Kantengewicht

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 Kantengewicht ist eine nichtnegative Zahl $A_{i,i'}$, das der Kante $\{i, i'\}$ eines weighted graphs zugewiesen wird und misst, wie stark die beiden von ihr verbundenen Knoten miteinander verknüpft sind. Gesammelt in der Gewichts-Matrix $\mathbf{A}$, bilden die Kantengewichte eines undirected graphs eine symmetrische Matrix mit einem Nulleintrag für jedes Paar, das keine Kante ist. Im machine learning (ML) sind die Knoten oft die Datenpunkte eines Datensatzs, und das Gewicht eines Paares ist die Gaußsche Ähnlichkeit ihrer Merkmalsvektoren, eine abnehmende Funktion ihrer euklidischer Abstand mit einer Breite σ als Hyperparameter; ein k-nearest-neighbor graph (k-NN graph) behält nur die Gewichte benachbarter Paare bei. Kantengewichte können auch eine mit den Daten gegebene Beziehung kodieren, wie Zitationen zwischen Artikeln oder die Zusammenarbeit zwischen den devices eines federated learning network (FL network)s. ML-Methoden verwenden die Gewichte durch gewichtete Summen über die Kanten. Die quadratische Form der Laplacian matrix ist die gewichtete Summe der quadrierten Differenzen eines Knotensignals, die für den Indikator eines Clusters das Gesamtgewicht der geschnittenen Kanten ist. Der Strafterm des convex clusterings berechnet die Abstand zwischen zwei Clusterzentren proportional zum Gewicht ihrer Kante. Kantengewichte werden vor dem Training angegeben und unterscheiden sich von den Gewichte eines künstliches neuronales Netz (KNN)s, die Modellparameter sind.

Definition

Man betrachte acht Tage des Jahres 2024 in Krems an der Donau (Österreich), jeweils den 46. Tag ab dem 1. Januar, jeder ein Datenpunkt mit einem Merkmalsvektor $\mathbf{x} \in \mathbb{R}^2$, der die Lufttemperatur um 06:00 und um 15:00 (UTC) des Tages enthält. Zwei Tage mit ähnlichen Temperaturen sind ähnliche Datenpunkte. Eine Möglichkeit, die Ähnlichkeit jedes Tagepaares zu erfassen, ist ein Graph. Die Tage sind seine Knoten $i = 1, \ldots, n$, hier mit $n = 8$, und eine Kante $\{i, i'\}$ zwischen zwei Tagen trägt eine Zahl, die angibt, wie ähnlich sie sind: das Kantengewicht $A_{i,i'} \geq 0$. In einem solchen Graph über acht Tage verbindet eine dicke Kante den 19. Mai und den 4. Juli, deren Temperaturen sich um ein halbes Grad Celsius unterscheiden, während die Kante zwischen dem 19. August und dem 19. November zu schwach ist, um sie zu sehen.

Ein weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ besteht aus einer Knotenmenge $\mathcal{V} = \{1, \ldots, n\}$, einer Kantenmenge $\mathcal{E}$ und einem Kantengewicht $A_{i,i'}$ für jede Kante $\{i, i'\} \in \mathcal{E}$. Die Gewichte eines undirected graphs ohne Selbstschleifen werden in der GewichtsMatrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ gesammelt. Ihre Einträge erfüllen

$$A_{i,i'} = A_{i',i} \geq 0, \quad A_{i,i} = 0, \quad \text{und} \quad A_{i,i'} = 0 \text{ for } \{i, i'\} \notin \mathcal{E}. \tag{1}$$

Die erste Bedingung in der GewichtsMatrix (1) gilt, weil die Kante $\{i, i'\}$ eines undirected graphs die Kante $\{i', i\}$ ist, daher ist die GewichtsMatrix eine symmetrische Matrix. Die dritte Bedingung macht ein Nullgewicht und eine fehlende Kante ununterscheidbar, sodass die GewichtsMatrix den Graph bestimmt und seine adjacency matrix ist (Chung, 1997, Sect. 1.4; von Luxburg, 2007, Sect. 2). Ein ungewichteter Graph ist der Spezialfall, in dem jede Kante das Gewicht 1 hat. Der gewichtete node degree von Knoten i ist die Zeilensumme $\sum_{i'=1}^{n} A_{i,i'}$ der GewichtsMatrix (Hastie et al., 2009, Sect. 14.5.3).

Im machine learning (ML) stammen Kantengewichte aus zwei Quellen. Die erste Quelle sind die Merkmalsvektoren der Datenpunkte: die m Datenpunkte eines Datensatzs werden zu den Knoten eines similarity graphs, mit $n = m$, und das Gewicht eines Paares ist eine abnehmende Funktion der Abstand zwischen ihren Merkmalsvektoren. Die übliche Wahl ist die Gaußsche Ähnlichkeit

$$A_{i,i'} := \exp\Big(- \Big\| \mathbf{x}^{(i)} - \mathbf{x}^{(i')} \Big\|_2^2 / (2\sigma^2) \Big) \quad \text{für } i \neq i', \tag{2}$$

mit einer Breite $\sigma > 0$ (Hastie et al., 2009, Sect. 14.5.3; von Luxburg, 2007, Sect. 2). Das Gaußsche Gewicht (2) ist 1 für identische Merkmalsvektoren, $\exp(-1/2) \approx 0.61$ bei euklidischer Abstand σ, und etwa 0.01 bei euklidischer Abstand 3σ. Für die acht Tage und $\sigma = 5$ Grad beträgt das Gewicht 0.99 für den 19. Mai und den 4. Juli, deren Merkmalsvektoren 0.5 Grad auseinanderliegen, und $4 \cdot 10^{-5}$ für den 19. August und den 19. November, 22.6 Grad auseinander. Sieben der 28 Paare wiegen mindestens 0.5, und neun wiegen unter 0.01 (Abb. 1(b)). Die Breite σ ist ein Hyperparameter. Mit $\sigma = 1$ wiegen nur zwei Paare mindestens 0.01, sodass der Graph in isolierte Tage zerfällt. Mit $\sigma = 50$ wiegt jedes Paar mindestens 0.90, sodass die Gewichte ähnliche von unähnlichen Tagen nicht mehr trennen. Eine Breite σ in der Größenordnung der mittleren Abstand eines Datenpunkts zu seinem k-ten nächsten Nachbar ist eine gängige Wahl (von Luxburg, 2007, Sect. 8).

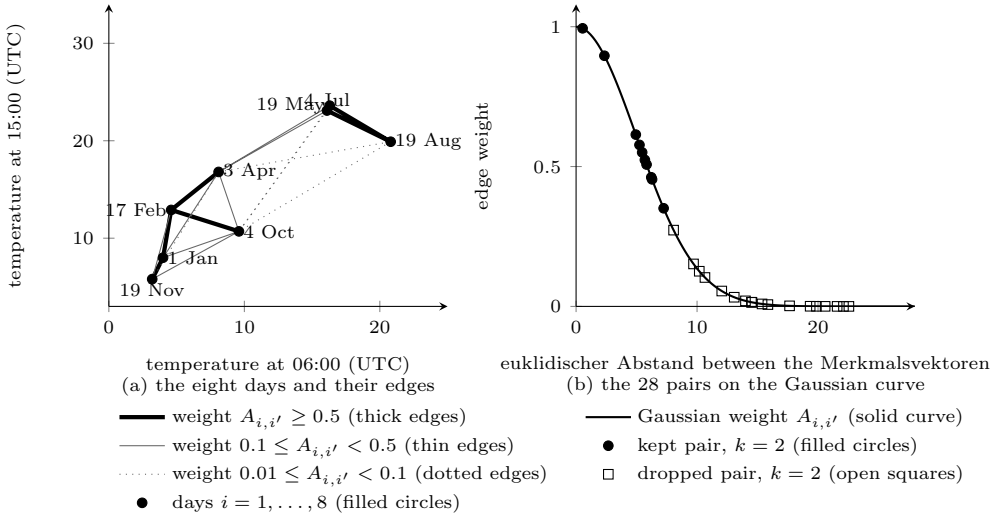

Abbildung 1: (a) Acht der 365 Tage des Jahres 2024 in Krems, jeder 46. Tag ab dem 1. Januar (gefüllte Kreise), mit der Lufttemperatur um 06:00 und um 15:00 Uhr (UTC) eines Tages als Merkmalsvektor. Jedes Tagepaar ist eine Kante mit dem Gaußschen Gewicht (2) für $\sigma = 5$ Grad: dicke Kanten wiegen mindestens 0.5, dünne Kanten zwischen 0.1 und 0.5, gepunktete Kanten zwischen 0.01 und 0.1, und die neun Paare unter 0.01 sind nicht gezeichnet. (b) Die 28 Paare auf der Gaußschen Kurve (durchgezogene Kurve) des Gewichts gegen den euklidischer Abstand zwischen den Merkmalsvektoren: gefüllte Kreise sind die zehn Paare, die beibehalten werden, wenn einer der Tage zu den zwei nächsten Nachbarn des anderen gehört, offene Quadrate sind die Paare, die diese Regel verwirft. Daten generiert von `pythondemos/edgeweight.py`

Das Gaußsche Gewicht (2) ist für jedes Paar positiv, sodass jedes Paar eine Kante ist und die GewichtsMatrix keine Nullen außerhalb ihrer Diagonale hat. Ein dünnbesetzter Graph wird aus dem k-nearest-neighbor graph (k-NN graph) erhalten. Das Paar $\{i, i'\}$ ist eine Kante, wenn $\mathbf{x}^{(i')}$ zu den k nächsten Nachbarn von $\mathbf{x}^{(i)}$ gehört oder $\mathbf{x}^{(i)}$ zu denen von $\mathbf{x}^{(i')}$. Eine Kante behält das Gaußsche Gewicht (2), und jedes andere Gewicht wird auf Null gesetzt (Hastie et al., 2009, Sect. 14.5.3; von Luxburg, 2007, Sect. 2). Für die acht Tage und $k = 2$ sind zehn der 28 Paare Kanten, und der Graph ist nicht connected: er zerfällt in die fünf kühlen Tage von Oktober bis April und den 19. Mai, 4. Juli und 19. August. Abb. 1(b) kennzeichnet die beibehaltenen Paare durch gefüllte Kreise und die verworfenen Paare durch offene Quadrate. An diesen acht Tagen sind die beibehaltenen Paare zufällig genau die Paare oberhalb eines Gewichtsschwellenwerts, da jedes beibehaltene Paar mindestens 0.35 und jedes verworfene Paar höchstens 0.27 wiegt.

Nichts garantiert diese Übereinstimmung, da die Regel Ränge und nicht Gewichte liest. Eine Erhöhung von k auf drei bricht dies bereits, und verbindet die

beiden Teile des Graphs zu einem: das verworfene Paar 3. April und 19. November, mit einem Gewicht von 0.055, übertrifft dann das beibehaltene Paar 3. April und 19. August, mit 0.033. Der k-NN graph verbindet daher Datenpunkte in dünn besiedelten und in dichten Regionen des Merkmalsraum gleichermaßen, weshalb er die empfohlene erste Wahl ist (von Luxburg, 2007, Sect. 8).

Die zweite Quelle ist eine mit den Daten gegebene Beziehung und nicht aus Merkmalsvektoren berechnet: Zitationen zwischen wissenschaftlichen Artikeln, Freundschaften zwischen den Nutzern eines sozialen Netzwerks, Links zwischen Webseiten oder die Anzahl der Messungen, die zwei Wetterstationen teilen. In einem Bild sind die Pixel die Knoten, und das Gewicht zweier Pixel nimmt mit der Differenz ihrer Helligkeit und mit ihrer Abstand im Bild ab, eine Konstruktion, die in der image segmentation durch normalisierte Schnitte verwendet wird (Shi and Malik, 2000). In einem federated learning network (FL network) sind die Knoten devices, und das Kantengewicht misst das Ausmaß ihrer Zusammenarbeit, z.B. die Kapazität der Kommunikationsverbindung zwischen ihnen oder die statistische Ähnlichkeit ihrer local datasets. Im Transfer-Learning sind die Knoten Lernaufgaben, und das Gewicht koppelt die Aufgaben benachbarter Wetterstationen stärker als die von entfernten.

Die Kanten eines künstliches neuronales Netz (KNN)s tragen ebenfalls Gewichte, und diese sind ein anderes Objekt: ein KNN-Gewicht ist ein Modellparameter, das durch Training gelernt wird, und es kann negativ sein. Die Kantengewichte eines weighted graphs sind nichtnegativ und werden vor Beginn des Trainings angegeben; sie gehören zu den Daten, nicht zum Modell.

Kantengewichte fließen in ML-Methoden durch gewichtete Summen über die Kanten ein. Ein Knotensignal $\mathbf{u} = (u_1, \dots, u_n)^\top \in \mathbb{R}^n$ weist jedem Knoten eine Zahl zu, wie z.B. ein Label oder eine Vorhersage. Die Laplacian matrix $\mathbf{L}^{(\mathcal{G})}$ des Graphs erfüllt

$$\mathbf{u}^\top \mathbf{L}^{(\mathcal{G})} \mathbf{u} = \sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} \big(u_i - u_{i'}\big)^2, \tag{3}$$

die Laplacesche quadratische Form (von Luxburg, 2007, Prop. 1). Sie ist eine gewichtete Summe von quadrierten Differenzen.
Eine schwere Kante gewichtet eine Differenz zwischen ihren beiden Knoten stärker als eine leichte Kante, und die Summe ist null für ein Signal, das auf jeder connected Komponente des Graph konstant ist. Für den Indikator einer Teilmenge $\mathcal{C} \subseteq \mathcal{V}$ der Knoten, $u_i = 1$ für $i \in \mathcal{C}$ und $u_i = 0$ sonst, ist die quadratische Form (3) das Gesamtgewicht der Kanten zwischen $\mathcal{C}$ und dem Rest des Graph, das Gewicht des Schnitts. Für die acht Tage schneiden die sechs Tage, deren 15:00-Messwert unter 20 Grad liegt, Kanten mit einem Gesamtgewicht von 1.31, was 19 Prozent des Gesamtgewichts von 6.74 aller Paare ausmacht. Die Aufteilung der Tage in die ersten vier und die letzten vier des Jahres schneidet 4.45, oder 66 Prozent. Graph clustering-Methoden suchen Cluster, die wenig Gewicht schneiden; spectral clustering minimiert annähernd das Schnittgewicht geteilt durch die Größe des Cluster mittels Eigenvektoren von $\mathbf{L}^{(\mathcal{G})}$ (von Luxburg, 2007, Sect. 5).

Semi-supervised learning (SSL) auf einem Graph sucht ein Knotensignal, das

mit den Labels der wenigen gelabelten Knoten übereinstimmt und die quadratische Form (3) klein macht (Zhu et al., 2003). Die Gewichte übertragen dann die Labels von gelabelten auf ungelabelte Knoten. Sie entscheiden, auf welche Datenpunkte außerhalb des Trainingsdatensatz eine gelernte Hypothese erweitert wird, was die Frage der Generalisierung ist.

Die generalized total variation (GTV) ersetzt die quadrierte Differenz in der quadratischen Form (3) durch ein measure der discrepancy zwischen den lokale Modelle der beiden Knoten. Generalized total variation minimization (GTV-Min) addiert es zu den lokalen Trainingsfehler, wie bei federated learning (FL) und Transfer-Learning. Das network Lasso ist der Fall des unquadrierten Abstand zwischen den Modellparameter der beiden lokale Modelle (Hallac et al., 2015).

Convex clustering verwendet die Kantengewichte in seiner Zielfunktion. Jeder Datenpunkt $\mathbf{x}^{(i)}$ erhält seinen eigenen Clusterzentrum $\mathbf{w}^{(i)}$, und die Clusterzentren minimieren

$$\sum_{i=1}^{n} \left\| \mathbf{x}^{(i)} - \mathbf{w}^{(i)} \right\|_2^2 + \alpha \sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} \left\| \mathbf{w}^{(i)} - \mathbf{w}^{(i')} \right\|_2, \tag{4}$$

mit einem Regularisierungsparameter $\alpha \geq 0$ (Hocking et al., 2011; Sun et al., 2021, Eq. (1)). Die erste Summe in der Zielfunktion (4) ist der clustering error der Clusterzentren, der Trainingsfehler, den empirische Risikominimierung (ERM) minimiert; die zweite Summe ist ein Strafterm, sodass die Methode durch Verlustpenalisierung regularisiert, eine der drei Routen, die Regularisierung unterscheidet.

Die Gewichte entscheiden, welche Paare von Clusterzentren zusammengezogen werden. Ein Paar mit Gewicht null wird nie belastet, und ein schweres Paar wird proportional zu seinem Gewicht belastet. Für ausreichend großes α fallen die Clusterzentren eines schweren Paares zusammen, was die beiden Datenpunkte in ein Cluster einordnet. Für ein hartes Clustering, bei dem jeder Datenpunkt eines Cluster einen Clusterzentrum teilt, ist der Strafterm der Abstand zwischen den beiden Clusterzentren multipliziert mit dem Schnittgewicht. Für die oben genannten kalten und warmen Tage beträgt es 17.7, die Abstand 13.5 zwischen ihren Clusterzentren multipliziert mit dem Schnittgewicht 1.31.

Was Kantengewichte leisten, ist ein Ort, um Wissen zu platzieren, das die Merkmalsvektoren nicht tragen: eine Zitation, eine Freundschaft, die Kapazität einer Verbindung. Was sie kosten, ist eine Wahl, die vor dem Training getroffen und nie von diesem korrigiert wird. Die Breite σ zeigt den Preis deutlich: bei $\sigma = 1$ wiegen nur zwei der 28 Paare mindestens 0.01 und der Graph besteht aus acht isolierten Tagen, bei $\sigma = 50$ wiegt jedes Paar mindestens 0.90 und die Gewichte trennen nichts, und beide Extreme lassen jede der oben genannten Methoden ohne Ansatzpunkte zurück. Ein k-NN graph entfernt die Breite aus der Kantenmenge, behält sie aber in den Gewichten, und liest Ränge statt Abstände, sodass es ein leichtes Paar behalten und ein schwereres fallen lassen kann. Wo die Gewichte eine Schätzung sind, ist k-fache Kreuzvalidierung über σ und k gegen den Validierungsfehler der Methode, die den Graph konsumiert,

die übliche Antwort, und wo eine Relation mit den Daten gegeben ist, gibt es nichts zu wählen.

Siehe auch: weighted graph, Graph, undirected graph, adjacency matrix, Laplacian matrix, node degree, similarity graph, k-NN graph, Nachbar, graph clustering, spectral clustering, convex clustering, Clustering, clustering error, GTV, GTVMin, network Lasso, FL network, Transfer-Learning, SSL, Matrix, KNN.

Querverweise

Literatur

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

2. Hocking et al. (2011). Clusterpath: An Algorithm for Clustering using Convex Fusion Penalties. In: Proc. of the 28th International Conference on Machine Learning,.

3. Sun et al. (2021). Convex Clustering: Model, Theoretical Guarantee and Efficient Algorithm. J. Mach. Learn. Res., 22(9), pp. 1–32.

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

5. 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.

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

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

8. Zhu et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919.