Transfer-Learning

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

Transfer-Learning ist die Wiederverwendung dessen, was bei einer Lernaufgabe gelernt wurde, zur Lösung einer anderen Lernaufgabe. Eine Lernaufgabe lässt sich durch eine Wahrscheinlichkeitsverteilung von Datenpunkte charakterisieren, und die Wirksamkeit des Transfer-Learnings hängt von der Ähnlichkeit der entsprechenden Wahrscheinlichkeitsverteilungen ab. Die Ähnlichkeit von Aufgaben kann als Ähnlichkeitsgraph codiert werden, dessen Kantengewichte die Lernaufgaben an seinen Knoten koppeln; Verfahren des Transfer-Learnings lassen sich dann durch Optimierung über diesen Graph gewinnen.

Definition

Die Wetterstationen Krems an der Donau und Melk in der Wachau zeichnen jeweils stündlich die Lufttemperatur auf. Die Vorhersage des Messwerts y um 15:00 Uhr (UTC) eines Tages an einer Station, des Labels, aus ihrem Messwert x um 06:00 Uhr (UTC), dem Merkmal, ist eine Lernaufgabe. Jede Wetterstation definiert eine eigene Lernaufgabe. Abb. 1 zeigt die beiden Aufgaben nebeneinander, jeweils mit einem Entscheidungsbaum der Tiefe zwei, der Hypothese, die aus einem Trainingsdatensatz von fünf gelabelten Tagen des Jahres 2020 gelernt wurde. Eine Lernaufgabe lässt sich durch die Wahrscheinlichkeitsverteilung ihrer Datenpunkte charakterisieren: An den beiden Stationen sind diese Verteilungen ähnlich. Transfer-Learning nutzt eine solche Ähnlichkeit. Es koppelt die Lernaufgaben, sodass das an einer Station Gelernte die Vorhersagen an der anderen verbessert, und seine Wirksamkeit hängt davon ab, wie ähnlich die Wahrscheinlichkeitsverteilungen der gekoppelten Aufgaben sind (Pan and Yang, 2010).

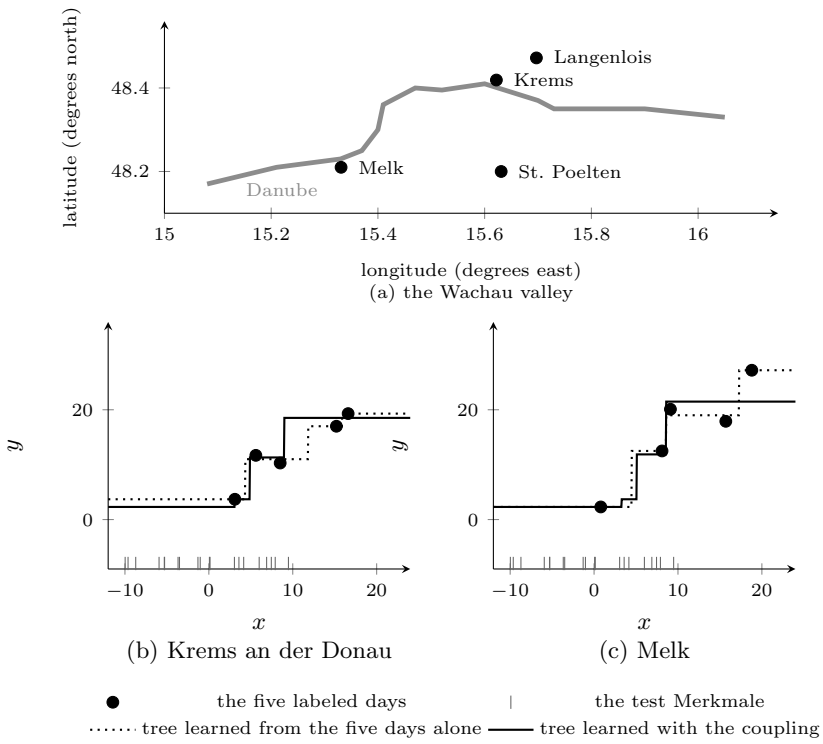


Abbildung 1: Zwei Lernaufgaben in der Wachau. (a) Der Verlauf der Donau und vier Wetterstationen rund um das Tal. (b) und (c) Die Tage des Jahres 2020 in Krems an der Donau und in Melk, mit dem Messwert eines Tages um 06:00 Uhr (UTC) als Merkmal x und seinem Messwert um 15:00 Uhr (UTC) als Label y. An jeder Station bilden fünf gelabelte Tage (ausgefüllte Kreise) das Trainingsdatensatz. Die gepunktete Treppenfunktion ist der allein aus diesen fünf Tagen gelernte Entscheidungsbaum; die durchgezogene ist der Entscheidungsbaum, der gelernt wird, wenn die beiden Aufgaben gekoppelt werden, indem gefordert wird, dass ihre Vorhersagen auf den Test-Merkmale $\mathcal{T}$ übereinstimmen (20 der 40, jeder zweite, als Markierungen entlang der horizontalen Achse). Daten erzeugt mit `pythondemos/transferlearning.py`

Die Kopplung in Abb. 1 wird umgesetzt, indem der Zielfunktion des Modell-Trainings an jeder Station ein Strafterm hinzugefügt wird. Seien $h^{(1)}$ und $h^{(2)}$ die Hypothesen der beiden Stationen und $\mathcal{T}$ ein Testdatensatz von Merkmal-Werten, die zufällig aus dem Bereich $[-10, 10]$ der Messwerte um 06:00 Uhr ausgewählt wurden. Transfer-Learning minimiert dann die Summe der beiden Trainingsfehler zuzüglich $\alpha \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \left(h^{(1)}(x) - h^{(2)}(x)\right)^2$, der mittleren quadratischen Abweichung der beiden Hypothesen auf den Test-Merkmale. Für Entscheidungsbäume ist die Minimierung über einen Baum bei festgehaltenem anderem Baum ein gewichtetes Training dieses Baums auf seinen eigenen Tagen

zusammen mit den vom anderen Baum gelabelten Test-Merkmale. Das Abwechseln der beiden Minimierungen ist ein Blockkoordinatenabstieg für das gekoppelte Problem: Eine exakte Minimierung über einen Baum kann die gekoppelte Zielfunktion nicht vergrößern, sodass die Folge der Zielfunktionswerte monoton fallend ist. Ein Entscheidungsbaum wird gierig gelernt, sodass jeder Schritt nur eine näherungsweise Blockminimierung ist und das Abwechseln eine Heuristik für das gekoppelte Problem darstellt statt eines Verfahrens, das es exakt löst. Mit fünf gelabelten Tagen pro Station stimmen die gekoppelten Bäume auf den Test-Merkmale überein (mittlere quadratische Abweichung $5.1 \to 0.5$), doch ihr mittlerer Fehler auf den zurückgehaltenen Tagen der beiden Stationen sinkt nicht: Er steigt von 28.6 auf 29.8. Bei einem einzigen Nachbarn kann der Übereinstimmungsstrafterm einen Baum von seinen eigenen Tagen wegziehen; erst der nachfolgende Ähnlichkeitsgraph macht den Transfer lohnend.

Die Kopplung zweier Aufgaben lässt sich auf eine Sammlung von Lernaufgaben erweitern, eine pro Knoten $i = 1, \ldots, n$ eines ungerichteten gewichteter Graph $\mathcal{G}$, des Ähnlichkeitsgraphen. Eine gewichtete Kante $\{i, i'\}$ mit Kantengewicht $A_{i,i'}$ codiert die Ähnlichkeit zwischen zwei Lernaufgaben; ein Gewicht von null lässt sie ungekoppelt. Jeder Knoten i trägt ein lokales Modell, einen lokaler Datensatz und die daraus gelernte lokale Hypothese $h^{(i)}$. Die Minimierung der Summe der lokalen Trainingsfehler zuzüglich des α-Fachen der verallgemeinerte totale Variation (GTV)

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

wobei $d^{(i,i')}$ ein Maß für die Diskrepanz zwischen den lokalen Hypothesen $h^{(i)}$ und $h^{(i')}$ ist, wird als verallgemeinerte Totalvariationsminimierung (GTVMin) bezeichnet (SarcheshmehPour et al., 2023). Die Kopplung in Abb. 1 ist eine GTVMin auf einem Graph mit zwei Knoten, mit der mittleren quadratischen Abweichung auf den Test-Merkmale $\mathcal{T}$ als Maß der Diskrepanz; dieses Maß ist auf jedes lokales Modell anwendbar, einschließlich der Entscheidungsbäume. Für lokale Modelle mit Modellparameter $\mathbf{w}^{(i)}$ ist der quadrierte Abstand $\left\|\mathbf{w}^{(i)} - \mathbf{w}^{(i')}\right\|_2^2$ eine gängige Wahl, und das Netzwerk-Lasso ist die Variante mit dem nicht quadrierten Abstand (Hallac et al., 2015). Die GTVMin lernt alle lokalen Hypothesen gleichzeitig und überträgt Wissen zwischen gut verbundenen Lernaufgaben, wie beim föderiertes Lernen (FL). Abb. 2 zeigt einen Ähnlichkeitsgraph von neun Stationen in Niederösterreich, mit Kantengewichte, die mit dem Abstand zwischen den Stationen abfallen und jeweils für die nächsten Nachbarn beibehalten werden. Mit fünf gelabelten Tagen und einem Entscheidungsbaum an jeder Station senkt die GTVMin mit der Abweichung auf $\mathcal{T}$ den mittleren Fehler auf den zurückgehaltenen Daten der neun Stationen von 41.5 auf 34.6 sowie den Fehler an sieben von ihnen; in Krems und in Melk, den beiden Stationen aus Abb. 1, erhöht die Kopplung ihn, sodass ein Transfer, der sich im Mittel auszahlt, sich nicht an jedem Knoten auszahlen muss.

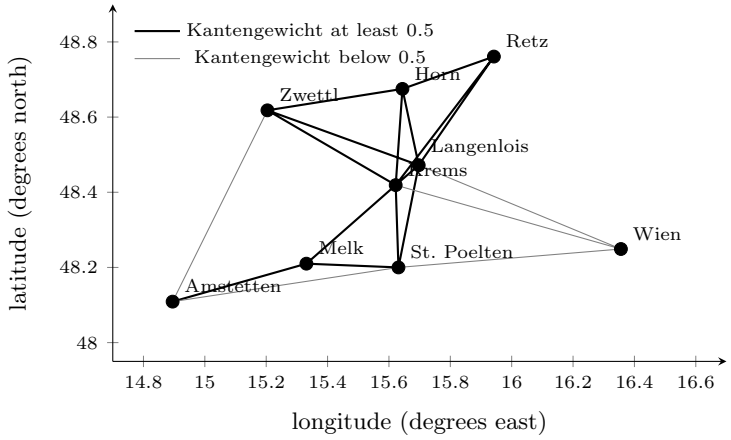


Abbildung 2: Ein Ähnlichkeitsgraph von neun Wetterstationen in Niederösterreich, von denen jede ein Knoten mit eigener Lernaufgabe ist. Das Kantengewicht zwischen zwei Stationen beträgt $\exp(-d^2/(2s^2))$ für ihren Abstand d und $s = 40$ Kilometer, beibehalten für die drei nächsten Nachbarn jeder Station: Dicke Kanten haben ein Gewicht von mindestens 0.5, dünne Kanten weniger. Daten erzeugt mit `pythondemos/transferlearning.py`

Ein Spezialfall des Transfer-Learnings ist ein Knoten ohne jegliche gelabelte Datenpunkte. Sein eigener Trainingsfehler fehlt in der Zielfunktion, sodass die GTVMin seine Hypothese allein über die GTV aus den Vorhersagen seiner Nachbarn auf den Test-Merkmale bestimmt. Der ungelabelte Knoten entspricht dem Fall des semi-überwachtes Lernen (SSL): Er verfügt weiterhin über Datenpunkte, und seine Position im Ähnlichkeitsgraph ermöglicht deren Nutzung (Chapelle et al., 2006).

Eine zweite Form des Transfer-Learnings verortet die Ähnlichkeit zwischen einer Vortrainingsaufgabe und der neuen Aufgabe statt zwischen den Stationen eines Ähnlichkeitsgraphen. Sie passt ein auf einem großen Datensatz trainiertes tiefes Netz an einen kleinen an: Die von seinen frühen Schichten berechneten Merkmale werden beibehalten (siehe Repräsentationslernen), und die übrigen Modellparameter werden durch Feinabstimmung angepasst. Wenn die für die erste Aufgabe nützlichen Merkmale für die zweite nicht nützlich sind, kann die angepasste Hypothese ein größeres Risiko aufweisen als eine von Grund auf neu trainierte, was als negativer Transfer bezeichnet wird (Pan and Yang, 2010). Der Fall, in dem die Lernaufgabe fest bleibt und sich nur die Wahrscheinlichkeitsverteilung der Merkmalsvektoren ändert, ist die Domänenadaption (Pan and Yang, 2010, Abschn. 2.3).

Siehe auch: Lernaufgabe, GTVMin, GTV, lokales Modell, lokaler Datensatz, Kantengewicht, SSL, Multitask-Lernen, Feinabstimmung, Domänenadaption, FL.

Querverweise

Literatur

1. Chapelle et al. (2006). Semi-Supervised Learning. MIT Press, Cambridge, MA, USA.

2. SarcheshmehPour et al. (2023). Clustered Federated Learning via Generalized Total Variation Minimization. IEEE Trans. Signal Process., 71, pp. 4240–4256.

3. 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.

4. Pan and Yang (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), pp. 1345–1359.