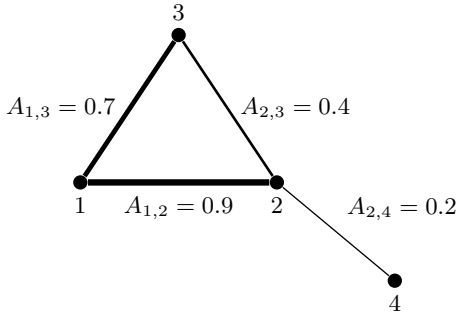
Betrachtet seien zehn Tage des Jahres 2024 in Krems an der Donau (Österreich), jeder 37. Tag vom 1. Januar bis zum 29. November, jeweils ein Datenpunkt mit einem Merkmalsvektor $\mathbf{x} \in \mathbb{R}^2$, der die Lufttemperatur um 06:00 Uhr und um 15:00 Uhr (UTC) des jeweiligen Tages enthält. Zwei Tage sind sich ähnlich, wenn ihre Merkmalsvektoren nahe beieinander liegen. Eine Zeichnung hält die Ähnlichkeiten fest: Jeder Tag ist ein Punkt, Knoten genannt, zwei ähnliche Tage sind durch eine Linie, Kante genannt, verbunden, und eine Zahl an der Kante gibt an, wie ähnlich sich die beiden Tage sind. Abb. 2(a) zeigt diese Zeichnung für die zehn Tage, mit einer Kante zwischen zwei Tagen immer dann, wenn einer von ihnen zu den drei dem anderen am nächsten gelegenen Tagen gehört, und einer Zahl zwischen 0 und 1, die mit der euklidischer Abstand zwischen ihnen abfällt. Knoten, Kanten und eine Zahl an jeder Kante bilden einen gewichteten Graphen (von Luxburg, 2007, Abschn. 2.1; Newman, 2010).

Formal besteht ein gewichteter Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ aus einer endlichen Menge von Knoten $\mathcal{V} = \{1, \ldots, n\}$, einer Menge $\mathcal{E}$ von Kanten und einem Kanten-

gewicht für jede Kante. Jede Kante ist ein ungeordnetes Paar $\{i, i'\}$ zweier verschiedener Knoten, sodass der Graph ein ungerichteter Graph ohne Selbstschleifen ist und die beiden Knoten einer Kante Nachbarn voneinander sind. Das Gewicht einer Kante $\{i, i'\} \in \mathcal{E}$ ist eine positive Zahl $A_{i,i'} > 0$, und ein Knotenpaar, das keine Kante bildet, hat das Gewicht $A_{i,i'} = 0$ (Chung, 1997, Abschn. 1.4). Mit dieser Konvention bestimmen die Gewichte die Kantenmenge, und sie werden in der Gewichtsmatrix zusammengefasst:

$$\mathbf{A} \in \mathbb{R}^{n \times n} \text{ mit Einträgen } A_{i,i'} = A_{i',i} \geq 0 \text{ und } A_{i,i} = 0 \text{ für } i, i' \in \mathcal{V}. \tag{1}$$

Die Gewichtsmatrix (1) ist eine symmetrische Matrix mit Nulldiagonale. Sie ist die Adjazenzmatrix des Graphen: Der Eintrag in Zeile $i$ und Spalte $i'$ ist das Gewicht der Kante zwischen $i$ und $i'$ oder null, wenn keine Kante vorliegt (Hastie et al., 2009, Abschn. 14.5.3; von Luxburg, 2007, Abschn. 2.1). Ein ungewichteter Graph ist der Spezialfall, in dem jede Kante das Gewicht 1 hat, sodass die Einträge seiner Gewichtsmatrix (1) gleich 0 und 1 sind (Chung, 1997, Abschn. 1.4). Abb. 1 zeigt einen gewichteten Graphen mit vier Knoten neben seiner Gewichtsmatrix.




$$\mathbf{A} = \begin{pmatrix} 0 & 0.9 & 0.7 & 0 \\ 0.9 & 0 & 0.4 & 0.2 \\ 0.7 & 0.4 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \end{pmatrix}$$

Abbildung 1: Ein gewichteter Graph mit vier Knoten $\mathcal{V} = \{1, 2, 3, 4\}$ und vier Kanten, jeweils als Linie gezeichnet, deren Breite mit ihrem Gewicht wächst, neben seiner Gewichtsmatrix (1). Der Eintrag in Zeile 2 und Spalte 4 ist das Gewicht 0.2 der Kante $\{2, 4\}$, und die Null in Zeile 1 und Spalte 4 hält fest, dass die Knoten 1 und 4 nicht durch eine Kante verbunden sind

Die Summe der Gewichte der Kanten an einem Knoten ist sein gewichteter Knotengrad,

$$d^{(i)} = \sum_{i' \in \mathcal{V}} A_{i,i'}, \tag{2}$$

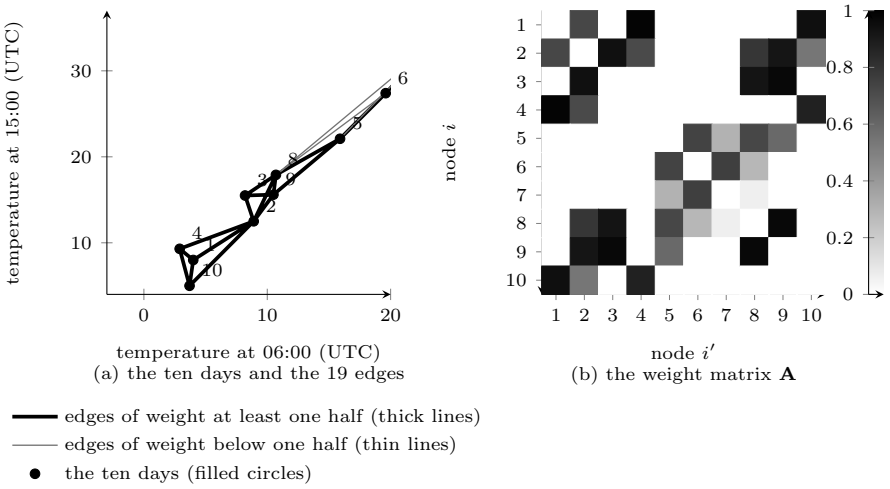
was der Summe der Zeile $i$ der Gewichtsmatrix entspricht (Chung, 1997, Abschn. 1.4; von Luxburg, 2007, Abschn. 2.1). Für einen ungewichteten Graphen zählt der Knotengrad (2) die Nachbarn des Knotens. Die Laplace-Matrix des Graphen wird aus der Gewichtsmatrix und den Graden (2) gebildet, und ihre Eigenvektoren sind das, was das Spektrales Clustering berechnet (von Luxburg, 2007, Abschn. 3).

Was ein Gewicht bedeutet, wird durch die Anwendung festgelegt, wobei zwei Konventionen gebräuchlich sind. In einem Straßennetz sind die Knoten Kreuzungen, die Kanten Straßenabschnitte, und das Gewicht einer Kante ist die Länge des Abschnitts oder die für seine Durchfahrt benötigte Zeit (Newman, 2010). Ein großes Gewicht bedeutet dann weit voneinander entfernt, und die kürzeste Route zwischen zwei Kreuzungen ist der Pfad, dessen Gewichte sich zum kleinsten Wert summieren. Im maschinelles Lernen (ML) misst das Gewicht einer Kante üblicherweise Ähnlichkeit: Ein großes Gewicht bedeutet nahe beieinander. Ein Verfahren, das eine Ähnlichkeit als Länge liest oder eine Länge als Ähnlichkeit, kehrt jede gezogene Schlussfolgerung um, weshalb die Konvention angegeben werden muss. Der Graph in Abb. 2(a) verwendet die Ähnlichkeitskonvention.

Die Hauptanwendung eines gewichteten Graphen im ML ist der Ähnlichkeitsgraph eines Datensatzs $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$. Seine Knoten sind die Datenpunkte, $\mathcal{V} = \{1, \ldots, m\}$, wobei der Knoten $r$ für $\mathbf{x}^{(r)}$ steht, und das Gewicht einer Kante misst die Ähnlichkeit ihrer beiden Datenpunkte. Für Merkmalsvektoren im $\mathbb{R}^d$ ist eine gängige Wahl das Gauß-Gewicht

$$A_{r,r'} = \exp\Big( - \Big\| \mathbf{x}^{(r)} - \mathbf{x}^{(r')} \Big\|_2^2 / (2\sigma^2) \Big), \tag{3}$$

das für zwei identische Merkmalsvektoren gleich 1 ist und mit deren euklidischer Abstand mit einer durch die Breite $\sigma > 0$ festgelegten Rate abfällt (Hastie et al., 2009, Abschn. 14.5.3; von Luxburg, 2007, Abschn. 2.2; Shalev-Shwartz and Ben-David, 2014, Abschn. 22.3). Weist man jedem Paar von Datenpunkte das Gauß-Gewicht (3) zu, entsteht ein vollständig verbundener Graph mit $m(m-1)/2$ Kanten, dessen Gewichtsmatrix außerhalb der Diagonalen keine Nullen aufweist. Eine dünnbesetzte Alternative ist der k-Nächste-Nachbarn-Graph: Eine Kante verbindet $r$ und $r'$, wenn einer der beiden Datenpunkte zu den $k$ nächsten Datenpunkte des anderen gehört; sie trägt das Gauß-Gewicht (3), und jedes andere Paar hat das Gewicht null (von Luxburg, 2007, Abschn. 2.2). Die Breite $\sigma$ und die Anzahl $k$ der Nachbarn sind Hyperparameter der Konstruktion; eine übliche Wahl für $\sigma$ ist die mittlere Abstand eines Datenpunkts zu seinem $k$-ten nächsten Nachbar (von Luxburg, 2007, Abschn. 8.1). Für die zehn Tage in Krems und $k = 3$ liefert diese Wahl $\sigma = 8.16$ Grad, und 19 der 45 Tagespaare sind Kanten. Ihre Gewichte reichen von 0.06 bis 0.98, zwischen dem 16. September und dem 23. Oktober, und die gewichteten Grade (2) von 1.11, am 11. August, bis 4.59, am 8. Februar. Abb. 2(b) zeigt die Gewichtsmatrix.




(a) the ten days and the 19 edges (b) the weight matrix $\mathbf{A}$

Abbildung 2: (a) Zehn der 365 Tage des Jahres 2024 in Krems, jeder 37 Tag, in datumsweiser Reihenfolge von 1 bis 10 nummeriert (Knoten 1 ist der 1. Januar, Knoten 10 der 30. November), als die Knoten ihres 3-Nächste-Nachbarn-Graphen, gezeichnet in der Ebene des 06:00- und des 15:00-Messwerts: Eine Kante mit einem Gewicht von mindestens 0.5 ist eine dicke Linie, eine Kante mit einem Gewicht unter 0.5 eine dünne Linie. (b) Die Gewichtsmatrix (1) des Graphen: Je dunkler die Zelle in Zeile $i$ und Spalte $i'$, desto größer das Gewicht $A_{i,i'}$, und eine weiße Zelle steht für ein Tagespaar, das keine Kante bildet. Daten erzeugt durch `pythondemos/weightedgraph.py`

Ein Ähnlichkeitsgraph macht aus dem Clustering eine Frage nach den Gewichten. Ein Cluster ist eine Teilmenge $\mathcal{C} \subseteq \mathcal{V}$ von Knoten, deren innere Kanten große Gewichte und deren ausgehende Kanten kleine Gewichte tragen. Das Gesamtgewicht der Kanten, die die Teilmenge verlassen,

$$\sum_{i \in \mathcal{C}} \sum_{i' \in \mathcal{V} \setminus \mathcal{C}} A_{i,i'}, \tag{4}$$

ist der Schnitt von $\mathcal{C}$. Methoden des Graph-Clusterings wie das Spektrales Clustering wählen die Cluster so, dass ihre Schnitte (4) im Verhältnis zu ihrer Größe klein werden (von Luxburg, 2007, Abschn. 5; Shalev-Shwartz and Ben-David, 2014, Abschn. 22.3.1; Shi and Malik, 2000). Für die zehn Tage in Krems seien die sieben Tage, deren Messwert um 15:00 Uhr unter 18 Grad liegt, als Teilmenge $\mathcal{C}$ gewählt. Ihr Schnitt (4) beträgt 1.64 bei einem gesamten Kantengewicht von 13.62: Die vier Kanten, die sie verlassen, machen 21 Prozent der Kanten aus und tragen 12.1 Prozent des Gewichts. Über alle 365 Tage des Jahres 2024 und mit $k = 10$ hat der Graph 2214 Kanten. Die Kanten zwischen den 211 kalten und den 154 warmen Tagen tragen 1.9 Prozent des Gesamtgewichts, und keine Kante verbindet einen Tag im Januar oder Februar mit einem Tag im Juli oder August. Auf einem in Ausschnitte zerlegten Foto liefert dieselbe Konstruktion

mit der mittleren Farbe eines Ausschnitts als dessen Merkmalsvektor einen gewichteten Graphen, dessen Cluster die Regionen einer Bildsegmentierung sind. Ein gewichteter Graph geht auch über einen Strafterm in die empirische Risikominimierung (ERM) ein. Beim Transfer-Learning und beim föderiertes Lernen (FL) sind die Knoten Lernaufgaben, jede mit ihrem eigenen lokaler Datensatz und einer lokalen Hypothese $h^{(i)}$, und das Gewicht $A_{i,i'}$ gibt an, wie ähnlich sich zwei Aufgaben sind. Die Verallgemeinerte Totalvariationsminimierung (GTVMin) minimiert die Summe der lokalen Trainingsfehler zuzüglich der verallgemeinerte totale Variation (GTV)

$$\sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} d^{(i,i')}, \tag{5}$$

wobei $d^{(i,i')}$ ein Maß für die Diskrepanz zwischen $h^{(i)}$ und $h^{(i')}$ ist. Die GTV (5) koppelt die Aufgaben: Eine Kante mit großem Gewicht zieht die beiden Hypothesen an ihren Knoten zusammen, und ein Gewicht von null lässt sie unabhängig. Das Netzwerk-Lasso ist der Spezialfall mit der euklidischer Abstand zwischen den lokalen Modellparameter als Diskrepanz (Hallac et al., 2015). Das Hinzufügen eines Strafterms ist einer der drei Wege, die die Regularisierung unterscheidet, und der gewichtete Graph ist der Ort, an dem dieser Strafterm festgelegt wird. Auf einem Ähnlichkeitsgraph aus gelabelten und ungelabelten Datenpunkte ist der Strafterm (5) mit der quadrierten Differenz $\big(h(\mathbf{x}^{(r)}) - h(\mathbf{x}^{(r')})\big)^2$ als Diskrepanz der Laplace-Regularisierer des semi-überwachtes Lernen (SSL) (Belkin et al., 2006, Abschn. 2.2). Er verlangt von einer Hypothese, auf zwei durch eine Kante mit großem Gewicht verbundenen Datenpunkte ähnliche Werte anzunehmen, wodurch die ungelabelten Datenpunkte die Hypothese formen. Ein Graph Neural Network (GNN) nutzt die Gewichte für einen dritten Zweck, nämlich um die Merkmalsvektoren der Nachbarn jedes Knotens zu einem neuen Merkmalsvektor des Knotens zu kombinieren.

Siehe auch: Graph, ungerichteter Graph, Kantengewicht, Adjazenzmatrix, Laplace-Matrix, Knotengrad, Nachbar, Ähnlichkeitsgraph, k-Nächste-Nachbarn-Graph, Graph-Clustering, Spektrales Clustering, Clustering, Transfer-Learning, GTV, GTVMin, SSL, GNN, Matrix.

## Querverweise

## Literatur


1. Belkin et al. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J. Mach. Lear. Research, 7, pp. 2399–2434.

2. Chung (1997). Spectral Graph Theory. American Mathematical Society, Providence, RI, USA.

3. von Luxburg (2007). A tutorial on spectral clustering. Statist. Comput., 17(4), pp. 395–416.

4. Hallac et al. (2015). Network Lasso: Clustering and Optimization in Large Graphs. In: Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 387–396.

5. Newman (2010). Networks: An Introduction. Oxford Univ. Press, New York, NY, USA.

6. Shalev-Shwartz and Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge Univ. Press, New York, NY, USA.

7. Shi and Malik (2000). Normalized Cuts and Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8), pp. 888–905.

8. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.