Merkmalstransformation

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

Eine Merkmalstransformation ist eine Funktion, die den Merkmalsvektor eines Datenpunkt auf einen neuen Merkmalsvektor abbildet, sodass eine machine learning (ML)-Methode auf den neuen Merkmale anstelle der ursprünglichen läuft. Was gelernt wird, ist dann die Komposition der Transformation mit einer Hypothese auf dem neuen Merkmalsraum, weshalb ein lineares Modell der neuen Merkmale kein lineares Modell der ursprünglichen ist. Methoden unterscheiden sich darin, woher die Transformation stammt: Hauptkomponentenanalyse (PCA) erhält sie allein aus den ursprünglichen Merkmale, linear discriminant analysis (LDA) verwendet auch die Labels, eine random projection zieht sie zufällig, und ein Autoencoder lernt sie mit einem künstliches neuronales Netz (KNN). Jede versteckte Schicht eines tiefes Netz ist eine weitere Merkmalstransformation, und eine Kernel-Methode legt eine durch die Wahl eines Kernel fest, ohne sie jemals zu evaluieren. Die Anzahl der neuen Merkmale bestimmt die Größe des Hypothesenraum der Methode, die auf ihnen läuft, daher ist es model pruning, diese klein zu halten, eine der drei Wege, die die Regularisierung unterscheidet. Für die 15:00-Uhr-(UTC)-Messung in Krems an der Donau (Österreich) verwandeln die ersten fünf Potenzen des Jahrestages eine nahezu flache Hypothese in eine Kurve, die den Jahreszeiten folgt, und reduzieren den Validierungsfehler von 90.3 auf 24.4 Quadratgrad. Die Wahl ist ebenso wichtig wie die Anzahl: Die drei ungeraden Potenzen des Tages lassen den Validierungsfehler über dem der konstanten Hypothese, weshalb die Transformation oft aus dem Datensatz gelernt und nicht von Hand gewählt wird.

Definition

Eine Wetterstation in Krems an der Donau (Österreich) zeichnet die Lufttemperatur um 15:00 Uhr (UTC) an jedem Tag der Jahre 2023 und 2024 auf. Jeder Tag ist ein Datenpunkt, sein Label ist dieser Messwert, und sein einziges ursprüngliches Merkmal x ist die Position des Tages im Jahr, skaliert auf das Intervall $[-1, 1]$. Die 364 Tage des Jahres 2023, die den Messwert tragen, bilden das Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ und die 365 solchen Tage des Jahres 2024 das Validierungsdatensatz $\mathcal{D}^{(\text{val})}$. Eine lineare Regression auf x lernt eine nahezu flache Hypothese aus $\mathcal{D}^{(\text{train})}$ mittels least squares: Sie steigt um 6.0 Grad vom ersten Tag des Jahres bis zum letzten, gegenüber einem Label-Bereich von 38.6

Grad. Ihr Trainingsfehler beträgt 81.7 Quadratgrad und ihr Validierungsfehler 90.3, über den 87.4 der konstanten Hypothese, die immer den Stichprobenmittel der Labels in $\mathcal{D}^{(\text{train})}$ vorhersagt. Das Ersetzen von x durch die fünf Zahlen $x, x^2, \ldots, x^5$ und das Anwenden derselben Methode darauf ergibt eine Kurve, die den Jahreszeiten folgt, mit einem Validierungsfehler von 24.4. Abb. 1(a) zeigt die beiden gelernten Hypothesen über die Tage von $\mathcal{D}^{(\text{train})}$.

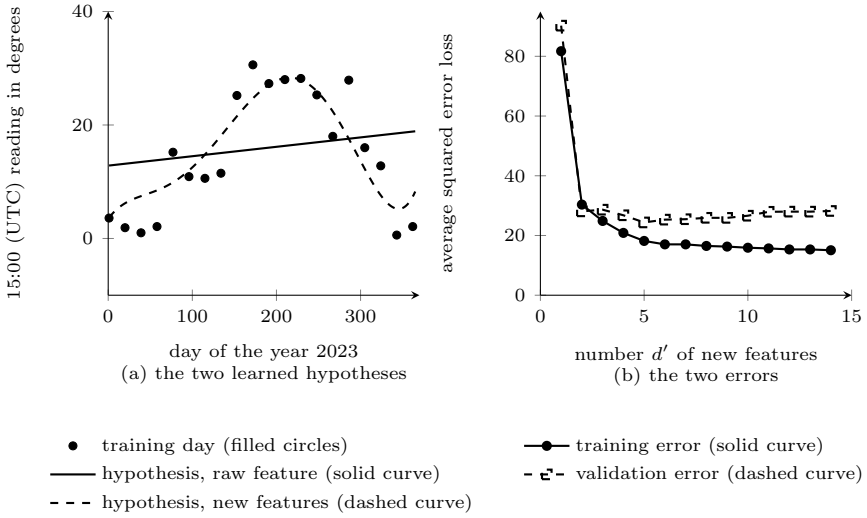

Abbildung 1: Eine Merkmalstransformation des einzelnen ursprünglichen Merkmal eines Tages in Krems, seiner Position im Jahr. (a) Zwanzig der 364 Tage von $\mathcal{D}^{(\text{train})}$, jeder 19 in kalendarischer Reihenfolge (gefüllte Kreise), die Hypothese, die allein aus dem ursprünglichen Merkmal gelernt wurde (durchgezogene Kurve), und die Hypothese, die aus den $d' = 5$ neuen Merkmale gelernt wurde (gestrichelte Kurve), die ein Polynom des ursprünglichen Merkmal ist. (b) Der Trainingsfehler (durchgezogene Kurve, gefüllte Kreise) und der Validierungsfehler (gestrichelte Kurve, offene Quadrate) gegen die Anzahl d' der neuen Merkmale. Daten generiert von `pythondemos/featuretransformation.py`

Eine Merkmalstransformation ist eine Funktion

$$\mathbf{\Phi} : \mathcal{X} \to \mathcal{X}', \qquad \mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x}), \tag{1}$$

die den Merkmalsvektor $\mathbf{x}$ eines Datenpunkt von seinem Merkmalsraum $\mathcal{X}$ zu einem neuen Merkmalsvektor in einem neuen Merkmalsraum $\mathcal{X}'$ überführt. Für ein einzelnes ursprüngliches Merkmal sind die Potenzen des einführenden Beispiels die Transformation

$$\mathbf{\Phi}(x) := \left(x, x^2, \ldots, x^{d'}\right)^\top, \tag{2}$$

die d' neue Merkmale liefert. Eine machine learning (ML)-Methode läuft dann auf den neuen Merkmale anstelle der ursprünglichen und liefert eine Hypothese

h' auf $\mathcal{X}'$. Was insgesamt gelernt wird, ist dann die Komposition

$$h = h' \circ \mathbf{\Phi} : \mathcal{X} \to \mathcal{Y}, \qquad \mathbf{x} \mapsto h'\big(\mathbf{\Phi}(\mathbf{x})\big). \tag{3}$$

Die Komposition (3) ist der Grund, warum ein lineares Modell der neuen Merkmale kein lineares Modell der ursprünglichen ist. Mit den Potenzen (2) ist die gelernte h ein Polynom vom Grad d' in x, was polynomial regression ist (Bishop, 2006, Sect. 3.1).

Methoden unterscheiden sich darin, woher $\mathbf{\Phi}$ stammt. Hauptkomponentenanalyse (PCA) erhält sie allein aus den ursprünglichen Merkmale, als die lineare Abbildung eines gegebenen Rangs, deren Rekonstruktionsfehler am kleinsten ist (Hastie et al., 2009, Sect. 14.5.1). Linear discriminant analysis (LDA) verwendet auch die Labels und projiziert auf den Unterraum, der von den Klassenzentroiden aufgespannt wird (Hastie et al., 2009, Sect. 4.3.3). Eine random projection zieht $\mathbf{\Phi}$ zufällig, anstatt sie zu lernen (Johnson and Lindenstrauss, 1984), und ein Autoencoder lernt eine nichtlineare $\mathbf{\Phi}$ mit einem künstliches neuronales Netz (KNN) (Hinton and Salakhutdinov, 2006). Jede versteckte Schicht eines tiefes Netz ist eine weitere Merkmalstransformation, die gemeinsam mit der Ausgabe Schicht gelernt wird (Mallat, 2016). Eine Kernel-Methode legt $\mathbf{\Phi}$ durch die Wahl eines Kernel fest, der das Skalarprodukt zweier transformierter Merkmalsvektoren zurückgibt (Schölkopf and Smola, 2002, Sect. 1.1). Eine solche Methode verwendet die transformierten Merkmalsvektoren nur über diese Skalarprodukte, sodass $\mathbf{\Phi}$ niemals evaluiert wird (Schölkopf and Smola, 2002, Sect. 1.5). Eine Transformation, die weniger neue Merkmale liefert, als der Datenpunkt ursprüngliche hat, ist Dimensionsreduktion, und das Erlangen von $\mathbf{\Phi}$ aus einem Datensatz anstatt von Hand ist Gegenstand von Repräsentationslernen.

Die Anzahl d' der neuen Merkmale in der Definition (1) bestimmt die Größe des Hypothesenraum der Methode, die auf ihnen läuft. Für die Potenzen (2) ist jede Hypothese, die bei d' verfügbar ist, auch bei $d' + 1$ verfügbar, sodass der Trainingsfehler, der durchschnittliche squared error loss über $\mathcal{D}^{(\text{train})}$, nicht mit d' steigen kann: Er fällt von 81.7 auf 15.1 Quadratgrad, wenn d' von 1 auf 14 wächst. Der Validierungsfehler auf $\mathcal{D}^{(\text{val})}$ verhält sich anders. Er ist am kleinsten bei $d' = 5$ mit 24.4 Quadratgrad und steigt wieder auf 28.2 bei $d' = 14$, das Überanpassung, das ein großer Hypothesenraum mit sich bringt (Bishop, 2006, Sect. 1.1). Abb. 1(b) zeigt die beiden Fehler. Das Kleinhalten von d' verkleinert daher den Hypothesenraum, was model pruning ist, einer der drei Wege, die die Regularisierung unterscheidet.

Zwei andere Namen sind für die Funktion (1) in Gebrauch. Die Literatur zur Kernel-Methode nennt sie Feature Map (Schölkopf and Smola, 2002, Sect. 2.1), und die Literatur zum statistischen Lernen eine Basiserweiterung (Hastie et al., 2009, Sect. 5.1). Die beiden Bedeutungen von *Feature Map* sind unterschiedlich: Eine Merkmalstransformation ist die Abbildung zwischen Merkmalsräume der Definition (1), während eine Merkmalsabbildung das Gitter von Aktivierungen ist, das von einem Kanal einer konvolutionalen Schicht geliefert wird.

Da der Trainingsfehler nicht mit d' steigen kann, kann er die Kosten einer Transformation, die zu viele neue Merkmale liefert, nicht anzeigen, daher ist

d' ein Hyperparameter und wird auf zurückgehaltenen Datenpunkte durch validation gewählt. Die Anzahl ist nicht die ganze Wahl, da eine Transformation, die festgelegt wird, bevor die Labels betrachtet werden, das verwerfen kann, wovon das Label abhängt. Die drei ungeraden Potenzen x, x^3, x^5 lassen den Validierungsfehler bei 87.9 Quadratgrad, über den 87.4 der konstanten Hypothese, während die beiden geraden Potenzen x^2, x^4 allein 23.9 ergeben. Jede ungerade Potenz ist antisymmetrisch um $x = 0$, die Mitte des Jahres, und die 15:00-Uhr-Messung in Krems ist nahezu symmetrisch um diesen Punkt: 77 Prozent der durchschnittlichen quadratischen Abweichung der Labels von ihrem Stichprobenmittel liegen im symmetrischen Teil. Die Komposition (3) ändert auch die Bedeutung eines Gewicht: Ein Gewicht eines lineares Modell auf den neuen Merkmale multipliziert eines davon, in Krems einen Kubus einer reskalierten Kalenderposition, und nicht eine von der Station aufgezeichnete Größe.

Im Gegenzug erreicht least squares auf den neuen Merkmale Hypothesen, die kein lineares Modell des ursprünglichen Merkmal enthält, in Krems eine Kurve, die den Jahreszeiten folgt, wo eine gerade Linie nutzlos ist. Zwei Alternativen vermeiden die Wahl von $\mathbf{\Phi}$ von Hand. Eine behält die ursprünglichen Merkmale bei und wählt eine Methode, deren Hypothesenraum kein lineares Modell von ihnen ist, wie zum Beispiel einen Entscheidungsbaum. Die andere erhält $\mathbf{\Phi}$ aus dem Datensatz, was Repräsentationslernen ist, mit einem Autoencoder oder den versteckten Schichten eines tiefes Netz.

Synonyme: Feature Map, Basiserweiterung.

Siehe auch: Repräsentationslernen, Merkmal, Merkmalsraum, PCA, LDA, random projection, Autoencoder, Dimensionsreduktion, Kernel-Methode, Merkmalsabbildung, polynomial regression, Regularisierung, Hyperparameter, validation, Entscheidungsbaum.

Querverweise

Literatur

1. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

2. Hinton and Salakhutdinov (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), pp. 504–507.

3. Schölkopf and Smola (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA.

4. Mallat (2016). Understanding deep convolutional networks. Philos. Trans. Roy. Soc. A, 374(2065), pp. Art. no. 20150203.

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

6. Johnson and Lindenstrauss (1984). Extensions of Lipschitz mappings into a Hilbert space. Contemporary Math., 26, pp. 189–206.