

Themenblock Prognosemodelle (Klassifikation und Regression):
02_02Evaluation_Validation_02

Hyperparameteroptimierung

Erarbeitet von
Dr. Katja Theune

Lernziele	1
Inhalt	2
Einstieg	2
Hyperparameter und ihre Optimierung	2
Nested cross-validation	3
Trade-off Verzerrung und Varianz	3
Abschluss	3
Weiterführendes Material	4
Disclaimer	4

Lernziele

- Du kannst erklären, was Hyperparameter sind
- Du kannst die Idee und Vorgehensweise der nested cross-validation erläutern
- Du kannst den Trade-Off zwischen Verzerrung und Varianz erläutern

Inhalt

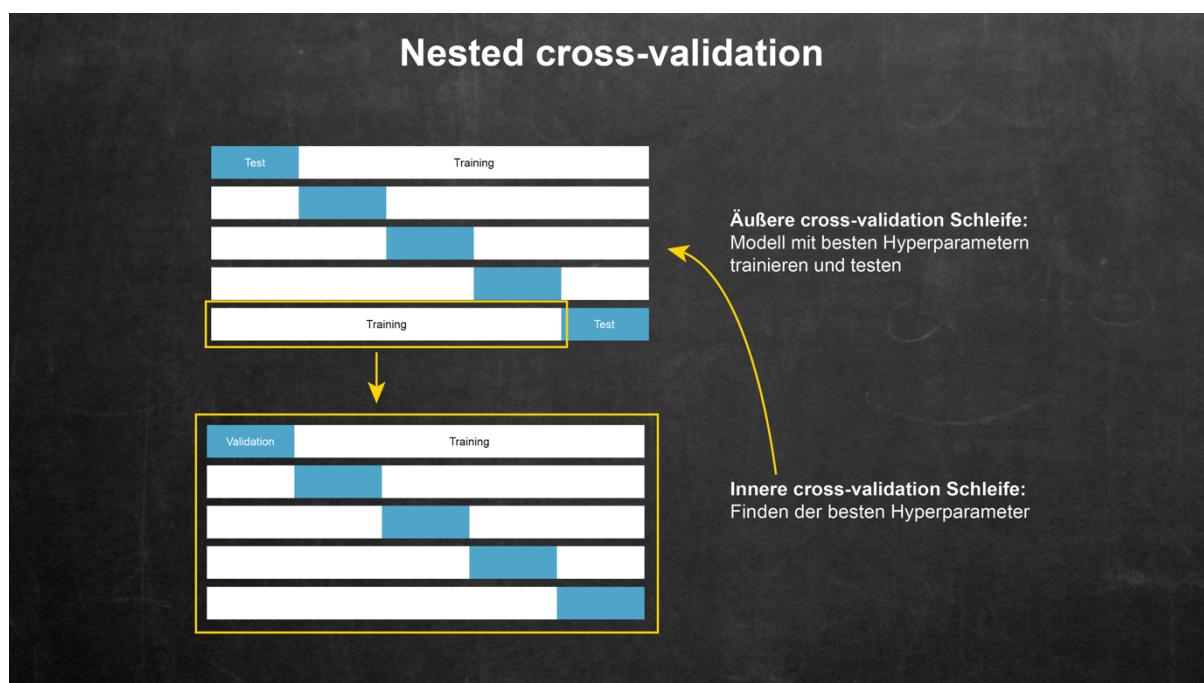
Einstieg

Die sogenannten Hyperparameter der verschiedenen Verfahren des maschinellen Lernens sind wichtige Stellschrauben, die man optimieren kann und die einen Einfluss auf die Über- bzw. Unteranpassung der Prognosemodelle haben. Die verschiedenen Ideen und Konzepte zur Evaluation können wir auch auf die Optimierung dieser Hyperparameter übertragen.

Hyperparameter und ihre Optimierung

Hyperparameter sind Parameter, welche nicht aus den Daten gelernt, sondern von Anwendern und Anwenderinnen festlegt werden, um z. B. die Prognosegenauigkeit zu verbessern. Beispiele für Hyperparameter sind beim k-nearest neighbours Algorithmus die Anzahl an nächsten Nachbarn, bei Bäumen und Wäldern die Baumtiefe oder die Anzahl an Bäumen, oder bei neuronalen Netzen die Anzahl an Layern und Neuronen.

Ein bekanntes Verfahren bei der Hyperparameteroptimierung ist die sogenannte grid search. Hier werden alle Parameterkombinationen eines vorher festgelegten Wertebereichs ausprobiert und evaluiert. Bei der random search werden dagegen nur zufällig ausgewählte Kombinationen verwendet. Es gibt hier natürlich noch weitere, teilweise komplexere Möglichkeiten. Evaluiert werden die Modelle mit den verschiedenen Parameterkombinationen dann üblicherweise mit einer cross-validation.



Allerdings haben wir hier wieder das Problem, dass wenn wir die gleichen Daten für das Optimieren der Parameter und das Evaluieren des Modells nutzen, es zur Überanpassung und einer zu optimistischen Einschätzung des Fehlers kommen kann.

Nested cross-validation

Um auch hier realistische Einschätzungen der Fehler zu bekommen, ist die Idee, unseren vorliegenden Datensatz zusätzlich zu den Trainings- und Testdaten noch in Validierungsdaten aufzuteilen und sozusagen eine cross-validation in der cross-validation durchzuführen. In der inneren Schleife werden die Trainingsdaten nochmal in Trainings- und Validierungsdaten geteilt. Mit Hilfe der Trainingsdaten wird ein Modell mit einer bestimmten Parameterkombinationen trainiert und mit den Validierungsdaten evaluiert. Das wird für mehrere Kombinationen gemacht und die besten Parameter ausgewählt. In der äußeren Schleife wird mit den ursprünglichen Trainingsdaten das Modell mit den besten Parametern trainiert und mit den Testdaten der Testfehler des Modells bestimmt. Man nennt dies dann nested cross-validation. Das hilft insbesondere, um eine generelle und realistische Einschätzung der Leistung unseres Modells zu erhalten.

Trade-off Verzerrung und Varianz

Der Fehler eines Modells kann in Verzerrung, bzw. Bias und Varianz aufgeteilt werden. Eine hohe Verzerrung bedeutet, dass das Modell die Muster und Gesetzmäßigkeiten in den Trainingsdaten nicht gut erkennt und ggf. nicht komplex genug ist. Es macht dann keine guten Vorhersagen. Wir sprechen dann von Unteranpassung. Eine hohe Varianz bedeutet, dass das Modell zu sehr auf die Trainingsdaten angepasst und ggf. zu komplex ist. Wir sprechen dann von Überanpassung. Das Modell reagiert dann zu empfindlich auf Änderungen in den Daten. Daher kann es bei der Anwendung auf einen neuen Datensatz zu ganz anderen Ergebnissen kommen.

Am liebsten wollen wir natürlich ein Prognosemodell, welches eine kleine Verzerrung und eine kleine Varianz hat. Es soll also die Gesetzmäßigkeiten in den Trainingsdaten gut erkennen und auch übertragbar sein auf neue Daten, also generalisierbar sein. Das ist leider normalerweise unmöglich und es müssen Kompromisse zwischen beiden Aspekten gefunden werden.

Dieser Trade-off hängt auch mit der Wahl bestimmter Hyperparameter unserer Verfahren zusammen. Z. B. führt im k-nearest neighbours Algorithmus ein hoher Wert von k tendenziell zu einer niedrigen Varianz und einer hohen Verzerrung. Bei Entscheidungsbäumen beeinflusst die Tiefe des Baumes Verzerrung und Varianz.

Abschluss

Wir haben gesehen, dass wir die Idee der cross-validation auch für die gleichzeitige Hyperparameteroptimierung verwenden können. Allerdings unterliegt der Fehler eines Modells einem Verzerrungs-Varianz Trade-off, der auch von den Hyperparametern eines Modells beeinflusst wird.

Weiterführendes Material

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3. Auflage). Morgan Kaufmann.

James, G., Witten, D., Hastie, T., & Tibshirani, R., & Tylor, J. (2023). *An Introduction to Statistical Learning - with Applications in Python*. Springer.

Lantz, B. (2015). *Machine learning with R* (2. Auflage). Packt Publishing Ltd, Birmingham.

Disclaimer

Transkript zu dem Video „Prognosemodelle (Klassifikation und Regression):

Hyperparameteroptimierung“, Dr. Katja Theune.

Dieses Transkript wurde im Rahmen des Projekts ai4all des Heine Center for Artificial Intelligence and Data Science (HeiCAD) an der Heinrich-Heine-Universität Düsseldorf unter der Creative Commons Lizenz [CC-BY 4.0](#) veröffentlicht. Ausgenommen von der Lizenz sind die verwendeten Logos, alle in den Quellen ausgewiesenen Fremdmaterialien sowie alle als Quellen gekennzeichneten Elemente.