k-fache Kreuzvalidierung

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

Die k-fache Kreuzvalidierung (k-fold CV) teilt ein Datensatz in k Falten gleicher Größe und verwendet jede Falte einmal als Validierungsdatensatz, während die anderen $k-1$ Falten das Trainingsdatensatz einer gegebenen machine learning (ML)-Methode bilden. Die k resultierenden Validierungsfehler werden zu einer einzigen Zahl gemittelt, einem Indikator für die Generalisierung über das Trainingsdatensatz hinaus. Während ein einzelner validation-Schritt das Risiko einer einzelnen Hypothese schätzt, schätzt die k-fache Kreuzvalidierung das Risiko einer ML-Methode. Leave-one-out cross-validation (LOO-CV) ist der Extremfall der k-fachen Kreuzvalidierung mit einem Datenpunkt pro Falte.

Definition

Abb. 1 zeigt vier Datenpunkte, die tägliche Wetterbedingungen in Krems an der Donau (Österreich) repräsentieren. Jeder Datenpunkt ist durch die Temperatur um 06:00 (UTC) eines Tages als Merkmal x und die Temperatur um 15:00 (UTC) desselben Tages als Label y charakterisiert. Drei der vier Datenpunkte werden als Trainingsdatensatz für polynomial regression mit Grad drei verwendet. Der verbleibende Datenpunkt dient als Validierungsdatensatz. Polynomial regression liefert eine Hypothese, die genau durch die drei Datenpunkte im Trainingsdatensatz verläuft und somit einen Trainingsfehler von null erreicht.

Wie in Abb. 1 dargestellt, generalisiert die gelernte Hypothese $\widehat{h}$ jedoch nicht gut über das Trainingsdatensatz hinaus, d. h. polynomial regression mit Grad drei ist Überanpassung. Der Validierungsfehler auf dem zurückgehaltenen Datenpunkt kann dieses Überanpassung aufdecken, aber eine einzelne Aufteilung tut dies unzuverlässig. Tatsächlich ergibt das Zurückhalten des ersten Datenpunkt aus Abb. 1 einen Validierungsfehler von 1.37, was auf eine zuverlässige Hypothese hindeutet. Das Zurückhalten des zweiten, dritten oder vierten Datenpunkt hingegen ergibt Validierungsfehler von 8.05, 7.02 und 72.42, wie die Teilbilder (b) bis (d) von Abb. 1 zeigen. Ihr Mittelwert 22.22 ist die Schätzung der 4-fachen Kreuzvalidierung. Die Idee ist, dass verschiedene Aufteilungen eines Datensatz in Trainingsdatensatz und Validierungsdatensatz eine verlässlichere Bewertung einer machine learning (ML)-Methode ermöglichen.

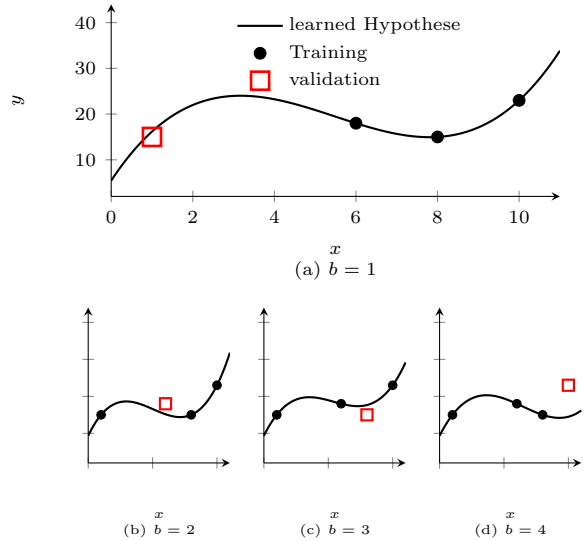


Abbildung 1: (a) Vier Datenpunkte, die jeweils einen Tag in Krems an der Donau (Österreich) mit der Temperatur um 06:00 als Merkmal x und der Temperatur um 15:00 als Label y repräsentieren. Die ausgefüllten Kreise markieren eine mögliche Wahl eines Trainingsdatensatz aus drei Datenpunkte. Das rote offene Quadrat markiert den resultierenden zurückgehaltenen Datenpunkt, der als Validierungsdatensatz dient. Die Kurve ist die von polynomial regression mit Grad drei gelernte Hypothese. Der Faltenindex b nummeriert den zurückgehaltenen Datenpunkt. (b) bis (d) Dasselbe für $b = 2, 3, 4$: Das Zurückhalten eines anderen Datenpunkt verändert die gelernte Hypothese und ihren Validierungsfehler. Daten erzeugt mit `pythondemos/kfoldcv.py`

Betrachte eine ML-Methode $\mathcal{A}$, die ein Trainingsdatensatz auf eine gelernte Hypothese $\widehat{h} = \mathcal{A}(\mathcal{D}^{(\text{train})})$ abbildet. Die k-fache Kreuzvalidierung schätzt den erwarteten Verlust (oder das Risiko) der Methode $\mathcal{A}$ wie folgt (Hastie et al., 2009, Abschnitt 7.10; Stone, 1974). Zunächst teilt sie ein Datensatz $\mathcal{D}$ gleichmäßig in k Teilmengen, die Falten $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(k)}$ (Abb. 2). Für jede Falte $b = 1, \ldots, k$ trainiert eine Iteration das Modell auf der Vereinigung aller Falten außer $\mathcal{D}^{(b)}$ und bewertet die resultierende Hypothese auf der zurückgehaltenen Falte $\mathcal{D}^{(b)}$. Die k faltenweisen Validierungsfehler werden dann gemittelt, um den Kreuzvalidierungsfehler E_{CV}, die Schätzung des erwarteten Verlust, zu erhalten. Algorithmus 1 fasst dieses Verfahren zusammen.

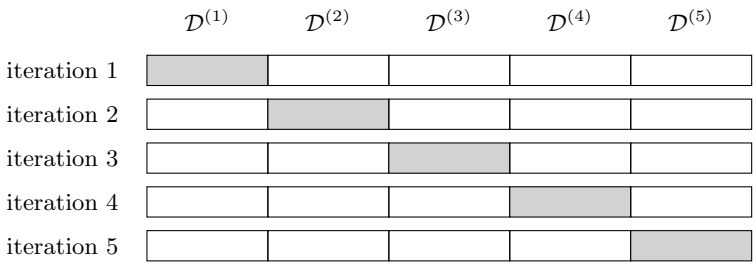


Abbildung 2: Ein Beispiel für k-fache Kreuzvalidierung mit $k = 5$: Das verfügbare Datensatz $\mathcal{D}$ wird gleichmäßig in fünf Falten $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(5)}$ aufgeteilt. Jede Zeile entspricht einer Iteration der k-fachen Kreuzvalidierung: Die schattierte Zelle markiert die Falte, die in dieser Iteration als Validierungsdatensatz verwendet wird, während die verbleibenden $k-1$ Falten das Trainingsdatensatz bilden

Algorithm 1 k-fache Kreuzvalidierung (Hastie et al., 2009, Abschnitt 7.10.1)

Input: ML method $\mathcal{A}$, Datensatz $\mathcal{D}$, number of folds k

1: divide $\mathcal{D}$ evenly into the folds $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(k)}$
2: **for** $b = 1, \ldots, k$ **do**
3: $\quad \mathcal{D}^{(\text{train})} := \mathcal{D} \setminus \mathcal{D}^{(b)}, \quad \mathcal{D}^{(\text{val})} := \mathcal{D}^{(b)}$
4: $\quad \widehat{h}^{(b)} := \mathcal{A}(\mathcal{D}^{(\text{train})})$
5: $\quad E_v^{(b)} := \widehat{L}\big(\widehat{h}^{(b)} | \mathcal{D}^{(\text{val})}\big)$
6: **end for**

Output: CV error $E_{\text{CV}} = (1/k) \sum_{b=1}^{k} E_v^{(b)}$

Die Anzahl der Falten k ist ein Hyperparameter von Algorithmus 1, der die rechnerischen und statistischen Eigenschaften entscheidend beeinflusst. Die k Iterationen führen eine Training-Methode k-mal aus. Die Wahl eines großen Wertes für k führt daher zu einer höheren rechnerischen Komplexität. Die Verwendung eines einzelnen Validierungsdatensatz bewertet eine einzelne Hypothese, nämlich diejenige, die $\mathcal{A}$ aus einem Trainingsdatensatz liefert (Bates et al., 2024, Abschnitt 3.4). Im Gegensatz dazu bewertet die k-fache Kreuzvalidierung keine einzelne Hypothese. Stattdessen lernt jede Iteration eine potenziell andere Hypothese; die Teilbilder (a) bis (d) von Abb. 1 zeigen vier solcher Hypothesen. Somit misst der durchschnittliche Validierungsfehler E_{CV} die Methode $\mathcal{A}$ selbst und nicht nur eine einzelne Hypothese $\mathcal{A}(\mathcal{D}^{(\text{train})})$, die sie für ein bestimmtes Trainingsdatensatz liefert. In Abb. 1 ist $\mathcal{A}$ polynomial regression mit Grad drei und $E_{\text{CV}} = 22.22$. Insbesondere misst E_{CV} das durchschnittliche Risiko der Hypothesen, die aus Trainingsdatensätze gewonnen werden, die aus einem Anteil von $(k-1)/k$ des Datensatz bestehen (Bates et al., 2024, Abschnitt 3; Hastie et al., 2009, Abschnitt 7.10.1).

Wichtige Varianten und Spezialfälle der k-fachen Kreuzvalidierung sind:

- *Stratified k-fold cross-validation (stratified k-fold CV)*: Sie erhält die Klassenverhältnisse des gesamten Datensatz in jeder Falte. Die Stratifizierung ist bei der Klassifikation mit imbalanced data wichtig: Eine einfache zufällige Aufteilung kann dazu führen, dass eine Falte fast keine Datenpunkte der Minderheitsklasse enthält. Der Validierungsfehler dieser Falte wird dann auf einem Validierungsdatensatz berechnet, dessen Klassenverhältnisse sich von denen des gesamten Datensatz unterscheiden, sodass er eine unzuverlässige Schätzung des Risiko ist (Kohavi, 1995). Wenn jede Falte getrennt aus jeder Klasse gezogen wird, wird dies ausgeschlossen: Keine Falte kann ohne Minderheitsklasse bleiben, sodass jeder Validierungsfehler auf einem Validierungsdatensatz berechnet wird, das das Datensatz repräsentiert (Kohavi, 1995, Abschnitt 2.2).

- *Leave-one-out cross-validation (LOO-CV)*: Sie ist der Extremfall $k = m$, wobei m die Anzahl der Datenpunkte im Datensatz ist, sodass jede Falte einen einzelnen Datenpunkt enthält (Stone, 1974). Sie trainiert in jeder Iteration auf nahezu dem gesamten Datensatz, zum Preis von m Training-Läufen und einer hohen Varianz der Schätzung.

- *Wiederholte k-fache Kreuzvalidierung*: Sie mittelt die Ergebnisse mehrerer Durchläufe mit unterschiedlichen zufälligen Faltenzuweisungen (Kim, 2009). Dies reduziert die durch die Aufteilung verursachte Varianz und ist nützlich bei kleinen Datensätze, bei denen eine einzelne Aufteilung die Schätzung stark beeinflusst.

- *Blockierte und gruppierte Falten*: Die Falten in Algorithmus 1 setzen die Annahme unabhängig und identisch verteilter Daten (u.i.v.-Annahme) voraus (Bates et al., 2024, Abschnitt 2). Für aufeinanderfolgende Datenpunkte einer Zeitreihe oder für mehrere Datenpunkte, die an einer Wetterstation aufgezeichnet wurden, platziert eine zufällige Aufteilung korrelierte Datenpunkte sowohl im Trainingsdatensatz als auch in der zurückgehaltenen Falte, sodass E_{CV} das Risiko unterschätzt. Falten, die aus zusammenhängenden Blöcken oder aus ganzen Gruppen gebildet werden, halten korrelierte Datenpunkte jeweils auf einer Seite der Aufteilung. Ein Schritt, der das gesamte Datensatz vor der Aufteilung verwendet, unterschätzt das Risiko auf die gleiche Weise. Hastie et al. (2009), Abschnitt 7.10.2 wählen die 100 Merkmale, die am stärksten mit dem Label über alle 50 Datenpunkte korreliert sind, und der E_{CV} der resultierenden Ein-Nächster-Nachbar-Hypothese beträgt im Mittel 3%, während keine Hypothese in diesem Setting ein Risiko unter 50% hat. Ein solcher Schritt gehört in jede Iteration.

Siehe auch: validation, Validierungsfehler, generalization gap, learning curve, LOO-CV, stratified k-fold CV, Modellauswahl, Bootstrap.

Querverweise

Literatur

1. Bates et al. (2024). Cross-Validation: What Does It Estimate and How Well Does It Do It?. Journal of the American Statistical Association, 119(546), pp. 1434–1445.

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

3. Kim (2009). Estimating Classification Error Rate: Repeated Cross-Validation, Repeated Hold-Out and Bootstrap. Computational Statistics & Data Analysis, 53(11), pp. 3735–3745.

4. Kohavi (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1137–1143.

5. Stone (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society. Series B (Methodological), 36(2), pp. 111–133.