Bachelorarbeit

Vorhersagebereiche über conformal prediction und Erweiterung für Transferlernen unter covariate shift

Philipps-Universität Marburg

Fachbereich 12 - Mathematik und Informatik

Vorgelegt von: Till Kaiser

Datum: 09.09.2025

Betreuer: Prof. Hajo Holzmann

Inhaltsverzeichnis

Einleitung

In vielen Anwendungsbereichen – von der Medizin über Finanzprognosen bis hin zu maschinellem Lernen – stehen wir vor der Aufgabe, zuverlässige Vorhersagen zu treffen *(Balasubramanian u. a. 2014)*. Häufig liefern gängige Verfahren lediglich einen Punktschätzer für den erwarteten Wert. Allerdings ist es häufig wichtig den wahren Wert mit einer bestimmten Wahrscheinlichkeit zu bestimmen. Dies ist mittels eines einzelnen Punktes in den meisten Fällen nicht möglich. Daher ist es für uns von Interesse einen ganzen Bereich vorherzusagen, welcher den wirklich realisierten Wert mit hoher Wahrscheinlichkeit enthält.
Conformal prediction liefert hierfür einen Algorithmus, der es unter einfachen Voraussetzungen ermöglicht, einen sicheren Vorhersagebereich zu konstruieren. Das besondere daran ist, dass wir nur sehr geringe Anforderungen an die Verteilung der Daten stellen müssen. Denn es genügt wenn die Bedingung der Austauschbarkeit erfüllt ist. Dabei benötigen wir keine weiteren Informationen zu der Verteilung *(Vovk u. a. (2022), Fontana u. a. (2023))*.

Das Prinzip von Vohersagebereichen oder Konfidenzintervallen ist eines der grundlegenden statistischen Konzepte und weit verbreitet. Dennoch ist der Ansatz von conformal prediction noch sehr jung. Die ersten Arbeiten zu diesem Thema erschienen in den späten 1990er Jahren, wie etwa von *Gammerman u. a. (1998)*. Eines der wichtigsten und umfangreichsten Werke zu conformal prediction ist das Buch *Algorithmic Learning in a Random World* von *Vovk u. a. (2022)*, dessen erste Auflage 2005 erschien. In den folgenden Jahren wurden viele weitere Erkenntnisse in diesem Bereich gewonnen, die eine einfachere Berechnung, oder die Anwendung auf Datensätze mit nicht-austauschbar verteilten Daten ermöglichen.

Das Ziel dieser Arbeit ist es, die Methoden von conformal prediction zu nutzen, um damit sichere Vorhersagebereiche auch unter covariate shift zu konstruieren. Dazu werden wir die Ergebnisse von *Tibshirani u. a. (2019)* darstellen und diese in Verbindung zur klassischen Theorie setzen. Außerdem wollen wir die Anwendung von conformal prediction unter covariate shift betrachten und damit Unterschiede zur klassischen Variante besser erkennen. Wir werden in Kapitel 1 damit beginnen die grundlegende Theorie hinter conformal prediction zu verstehen. Dazu werden wir zuerst einige neue Begriffe und Methoden einführen, mit denen wir einen Vorhersagebereich konstruieren können. Dabei orientieren wir uns bei dieser Einführung insbesondere an *Vovk u. a. (2022)* und *Balasubramanian u. a. (2014)*. Außerdem wollen wir überprüfen, ob es, unter den gegebenen Voraussetzungen, eine bessere Alternative zu conformal prediction gibt. Zum Ende des Kapitels untersuchen wir, wie es uns gelingen kann den Vorhersageberich noch präziser zu gestalten. Im Anschluss werden wir in Kapitel 2 die Anwendung von Regression, auf conformal prediction, betrachten und dafür einen Ansatz am Beispiel der Ridge Regression, nach *Vovk u. a. (2022)* vorstellen. Als letzten Teil dieses Kapitels führen wir den split conformal predictor ein, der vor allem durch die Arbeit von *Lei u. a. (2018)* als bedeutende Anwendung von conformal prediction etabliert wurde. In Kapitel 3 werden wir, basierend auf *Conformal Prediction Under Covariate Shift* von *Tibshirani u. a. (2019)*, einen etwas anderen Zugang zu conformal prediction wählen, um damit eine allgemeinere Version zu konstruieren. Diese gewichtete Variante ist unverzichtbar bei der Anwendung, wenn sich die Trainingsdaten, systematisch von den Testdaten unterscheiden. In diesem Zusammenhang gehen wir explizit auf covariate shift ein. Zum Abschluss werden wir in Kapitel 4 die Anwendung von conformal prediction, insbesondere unter covariate shift, anhand von zwei Simulationen demonstrieren.

1 Einführung

1.1 Was ist conformal prediction?

Nachdem wir die Idee von conformal prediction kurz umrissen und motiviert haben, wollen wir das grundlegende Konzept nun genauer fassen. Die Ausgangssituation ist wie folgt: Wir betrachten eine Folge von Werten bzw. Beobachtungen $z_1, z_2, z_3, \cdots \in \mathcal{Z}$, wobei $\mathcal{Z}$ den Beobachtungsraum bezeichnet. Wir nehmen an, dass diese Beobachtungen *austauschbar verteilt* sind, was wir später noch genauer erläutern werden (siehe auch Abschnitt 6.1).

Angenommen es sind $n-1$ Beobachtungen $z_1, \ldots, z_{n-1}$ (Trainingsdatensatz) bekannt. Für den nächsten Wert z_n wollen wir nun eine Vorhersagemenge mit Fehlerrate (Signifikanzniveau) $\alpha \in (0,1)$. Wir wollen also basierend auf unserem Trainingsdatensatz eine Menge Γ^α, die den wahren Wert z_n mit Wahrscheinlichkeit von mindestens $1-\alpha$ enthält *(Fontana u. a. 2023)*.

Zwei Eigenschaften sind für den Vorhersageberich Γ^α besonders wünschenswert (*vgl. Fontana u. a. 2023, Kapitel 2.1.4*):

1. **Validität:** Die Fehlerwahrscheinlichkeit α darf nicht überschritten werden, damit die Menge Γ^α ein verlässliches Ergebnis liefert.
2. **Effizienz:** Die Menge Γ^α ist möglichst klein, um eine informative Vorhersage zu gewährleisten.

Dabei ist die erste Eigenschaft die wichtigste. Es ist also unser Ziel eine Abbildung Γ zu finden, die für jedes Signifikanzniveau α einen validen Vorhersagebereich liefert. Diese sollte sich dabei auch monoton im Signifikanzniveau verhalten. Also für $0 < \alpha_1 \leq \alpha_2 < 1$ soll gelten (*vgl. Vovk u. a. 2022, S. 21*)

$$\Gamma^{\alpha_2}(z_1, \ldots, z_{n-1}) \subseteq \Gamma^{\alpha_1}(z_1, \ldots, z_{n-1}). \tag{1.1}$$

Diese Voraussetzung entspricht der Intuition, dass größere Mengen den wahren Wert mit höherer Wahrscheinlichkeit enthalten. Daraus ergibt sich unsere erste Definition.

Definition 1.1. (vgl. Vovk u. a. (2022), S. 21) Als *confidence predictor* bezeichnen wir eine Familie $(\Gamma^\alpha : \alpha \in (0,1))$ von messbaren Abbildungen

$$\Gamma : \mathcal{Z}^{n-1} \times (0,1) \to 2^{\mathcal{Z}} \tag{1.2}$$

welche (1.1) erfüllt für alle Signifikanzniveaus $0 < \alpha_1 \leq \alpha_2 < 1$ und für alle $n \in \mathbb{N}$.
($2^{\mathcal{Z}}$ bezeichne die Menge aller Teilmengen von $\mathcal{Z}$).

Bemerkung 1.2. (*vgl. Vovk u. a. 2022, S. 21*) Mit der Messbarkeit von Γ, in Definition 1.1, ist gemeint, dass die Menge

$$\{(\alpha, z_1, \ldots, z_n) : z_n \in \Gamma^\alpha(z_1, \ldots, z_{n-1})\}$$

eine messbare Teilmenge von $(0,1) \times \mathcal{Z}^n$ ist.

Wir werden erkennen, dass jeder conformal predictor auch ein confidence predictor ist. Außerdem werden wir zeigen, dass wir mit conformal prediction, unter unseren Bedingungen, den besten confidence predictor erhalten.

Als weiteres Werkzeug benötigen wir ein sogenanntes *nonconformity measure*. Dies soll messen wie ungewöhnlich eine neue Beobachtung im Vergleich zu den bisherigen erscheint. Das ist aus dem Grund wichtig, da wir im folgenden überprüfen müssen wie unterschiedlich die gegebenen Werte sind, um eine Aussage darüber zu treffen, welche Werte noch auftreten können.

Definition 1.3. (*vgl. Fontana u. a. 2023, S. 5*) Eine messbare Abbildung

$$A : \mathcal{Z}^n \times \mathcal{Z} \to \overline{\mathbb{R}},$$

welche den Unterschied von einem Wert $z \in \mathcal{Z}$ zu den Werten im Vektor $B \in \mathcal{Z}^n$ beschreibt, und die Eigenschaft

$$A(B, z) = A(B_\pi, z)$$

erfüllt, nennen wir *nonconformity measure* (B_π steht für jede mögliche Permutation von B). Die Abbildung A ist also invariant unter Permutationen im ersten Argument.
Wir schreiben dann $R = A(B, z)$ und bezeichnen R als *nonconformity score*.

Wenn wir nun den Vektor $B = (z_1, \ldots, z_n)^\top$ betrachten, dann schreiben wir für den nonconformity score von z_i für ein $i \in \{1, \ldots, n\}$,

$$R_i = A(B, z_i). \tag{1.3}$$

Die Schreibweise der nonconformity scores ist in der Literatur nicht einheitlich. In einigen Werken (wie etwa von Vovk u. a. 2022) wird statt R, häufig α verwendet und anstelle von α, für das Signifikanzniveau, ϵ.

Beispiel 1.4. Für eine Menge $\{z_1, \ldots, z_n\} \in \mathcal{Z}^n$ und den Wert $z_i \in \mathcal{Z}$ mit $\mathcal{Z} \subseteq \mathbb{R}$ ist der absolute Unterschied von z_i und dem Durchschnitt der Werte aus dem Vektor $(z_1, ..., z_n)^\top$ ein nonconformity measure. Für die nonconformity scores gilt dann

$$R_i = A((z_1, \ldots, z_n)^\top, z_i) = \left| \frac{\sum_{j=1}^n z_j}{n} - z_i \right|, \quad i = 1, \ldots, n.$$

Bemerkung 1.5. Im Kontext von conformal prediction wird in der gängigen Literatur häufig eine etwas abgewandelte Form, des nonconformity scores aus Definition 1.3, verwendet. Bei diesem betrachten wir eine Abbildung

$$A^{del} : \mathcal{Z}^{n-1} \times \mathcal{Z} \to \overline{\mathbb{R}},$$

wobei wir $(z_1, \ldots, z_n)^\top \in \mathcal{Z}^n$ gegeben haben. Wenn wir nun den Unterschied von z_i, für ein beliebiges $i = 1, \ldots, n$, zu den anderen Werten $z_1, \ldots, z_n$ messen wollen, löschen wir das Element z_i aus dieser Menge bzw. diesem Vektor und betrachten dann

$$R_i = A^{del}(B_{del}, z_i).$$

Der Vektor $B_{del} = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)^\top \in \mathcal{Z}^{n-1}$ enthält nun nicht mehr das zu vergleichende Element z_i *(vgl. Vovk u. a. 2022, Kapitel 2.9.3).*
In dieser Arbeit werden wir nonconformity scores nach Definition 1.3 nutzen, allerdings macht es für die theoretischen Aussagen keinen Unterschied welche Variante verwendet wird. Daher wird die genaue Wahl des nonconformity measures in der Regel auch erst bei der Anwendung relevant. Dort kann es unter Umständen sinnvoll sein diese gelöschte (engl. deleted)

Version zu verwenden. *Shafer und Vovk (2008)* stellen zudem heraus, dass diese nonconformity scores den Vorteil bieten, dass sie das sogenannte *online Modell* generalisieren. Dieses wird auch ausführlich durch *Vovk u. a. (2022)* behandelt, aber hier nicht weitergehend betrachtet.

Im nächsten Schritt wollen wir nun das entschiedene Kriterium für einen Conformal Predictor definieren. Dafür können wir die gerade eingeführten nonconformity scores nutzen, um den p-Wert für z_n zu erhalten. Da wir über den nonconformity score von z_n nicht bestimmen können wie ungewöhnlich z_n ist, müssen wir diesen mit den nonconformity scores aller anderen z_i für $i = 1, \dots, n$ vergleichen.

Definition 1.6. (*vgl. Fontana u. a. 2023, Kapitel 2.1.3*) Für $z_n \in \mathcal{Z}$ bezeichnen wir

$$p_{z_n} = \frac{|\{i = 1, \dots, n : R_i \geq R_n\}|}{n} \in \left[\frac{1}{n}, 1\right], \tag{1.4}$$

als den *p-Wert* von z_n, wobei

$$R_i = A((z_1, \dots, z_n)^\top, z_i)$$

für $i = 1, \dots, n$ der nonconformity score von z_i ist.

Der p-Wert gibt den Anteil der Beobachtungen an, die genauso oder weniger konform sind wie z_n.
Umgangssprachlich können wir sagen (*vgl. Vovk u. a. 2022, S. 30*)

$$p_{z_n} = \frac{\text{Anzahl der } z_i, \text{ die eine größere Abweichung aufweisen als } z_n}{n}.$$

Ist p_{z_n} groß, also nahe bei 1, bedeutet das, dass es viele z_i gibt, die ungewöhnlicher als z_n erscheinen. In diesem Fall ist z_n sehr konform. Wenn p_{z_n} allerdings klein ist (nahe bei $\frac{1}{n}$), dann ist z_n sehr ungewöhnlich im Vergleich zu den anderen z_i.

Der Konfidenzbereich für z_n, zum Konfidenzniveau $1-\alpha$ sei dann die Menge $\Gamma^\alpha(z_1, \dots, z_{n-1})$ welche alle $z_n \in \mathcal{Z}$ enthält mit $p_{z_n} > \alpha$ (dies wird in Satz 1.9 gezeigt). Somit erhalten wir

$$\Gamma^\alpha(z_1, \dots, z_{n-1}) = \{z_n \in \mathcal{Z} : p_{z_n} > \alpha\}. \tag{1.5}$$

Bemerkung 1.7. Wenn wir nun diese Menge berechnen wollen, müssen wir jedes mögliche z_n aus der Menge $\mathcal{Z}$ betrachten und dafür den p-Wert berechnen. Um nun den p-Wert zu erhalten müssen wir für jedes Mal alle n nonconformity scores bestimmen. Wir müssen also *card* $\mathcal{Z}$ mal die n nonconformity scores berechnen. Dies kann unter Umständen einen sehr großen Rechenaufwand bedeuten. Und wenn *card* $\mathcal{Z} = \infty$ ist es gar nicht möglich die Menge (1.5) direkt zu berechnen. Diese Problematik werden wir in Kapitel 2 noch genauer betrachten.

1.2 Validität des Vorhersagebereiches

In diesem Abschnitt wollen wir untersuchen wie sicher $\Gamma^\alpha(z_1, \dots, z_{n-1})$ den wahren Wert für z_n enthält. Diese Eigenschaft bezeichnen wir im Folgenden auch als Validität. Die Wahrscheinlichkeit für den Fehler, dass der wahre Wert für z_n nicht in $\Gamma^\alpha(z_1, \dots, z_{n-1})$ enthalten

ist, soll unser Signifikanzniveau α nicht überschreiten, um einen Konfidenzbereich zu erhalten der verlässliche Werte liefert. Wir wollen also erreichen, dass

$$\mathbb{P}(z_n \in \Gamma^\alpha(z_1, ..., z_{n-1})) \geq 1 - \alpha \tag{1.6}$$

gilt.

Definition 1.8. (*vgl. Fontana u. a. 2023, Kapitel 2.1.4*) Ein conformal predictor Γ^α ist *konservativ valide*, wenn für jede austauschbare Wahrscheinlichkeitsverteilung $\mathbb{P}$ auf $\mathcal{Z}^n$, die Bedingung (1.6) erfüllt ist.

Im folgenden zeigen wir nun, dass ein conformal predictor konservativ valide ist für austauschbar verteilte Werte. Genaueres zur *Austauschbarkeit* von Verteilungen oder Zufallsvariablen findet sich in Abschnitt 6.1.

Satz 1.9. *(Balasubramanian u. a. 2014, Proposition 1.2) Unter der Annahme, dass die Werte $(z_1, \ldots, z_n)$ von einer austauschbaren Verteilung auf $\mathcal{Z}^n$ erzeugt wurden, ist jeder conformal predictor konservativ. Das bedeutet für den wahren Wert z_n gilt*

$$\mathbb{P}(z_n \in \Gamma^\alpha(z_1, \ldots, z_{n-1})) \geq 1 - \alpha.$$

Beweis. Wir nutzen die Beweisskizze von *Balasubramanian u. a. (2014)* und erweitern diese zu einem vollständigen Beweis.

Seien $(R_1, \ldots, R_n)$ die nonconformity scores des nonconformity measures A. Zur Vereinfachung nehmen wir an, dass alle R_i unterschiedlich sind, allerdings gilt der Satz auch ohne diese Annahme. Der Wert z_n ist genau dann nicht in $\Gamma^\alpha(z_1, \ldots, z_{n-1})$ enthalten, wenn R_n eines der $\lfloor \alpha n \rfloor$ größten Elemente von $(R_1, \ldots, R_n)$ ist. Denn in diesem Fall gilt

$$|\{i = 1, \ldots, n : R_i \geq R_n\}| \leq \lfloor \alpha n \rfloor$$

und für den p-Wert folgt

$$p_{z_n} \leq \frac{\lfloor \alpha n \rfloor}{n} \leq \alpha.$$

Damit ist die Bedingung $p > \alpha$ für z_n nicht erfüllt und somit

$$z_n \notin \Gamma^\alpha(z_1, \ldots, z_{n-1}) = \{z_n \in \mathcal{Z} : p_{z_n} > \alpha\}.$$

Aufgrund der Austauschbarkeit ist die Verteilung der Werte $(z_1, \ldots, z_n)$ invariant unter Permutationen. Mit Lemma 1.11 gilt gleiches auch für die nonconformity scores $(R_1, \ldots, R_n)$. Insbesondere treten alle Permutationen mit der gleichen Wahrscheinlichkeit auf.
Da nach Annahme alle R_i unterschiedlich sind, ist die Wahrscheinlichkeit, dass eines der $\lfloor \alpha n \rfloor$ größten R_i an die n-te Stelle getauscht wird, $\frac{\lfloor \alpha n \rfloor}{n}$.
In der Tat können wir von einem endlichen Wahrscheinlichkeitsraum $(\Omega, \mathbb{P})$ ausgehen. Dabei ist $\Omega = \{R_1, \ldots, R_n\}$ und $\mathbb{P}$, wegen der gleichen Wahrscheinlichkeit jeder Permutation, die uniforme Verteilung auf Ω. Sei $A_{\alpha n}$ die Menge mit den $\lfloor \alpha n \rfloor$ größten R_i und somit $card\, A_{\alpha n} = \lfloor \alpha n \rfloor$. Dann gilt

$$\mathbb{P}(A_{\alpha n}) = \frac{\operatorname{card} A_{\alpha n}}{\operatorname{card} \Omega} = \frac{\lfloor \alpha n \rfloor}{n}.$$

Dies entspricht dann gerade der Wahrscheinlichkeit eines Fehlers. Also gilt

$$\mathbb{P}(z_n \notin \Gamma^\alpha(z_1, \dots, z_{n-1})) = \mathbb{P}(A_{\alpha n}) = \frac{\lfloor \alpha n \rfloor}{n} \leq \alpha$$

und die Behauptung folgt.

□

Bemerkung 1.10. Im Beweis haben wir angenommen, dass alle nonconformity scores R_i unterschiedlich sind. Dies haben wir getan, um den Beweis einfacher zu gestalten, wollen nun allerdings genauer darauf eingehen warum die Aussage auch allgemein gilt. Dazu ordnen wir die nonconformity scores so, dass $R_{(1)} \geq R_{(2)} \geq \cdots \geq R_{(n)}$ gilt. Denn wenn es nun Werte $R_i = R_j$ mit $i \neq j$ gibt, dann kann es sein, dass $R_{(\lfloor \alpha n \rfloor + 1)} = R_{(\lfloor \alpha n \rfloor)}$. In diesem Fall wäre das $\lfloor \alpha n \rfloor$ größte Element gleichzeitig das $\lfloor \alpha n \rfloor - 1$ größte Element und es folgt

$$|\{i = 1, \dots, n : R_i \geq R_{\lfloor \alpha n \rfloor}\}| = \lfloor \alpha n \rfloor + 1$$

und für den p-Wert gilt

$$p_{z_n} = \frac{\lfloor \alpha n \rfloor + 1}{n} > \alpha.$$

Es gilt also $z_n \notin \Gamma^\alpha(z_1, \dots, z_{n-1})$, genau dann wenn R_n eines der *eindeutigen* $\lfloor \alpha n \rfloor$ größten Elemente ist (also $R_n > R_{(\lfloor \alpha n \rfloor + 1)}$). Sei also $A_{\alpha n}$ die Menge mit den eindeutigen $\lfloor \alpha n \rfloor$ größten Elemente. Damit gilt allgemein *card* $A_{\alpha n} \leq \lfloor \alpha n \rfloor$ und wir folgern, dass die Aussage aus Satz 1.9 für beliebige nonconformity score R_i gilt:

$$\mathbb{P}(z_n \notin \Gamma^\alpha(z_1, ..., z_{n-1})) = \mathbb{P}(A_{\alpha n}) \leq \frac{\lfloor \alpha n \rfloor}{n} \leq \alpha.$$

Lemma 1.11. *Für austauschbar verteilte Werte $(z_1, \dots, z_n)$ sind die nonconformity scores $(R_1, \dots, R_n)$ ebenfalls austauschbar verteilt.*

Beweis. Sei π eine Permutation auf $\{1, \dots, n\}$. Nach Definition der nonconformity scores gilt $R_i = A((z_1, \dots, z_n)^\top, z_i) = A((z_{\pi(1)}, \dots, z_{\pi(n)})^\top, z_i)$. Somit können wir den nonconformity score mit Index $\pi(i)$ schreiben als

$$R_{\pi(i)} = A((z_{\pi(1)}, \dots, z_{\pi(n)})^\top, z_{\pi(i)}).$$

Wegen der Austauschbarkeit der Werte z_i folgt dann

$$(R_1, \dots, R_n) = (A((z_1, \dots, z_n)^\top, z_i))_{i \in \mathcal{I}} \overset{d}{=} (A((z_{\pi(1)}, \dots, z_{\pi(n)})^\top, z_{\pi(i)}))_{i \in \mathcal{I}} = (R_{\pi(1)}, \dots, R_{\pi(n)}).$$

(Siehe auch Definition 6.1 für die Austauschbarkeit.)

□

1.3 Optimalität von conformal prediction

Wir haben nun gezeigt, wie ein conformal predictor konstruiert werden kann und, dass dieser valide ist. Aufgrund der einfachen Konstruktion und der geringen Voraussetzungen scheinen conformal predictions ein äußerst vielversprechendes Verfahren zu sein. Es ist allerdings nicht klar, ob diese Methode wirklich sinnvoll ist und es möglicherweise eine Möglichkeit gibt bessere Prädiktoren zu erhalten.

Jedoch können wir zeigen, dass es für jeden confidence predictor (der ebenfalls invariant unter Permutationen ist) einen conformal predictor gibt, der mindestens genau so gut ist.

Definition 1.12. *(Vovk u. a. 2022, S. 53)* Wir sagen ein confidence predictor Γ_1 *dominiert* einen andern confidence predictor Γ_2 für jedes $n \in \mathbb{N}$, jedes $\alpha \in (0,1)$ und alle $z_1, z_2, \ldots,$ wenn gilt

$$\Gamma_1^\alpha(z_1, \ldots, z_{n-1}) \subseteq \Gamma_2^\alpha(z_1, \ldots, z_{n-1}).$$

Proposition 1.13. *(Balasubramanian u. a. 2014, Proposition 1.3) Sei Γ_1 ein confidence predictor, der invariant unter Permutationen und konservativ valide unter Austauschbarkeit ist. Dann existiert ein conformal predictor Γ_2, der Γ_1 dominiert.*

Beweis. Zuerst benötigen wir ein nonconformity measure für Γ_2. Sei also $(z_1, \ldots, z_n) \in \mathcal{Z}^n$ und der nonconformity score für z_i gegeben durch

$$\begin{aligned} R_i &:= 1 - inf\{\alpha : z_i \notin \Gamma_1^\alpha(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)\} \\ &= 1 - sup\{\alpha : z_i \in \Gamma_1^\alpha(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)\}, \end{aligned}$$

für $i = 1, \ldots, n$. Um nun zu überprüfen, ob Γ_1 durch Γ_2 dominiert wird müssen wir zeigen, dass für alle $z_n \in \Gamma_2^\alpha(z_1, \ldots, z_{n-1})$ folgt $z_n \in \Gamma_1^\alpha(z_1, \ldots, z_{n-1})$. Äquivalent wäre, dass aus $z_n \notin \Gamma_1^\alpha(z_1, \ldots, z_{n-1})$, $z_n \notin \Gamma_2^\alpha(z_1, \ldots, z_{n-1})$ folgen würde. Diese Implikation wollen wir zeigen.
Wir fixieren die Werte $(z_1, \ldots, z_n)$ und ein Signifikanzniveau $\tilde{\alpha}$. Angenommen es gilt $z_n \notin \Gamma_1^{\tilde{\alpha}}(z_1, \ldots, z_{n-1})$. Wegen der Annahme, dass Γ_1 konservativ valide ist, folgt, dass

$$z_i \notin \Gamma_1^{\tilde{\alpha}}(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$$

für maximal $\lfloor \tilde{\alpha} n \rfloor$ z_is gilt (Wir bezeichnen zur Vereinfachung die Werte, die nicht in $\Gamma_1^{\tilde{\alpha}}$ liegen als z_i und diejenigen, welche in $\Gamma_1^{\tilde{\alpha}}$ liegen als z_j).
Somit gilt für diese z_is $R_i \geq 1 - \tilde{\alpha}$, da $\tilde{\alpha} \in \{\alpha : z_i \notin \Gamma_1^\alpha(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)\}$ und das Infimum dieser Menge kleiner oder gleich $\tilde{\alpha}$ ist. Für die übrigen nonconformity scores, mit

$$z_j \in \Gamma_1^\alpha(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$$

(mindestens $n - \lfloor \tilde{\alpha} n \rfloor$), gilt dagegen $R_j \leq 1 - \tilde{\alpha}$, da $\tilde{\alpha} \in \{\alpha : z_i \in \Gamma_1^\alpha(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)\}$ und das Supremum dieser Menge größer oder gleich $\tilde{\alpha}$ ist. Daher muss R_n unter den $\lfloor \tilde{\alpha} n \rfloor$ größten R_i sein. Und es folgt direkt $z_n \notin \Gamma_2^{\tilde{\alpha}}(z_1, \ldots, z_{n-1})$.
Folglich haben wir einen conormal predictor Γ_2 gefunden, der Γ_1 dominiert. □

1.4 Geglätteter conformal predictor

Im vorherigen Abschnitt haben wir gezeigt, dass ein conformal predictor konservativ valide ist und die Ergebnisse somit mit der geforderten Wahrscheinlichkeit korrekt sind. Allerdings soll ein Konfidenzbereich Γ auch ein effizientes Ergebnis liefern und daher ein möglichst kleinen Bereich festlegen, der z_n mit Wahrscheinlichkeit $1 - \alpha$ enthält. Bei der Definition von Γ haben wir bereits festgelegt, dass ein größeres Signifikanzniveau α einen kleineren Konfidenzbereich bedingt (siehe (1.1)). Aus diesem Grund ist es erstrebenswert einen exakt validen conformal predictor zu konstruieren, um ein möglichst effizientes Ergebnis zu gewährleisten.

Bemerkung 1.14. (*vgl. Fontana u. a. 2023, Kapitel 2.1.4*) Der effizienteste Konfidenzbereich wäre trivial $\Gamma^\alpha(z_1, \ldots, z_{n-1}) = \emptyset$. Allerdings würde diese Menge offensichtlich keinen Nutzen haben, da sie keine Werte enthält. Genauso liefert der Konfidenzbereich $\Gamma^\alpha(z_1, \ldots, z_{n-1}) = \mathcal{Z}$ auch keine neuen Informationen, da dieser jeden möglichen Wert für z_n enthält und keine Werte ausschließt. Deswegen wird ein Konfidenzbereich daran gemessen, wie valide und effizient dieser ist.

Unser Ziel ist es daher, in diesem Abschnitt einen exakter conformal predictor zu konstruieren, der die Bedingung

$$\mathbb{P}(z_n \in \Gamma^\alpha(z_1, ..., z_{n-1})) = 1 - \alpha \tag{1.7}$$

erfüllt.

Definition 1.15. (*vgl. Vovk u. a. 2022, S. 10*) Ein conformal predictor Γ^α ist *exakt valide*, wenn für jede austauschbare Wahrscheinlichkeitsverteilung $\mathbb{P}$ auf $\mathcal{Z}^n$, die Bedingung (1.7) gilt.

Dazu modifizieren wir unseren bisherigen conformal predictor, indem wir den Fall $R_i = R_n$ gesondert betrachten, bei dem die nonconformity scores von z_i mit $i = 1, \ldots, n$ und z_n übereinstimmen. Bisher wird der p-Wert für jedes $i = 1, \ldots, n$ mit $R_i = R_n$ um $1/n$ erhöht. Stattdessen soll der p-Wert in solchen Fällen nur noch um einen zufälligen Wert zwischen 0 und $1/n$ erhöht werden.

Definition 1.16. (*vgl. Fontana u. a. 2023, Kapitel 2.1.4*) Der *geglättete p-Wert* für z_n ist definiert als

$$\bar{p}_{z_n} = \frac{|\{i = 1, ..., n : R_i > R_n\}| + \tau_n \, |\{i = 1, ..., n : R_i = R_n\}|}{n}, \tag{1.8}$$

wobei die Zufallsvariable τ_n uniform verteilt ist auf $(0, 1)$.

Der *geglättete conformal predictor* ergibt sich dann analog zu (1.5) mit dem einzigen Unterschied, dass der p-Wert p_{z_n} durch den geglätteten p-Wert $\bar{p}_{z_n}$ ersetzt wird.

$$\begin{aligned} \bar{\Gamma}^\alpha(z_1, \ldots, z_{n-1}) &= \left\{ z_n \in \mathcal{Z} : \frac{|\{i = 1, ..., n : R_i > R_n\}| + \tau_n \, |\{i = 1, ..., n : R_i = R_n\}|}{n} > \alpha \right\} \\ &= \{ z_n \in \mathcal{Z} : \bar{p}_{z_n} > \alpha \} \end{aligned} \tag{1.9}$$

Lemma 1.17. *(Spezialfall von Lemma 11.8 aus Vovk u. a. 2022) Seien die nonconformity scores $R_1, \ldots, R_n$ austauschbar und $\tau \sim Unif(0, 1)$ unabhängig von $R_1, \ldots, R_n$. Dann gilt*

$$\mathbb{P}(\bar{p}_{z_n} \leq \alpha) = \alpha$$

für alle $\alpha \in [0, 1]$, wobei $\bar{p}_{z_n}$ der geglättete p-Wert mit Zufallsvariable τ, aus Definition 1.16, ist.

Beweis. Wir nutzen die Idee des allgemeineren Lemma 11.8 aus *Vovk u. a. (2022)*, um einen spezielleren Beweis für Lemma 1.17 zu erhalten.

Sei

$$\bar{p}_{z_j} = \frac{|\{i = 1, ..., n : R_i > R_j\}| + \tau \, |\{i = 1, ..., n : R_i = R_j\}|}{n}$$

für $j = 1, \ldots, n$. Mit der Austauschbarkeit der nonconformity scores haben wir $\bar{p}_{z_n} \overset{d}{=} \bar{p}_{z_j}$ für jedes j.
Außerdem definieren wir für jeden Wert $\tilde{z} \in \sigma_n = \{z_1, \ldots, z_n\}$

$$p^+(\tilde{z}) = \frac{|\{i = 1, ..., n : A(\sigma_n, z_i) \geq A(\sigma_n, \tilde{z})\}|}{n},$$

$$p^-(\tilde{z}) = \frac{|\{i = 1, ..., n : A(\sigma_n, z_i) > A(\sigma_n, \tilde{z})\}|}{n}.$$

Dabei entspricht $p^+(\tilde{z})$ dem p-Wert aus Definition 1.6 und es ist klar, dass $p^-(\tilde{z}) \leq p^+(\tilde{z})$.
Für den geglätteten p-Wert gilt dann $\bar{p}_{z_n} = p^- + \tau_n(p^+ - p^-)$.
Jedes Beispiel $\tilde{z}$ erfüllt genau einen der folgenden drei Fälle

(1) $p^+(\tilde{z}) \leq \alpha$

(2) $p^-(\tilde{z}) > \alpha$

(3) $p^-(\tilde{z}) \leq \alpha < p^+(\tilde{z})$.

Nach Definition gilt $\bar{p}_{z_n} \leq \alpha$ wenn z_n Fall (1) erfüllt und $\bar{p}_{z_n} > \alpha$ wenn es (2) erfüllt. Unter der Bedingung, dass wir σ_n kennen, ist der Anteil der Werte, welche (1) erfüllen p^-. Also

$$\mathbb{P}((1)|\sigma_n) = p^-.$$

Nun bestimmen wir die Wahrscheinlichkeit für die z_n, welche ebenfalls $\bar{p}_{z_n} \leq \alpha$ erfüllen, aber durch (3) charakterisiert werden. Dazu bemerken wir, dass

$$\bar{p}_{z_n} = p^- + \tau_n(p^+ - p^-) \leq \alpha$$

äquivalent zu

$$\tau_n \leq \frac{\alpha - p^-}{p^+ - p^-}$$

ist. Da $\tau_n \sim Unif(0, 1)$, entspricht dies der Wahrscheinlichkeit für $\bar{p}_{z_n} \leq \alpha$. Und der Anteil der z_n, welche (3) erfüllen entspricht $p^+ - p^-$. Damit haben wir nun

$$\mathbb{P}(\{\bar{p}_{z_n} \leq \alpha\} \cap (3)|\sigma_n) = (p^+ - p^-)\frac{\alpha - p^-}{p^+ - p^-} = \alpha - p^-.$$

Für die gesamte bedingte Wahrscheinlichkeit folgt

$$\begin{aligned}\mathbb{P}(\bar{p}_{z_n} \leq \alpha|\sigma_n) &= \mathbb{P}((1)|\sigma_n) + \mathbb{P}(\{\bar{p}_{z_n} \leq \alpha\} \cap (3)|\sigma_n) \\ &= p^- + \alpha - p^- \\ &= \alpha.\end{aligned}$$

Durch Marginalisieren erhalten wir

$$\mathbb{P}(\bar{p}_{z_n} \leq \alpha) = \mathbb{E}[\mathbb{P}(\bar{p}_{z_n} \leq \alpha|\sigma_n)] = \mathbb{E}[\alpha] = \alpha.$$

□

Satz 1.18. *(Vovk u. a. 2022, Proposition 2.4) Angenommen die Beobachtungen $(z_1, \ldots, z_n)$ sind austauschbar verteilt. Dann ist jeder geglättete conformal prediktor $\bar{\Gamma}^\alpha$ exakt valide.*

Beweis. Wir wissen nach Lemma 1.11, dass die conformity scores $R_1, \ldots, R_n$ austauschbar sind. Daher können wir Lemma 1.17 anwenden und erhalten $\mathbb{P}(\bar{p}_{z_n} \leq \alpha) = \alpha$ für alle $\alpha \in [0, 1]$. Wegen der Äquivalenz

$$z_n \in \bar{\Gamma}^\alpha(z_1, \ldots, z_{n-1}) \Longleftrightarrow \bar{p}_{z_n} > \alpha.$$

folgt die Behauptung

$$\begin{aligned}\mathbb{P}(z_n \in \bar{\Gamma}^\alpha(z_1, \ldots, z_{n-1})) &= \mathbb{P}(\bar{p}_{z_n} > \alpha) \\ &= 1 - \mathbb{P}(\bar{p}_{z_n} \leq \alpha) \\ &= 1 - \alpha.\end{aligned}$$

□

Bemerkung 1.19. Satz 1.18 ist eine Verallgemeinerung von Satz 1.9. Denn für die p-Werte (1.4) und (1.7) gilt $\bar{p}_{z_n} \leq p_{z_n}$ und es somit ist klar, dass $\{z_n \in \mathcal{Z} : \bar{p}_{z_n} > \alpha\} \subseteq \{z_n \in \mathcal{Z} : p_{z_n} > \alpha\}$. Daher folgt mit Satz 1.18

$$\mathbb{P}(p_{z_n} > \alpha) \geq \mathbb{P}(\bar{p}_{z_n} > \alpha) = 1 - \alpha.$$

Die obere Grenze für den conformal predictor in Satz 1.9 lässt sich sogar explizit bestimmen. Dabei stellen wir fest, dass der Vorhersagebereich nicht viel konservativer wird als die untere Grenze $1 - \alpha$ vorgibt.

Proposition 1.20. *(Lei u. a. 2018, Theorem 2.1) In der Situation von Satz 1.9 gilt, für fast sicher unterschiedliche nonconformity scores $R_1, \ldots, R_n$, die obere Schranke*

$$\mathbb{P}(z_n \in \Gamma^\alpha(z_1, \ldots, z_{n-1})) \leq 1 - \alpha + \frac{1}{n}.$$

Beweis. (*vgl. Lei u. a. 2018, Proof of Theorem 2.1*) Da die nonconformity scores $R_1, \ldots, R_n$, nach Lemma 1.11, austauschbar sind und fast sicher unterschiedlich, ist der p-Wert p_{z_n} uniform verteilt auf $\{1/n, \ldots, n/n\}$. Wenn $y_n \in \Gamma^\alpha(z_1, \ldots, z_{n-1})$, dann ist R_n unter den $\lceil (1-\alpha)n \rceil$ kleinsten $R_1, \ldots, R_n$. Wenn $R_{(1)} < \cdots < R_{(n)}$ die nach Größe geordneten nonconformity scores sind (wir können $<$ annehmen, da die R_i f.s. unterschiedlich sind), dann gilt

$$\mathbb{P}(z_n \in \Gamma^\alpha(z_1, \ldots, z_{n-1})) = \mathbb{P}(R_n < R_{(\lceil (1-\alpha)n \rceil + 1)}).$$

Wegen der Gleichverteilung, der nonconformity scores, folgt

$$\mathbb{P}\left(R_n < R_{(\lceil (1-\alpha)n \rceil + 1)}\right) \leq \frac{\lceil (1-\alpha)n \rceil}{n} \leq \frac{(1-\alpha)n + 1}{n} = 1 - \alpha + \frac{1}{n}.$$

□

Aus diesem Grund ist es häufig ausreichend, den konservativ validen conformal predictor zu wählen. Dennoch kann es bei bestimmten Varianten von conformal prediction sinnvoll sein, eine geglättete Version zu verwenden, wenn die obere Schranke deutlich größer wird als in Proposition 1.20 (siehe Abschnitt 3.2).

2 Regression

2.1 Objekte und Kategorien

In diesem Abschnitt führen wir eine spezialisierte Version von conformal prediction ein und orientieren uns an *Fontana u. a. (2023), Kapitel 2.2.* Dabei haben wir immer noch Beobachtungen $z_1, z_2, \dots \in \mathcal{Z}$, jedoch besteht jede Beobachtung z_i aus einem Objekt $x_i \in \mathcal{X}$ und einer Kategorie (engl. label) $y_i \in \mathcal{Y}$. Es gilt somit

$$z_i = (x_i, y_i) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}. \tag{2.1}$$

Wir bezeichnen $\mathcal{X}$ als Objektraum und $\mathcal{Y}$ als Kategorienraum. Sei außerdem $\mathcal{X}$ nicht leer und $\mathcal{Y}$ soll mindestens zwei verschiedene Elemente enthalten.
Die größte Veränderung zu den bisherigen Voraussetzungen ist, dass nicht nur $(x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ gegeben sind, sondern auch das Objekt x_n. Damit müssen wir nur noch die Kategorie y_n vorhersagen. Die Menge in (1.5), welche bisher z_n vorhersagen sollte, bezieht sich nun nur noch auf Werte aus $\mathcal{Y}$ und wir schreiben

$$\begin{aligned} \Gamma^\alpha(x_1, y_1, \dots, x_{n-1}, y_{n-1}, x_n) &= \{y_n : (x_n, y_n) \in \Gamma^\alpha(z_1, \dots, z_{n-1})\} \\ &= \{y_n \in \mathcal{Y} : p_{(x_n, y_n)} > \alpha\}. \end{aligned} \tag{2.2}$$

Wir bemerken, dass dies nun eine Abbildung der Form

$$\Gamma : \mathcal{Z}^{n-1} \times \mathcal{X} \times (0,1) \to 2^{\mathcal{Y}}$$

ist. Der Unterschied zu dem confidence predictor in Definition 1.1 besteht darin, dass wir nun ein weiteres Argument $x \in \mathcal{X}$ nutzen können und auf eine Teilmenge von $\mathcal{Y}$ abbilden. Allerdings ist der p-Wert weiterhin wie in (1.4) und das zugehörige nonconformity measure wie in Definition 1.3.

Beispiel 2.1. Ein einfaches Beispiel für nonconformity scores, mit $\mathcal{Y} \subseteq \mathbb{R}$, ist

$$R_i = A(((x_1, y_1), \dots, (x_n, y_n))^\top, (x_i, y_i)) = \left| \frac{\sum_{j=1}^n y_j}{n} - y_i \right|, \quad i = 1, \dots, n.$$

Wir sehen, dass es nur eine sehr geringe Änderung zu Beispiel 1.4 braucht, um einen passenden nonconformity score im Fall von (2.1) zu erhalten.

Da (2.1) ein Spezialfall, der im vorherigen Kapitel betrachteten Beobachtungen ist, gelten auch alle bisherigen Ergebnisse für diesen Fall.

2.2 Beispiel Ridge Regression

In Regressions Problemen ist ein typisches nonconformity measure

$$R_i = \Delta(y_i, \hat{f}(x_i)) \tag{2.3}$$

wobei $\Delta : \mathcal{Y}^2 \to \mathbb{R}$ ein Maß, welches den Unterschied von zwei Kategorien misst, und $\hat{f}$ eine Vorhersage-Funktion ist (vgl. Fontana u. a. 2023). Die Vorhersage von $\hat{f}$ basiert auf dem Trainingsdatensatz $(z_1, \dots, z_{n-1})$. Die Funktion $\hat{f}(x)$ soll also den Schätzer $\hat{y}$ für die Kategorie von x angeben. Wir werden uns insbesondere mit dem Fall beschäftigen, wenn $\Delta(y, \hat{y}) = |y - \hat{y}|$ gilt.

Wenn wir allerdings nun Regression auf conformal prediction anwenden wollen, tritt ein Problem bei dem Bestimmen der Menge (2.2) auf. Denn es muss für jede mögliche Kategorie y der p-Wert ermittelt werden, wie wir schon in Bemerkung 1.7 festgestellt haben. Im Fall von Regression gibt es aber unendlich viele mögliche Werte für y, da $\mathcal{Y} = \mathbb{R}$ gilt (*vgl. Fontana u. a. 2023*). Wir müssen also einen Weg finden den Vorhersagebereich auf andere Weise zu ermitteln. Eine relativ einfache Möglichkeit dafür besteht zum Beispiel bei der Ridge Regression. In diesem Fall können wir mittels eines Algorithmus ein explizites Intervall berechnen, ohne unendlich viele Iterationen zu benötigen.

Wir betrachten nun den Fall von Ridge Regression, welcher einer der bekanntesten Algorithmen für Regression ist. Dafür nehmen wir an, dass $\mathcal{X} \subseteq \mathbb{R}^p$ und $\mathcal{Y} \subseteq \mathbb{R}$ gelten. Das lineare Modell, für den unbekannten Parametervektor $\boldsymbol{\beta} \in \mathbb{R}^p$, ist gegeben durch

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.4}$$

mit dem Vektor $\mathbf{Y} = (y_1, \ldots, y_n)^\top$, der Matrix $\mathbb{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top \in \mathbb{R}^{n \times p}$ und einem Zufallsvektor $\boldsymbol{\epsilon} \in \mathbb{R}^n$ von u.i.v Zufallsvariablen. Die Designematrix $\mathbb{X}$ hat vollen Rang p (mit $p \leq n$) und Zeilen $\mathbf{x}_i \in \mathbb{R}^p$.

Bemerkung 2.2. Der Vektor $\boldsymbol{\beta}$ wird, in der Methode der Ridge Regression, als Minimierer des Ausdrucks

$$a \left\|\mathbf{b}\right\|^2 + \left\|\mathbf{Y} - \mathbb{X}\mathbf{b}\right\|^2 = a \sum_{i=1}^{p} b_i + \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{b})^2, \quad \mathbf{b} \in \mathbb{R}^p$$

geschätzt. Die konstante $a > 0$ wird auch als Ridge Parameter bezeichnet. Der Ridge-Regressions-Schätzer (RRS) ist also gegeben als

$$\hat{\beta}^{RR} = \underset{\mathbf{b} \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ a \left\|\mathbf{b}\right\|^2 + \left\|\mathbf{Y} - \mathbb{X}\mathbf{b}\right\|^2 \right\}. \tag{2.5}$$

Für den Wert x ist die vorhergesagte Kategorie dann

$$\hat{f}(x) = \hat{y} = x \cdot \hat{\boldsymbol{\beta}}^{RR}.$$

Proposition 2.3. *(vgl. Fontana u. a. 2023, S.10) Im linearen Modell aus* (2.4) *erhalten wir als RRS* $\hat{\boldsymbol{\beta}}^{RR} = (\mathbb{X}^\top \mathbb{X} + aI_p)^{-1} \mathbb{X}^\top \mathbf{Y}$ *und somit als nonconformity score für die Ridge Regression*

$$R_i = |y_i - \mathbf{x}_i^\top (\mathbb{X}^\top \mathbb{X} + aI_p)^{-1} \mathbb{X}^\top \mathbf{Y}|. \tag{2.6}$$

Beweis. Nach *Saleh u. a. (2019) (Abschnitt 1.2)* gilt, dass der eindeutige RRS gegeben ist durch

$$\hat{\boldsymbol{\beta}}^{RR} = (\mathbb{X}^\top \mathbb{X} + aI_p)^{-1} \mathbb{X}^\top \mathbf{Y}.$$

Für den Wert $\mathbf{x} \in \mathcal{X}^p$ ist die geschätzte Kategorie y nun

$$\hat{y} = \mathbf{x}^\top \hat{\boldsymbol{\beta}}^{RR} = \mathbf{x}^\top (\mathbb{X}^\top \mathbb{X} + aI_p)^{-1} \mathbb{X}^\top \mathbf{Y}$$

und für das nonconformity measure gilt folglich

$$\begin{aligned} R_i &= \Delta(y_i, \hat{y}_i) \\ &= |y_i - \hat{y}_i| \\ &= |y_i - \mathbf{x}_i^\top (\mathbb{X}^\top \mathbb{X} + aI_p)^{-1} \mathbb{X}^\top \mathbf{Y}|. \end{aligned}$$

□

Bemerkung 2.4. Die Methode der Kleinsten Quadrate ist ein Spezialfall der Ridge Regression, bei dem der Ridge Parameter $a = 0$ ist. Somit gilt für den Kleinste-Quadrate-Schätzer $\hat{\beta}^{KQ} = (\mathbb{X}^T \mathbb{X})^{-1} X^\top \mathbf{Y}$ und daher

$$R_i = |y_i - \mathbf{x}_i^\top (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^\top \mathbf{Y}|.$$

Mit den nonconformity scores können wir nun den entsprechenden p-Wert ermitteln, womit wir dann den conformal predictor Γ^α bestimmen können. Nun wird allerdings die am Anfang bereits erwähnte Problematik deutlich, dass $\mathcal{Y}$ unendlich viele Elemente enthält. Denn nach unserer Definition des Vorhersagebereichs müssen wir den p-Wert für jedes y berechnen, um zu entscheiden, ob wir es zu der Menge Γ^α hinzufügen. Dies ist aber wegen der Unendlichkeit von $\mathcal{Y} = \mathbb{R}$ nicht möglich.

Es gibt dennoch einen machbaren Weg den gesuchten Vorhersagebereich zu bestimmen, indem wir den p-Wert umschreiben und durch Abschätzen ein Intervall bestimmen, welches mindestens $100 \cdot (1 - \alpha)\%$ der Trainingsdaten enthält. Allerdings werden wir dafür eine Abwandlung der nonconformity scores nutzen, um eine Lösung ohne Fallunterscheidung zu finden. Daher werden wir $R_i = y_i - \hat{y}_i$ als nonconformity score für den Rest dieses Abschnittes festlegen. Einen noch effizienteren Algorithmus für den nonconformity score $R_i = |y_i - \hat{y}_i|$ werden wir in Abschnitt 2.3 kennenlernen.
Für das weitere Vorgehen, welches auf dem von *Vovk u. a. (2022), Kapitel 2.3.2* basiert, benötigen wir noch folgende Definitionen.

Definition 2.5. Wir bezeichnen die Matrix

$$H = \mathbb{X}(\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top$$

als die *Hut-Matrix*, da diese die Werte y_i in die Form $\hat{y}_i$ transformiert. Die Werte $\hat{y}_i$ werden auch als *angepasste Werte* bezeichnet. Den Vektor der angepassten Werte können wir dann schreiben als

$$\hat{\mathbf{Y}} = (\hat{y}_1, \ldots, \hat{y}_n)^\top = H\mathbf{Y}.$$

Den Vektor der nonconformity scores, $R_i = y_i - \hat{y}_i$, können wir daher wie folgt darstellen:

$$(R_1, \ldots, R_n)^\top = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - H\mathbf{Y} = (I_n - H)\mathbf{Y}.$$

Definition 2.6. (Vovk u. a. 2022, *S. 39*) Seien

$$\begin{aligned} A &:= (I_n - H)(y_1, \ldots, y_{n-1}, 0)^\top \\ B &:= (I_n - H)(0, \ldots, 0, 1)^\top \end{aligned}$$

Vektoren $A = (a_1, \dots, a_n)^\top \in \mathbb{R}^n$ und $B = (b_1, \dots, b_n)^\top \in \mathbb{R}^n$. Dann gilt

$$(I_n - H)\mathbf{Y} = A + y_n B$$

und wir definieren die Menge

$$S_i = \{y_n : a_i + b_i y_n \geq a_n + b_n y_n\}. \tag{2.7}$$

Bemerkung 2.7. Es wird sofort deutlich, dass wir die Menge (2.7) ebenfalls schreiben können als $S_i = \{y_n : R_i \geq R_n\}$. Denn für die nonnconformity scores gilt $R_i = a_i + b_i y_n$, für alle $i = 1, \dots, n$.

Somit folgt für den p-Wert nach Definition 1.6,

$$p_{y_n} = \frac{|\{i = 1 \dots, n : R_i \geq R_n\}|}{n} = \frac{|\{i = 1, \dots, n : y_n \in S_i\}|}{n}. \tag{2.8}$$

Mit diesen Voraussetzungen können wir nun ein explizites Intervall berechnen, welches mindestens $100 \cdot (1 - \alpha)\%$ der n Kategorien $y_1, \dots, y_n$ überdeckt.

Proposition 2.8. *Es seien die folgenden Annahmen gegeben:*
(A1) Die Zufallsobjekte $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, 2, \dots$ sind u.i.v..
(A2) Die Zweitmomentmatrix $\mathbb{E}(\mathbf{x}_1 \mathbf{x}_1^T)$ von $\mathbf{x}_1$ existiert und ist invertierbar.
(A3) Der Vektor $\beta \in \mathbb{R}^p$ ist unabhängig von $\mathbf{x}_1, \mathbf{x}_2, \dots$.
(A4) Die Kategorien $y_1, y_2, \dots$ werden erzeugt durch $y_i = \beta \cdot \mathbf{x}_i + \epsilon_i$, wobei ϵ_i Gauss'sche Störgrößen sind mit Verteilung $\mathcal{N}(0, \sigma^2)$, die untereinander unabhängig sind sowie unabhängig von den Objekten $\mathbf{x}_i$ und von β.

Dann ist für hinreichend große n der beidseitige conformal predictor für die Ridge Regression, mit nonconformity score $R_i = a_i + b_i y_n$, gegeben durch das Intervall

$$\Gamma^\alpha = [t_{(\lfloor(\alpha/2)n\rfloor)}, t_{(\lceil(1-\alpha/2)n\rceil)}],$$

wobei $t_i := \dfrac{a_i - a_n}{b_n - b_i}$ und $t_{(\lfloor(\alpha/2)n\rfloor)}, t_{(\lceil(1-\alpha/2)n\rceil)}$ den, nach Größe geordneten t_i, $t_{(1)} \leq \dots \leq t_{(n-1)}$ entsprechen.

Beweis. (*vgl. Vovk u. a. 2022, S. 40-41*) Durch die Annahmen (A1)-(A4) folgt mit Lemma 2.9 und 2.10, dass wir für hinreichend große n, $b_n > b_i$ für alle $i < n$ fast sicher annehmen können. Wir können dann die Ungleichung in (2.7) umschreiben zu

$$y_n \leq \frac{a_i - a_n}{b_n - b_i}$$

und erhalten somit die Menge

$$S_i = (-\infty, t_i], \text{ mit } t_i = \frac{a_i - a_n}{b_n - b_i},$$

für $i = 1, \dots, n-1$ und $S_n = \mathbb{R}$. Es folgt somit für den p-Wert aus (2.8)

$$p_{y_n} = \frac{|i = 1, \dots, n-1 : t_i \geq y_n| + 1}{n},$$

wobei wir beachten, dass t_n nicht definiert ist. Seien nun $t_{(1)} \leq, \dots, \leq, t_{(n-1)}$ die nach Größe

geordneten t_i, dann gilt für den rechtsseitigen conformal predictor

$$\Gamma_r^\alpha = (-\infty, t_{(\lceil (1-\alpha)n \rceil)}].$$

Dieses Intervall enthält die kleinsten $\lceil (1-\alpha)n \rceil$ Werte von $(t_1, \ldots, t_n)$. Analog folgt für den linksseitigen conformal predictor

$$\Gamma_l^\alpha = [t_{\lfloor \alpha n \rfloor}, \infty),$$

welcher das Intervall ist, das die größten $n - \lfloor \alpha n \rfloor + 1 = \lfloor n - \alpha n \rfloor + 1 = \lceil (1-\alpha)n \rceil$ Werte von $(t_1, \ldots, t_n)$ enthält. Wir bemerken, dass wenn es Werte $t_i = t_j$ mit $i \neq j$ gibt, die Intervalle Γ_r^α und Γ_l^α mehr als $\lceil (1-\alpha)n \rceil$ Elemente aus $(t_1, \ldots, t_n)$ enthalten können.
Um nun einen beidseitigen conformal predictor zu erhalten kombinieren wir die beiden einseitigen Intervalle. Damit gilt

$$\Gamma^\alpha = \Gamma_r^{\alpha/2} \cap \Gamma_l^{\alpha/2} = [t_{(\lfloor (\alpha/2)n \rfloor)}, t_{(\lceil (1-\alpha/2)n \rceil)}]$$

□

Damit wir die Aussage so beweisen konnten, mussten wir annehmen, dass $b_n > b_i$ gilt für $i < n$. Wie die folgenden zwei Lemmata zeigen ist diese Aussage wahr, allerdings gilt dies nur für große n. Die Beweise basieren dabei auf *Vovk u. a. (2022)*, werden hier allerdings zum besseren Verständnis deutlich kleinschrittiger präsentiert.

Lemma 2.9. *(Vovk u. a. 2022, Lemma 2.16) Angenommen wir haben*

$$(c \cdot \mathbf{x}_n)^2 < \sum_{i=1}^{n-1} (c \cdot \mathbf{x}_i)^2 + a\|c\|^2, \tag{2.9}$$

für jedes $c \in \mathbb{R}^p \setminus \{0\}$, wobei $\|\cdot\|$ für die euklidische Norm steht und $c \cdot \mathbf{x}$ das Skalarprodukt von c und $\mathbf{x}$ bezeichnet. Dann ist $b_n > b_i$ für alle $i = 1, \ldots, n-1$.

Beweis. Wir zeigen die Aussage zuerst für den Fall $a = 0$. In diesem Fall ist die Hut-Matrix H die Projektionsmatrix auf den Spaltenraum von $\mathbb{X}$. Diesen bezeichnen wir im weiteren mit $\mathcal{C} \subset \mathbb{R}^n$. Und somit ist $I_n - H$ die Projektionsmatrix auf das orthogonale Komplement $\mathcal{C}^\perp$ von $\mathcal{C}$. Damit ist $B = (b_1, \ldots, b_n)^\top = (I_n - H)e_n$ mit $e_n = (0, \ldots, 0, 1)^\top$ eine Projektion auf $\mathcal{C}^\perp$. Die Hyperebene $\mathbb{R}^{n-1} \times \{0\}$ hat Normalenvektor e_n, da dieser offensichtlich senkrecht auf der Hyperebene steht. Somit entspricht der Winkel θ von $\mathcal{C}$ und $\mathbb{R}^{n-1} \times \{0\}$ dem Winkel $90° - \beta$, wobei β der Winkel zwischen b und e_n ist. Es folgt somit

$$\cos(\beta) = \frac{b \cdot e_n}{\|b\| \cdot \|e_n\|} = \frac{b_n}{\|b\|}.$$

Angenommen es gilt $b_n \leq b_i$ für $i < n$. Dann folgt

$$\frac{b_n}{\|b\|} = \frac{b_n}{\sqrt{\sum_{i=1}^n b_i^2}} \leq \frac{b_n}{\sqrt{n} \cdot b_n} = \frac{1}{\sqrt{n}}$$

und somit

$$\beta = \arccos\left(\frac{b_n}{\|b\|}\right) \geq \arccos\left(\frac{1}{\sqrt{n}}\right) \geq \arccos\left(\frac{1}{\sqrt{2}}\right) = 45°.$$

Den Winkel θ können wir daher mit β abschätzen als $\theta = 90° - \beta \leq 45°$.

Es gilt also $b_n \leq b_i$ nur wenn der Winkel zwischen $\mathcal{C}^\perp$ und der Hyperebene $\mathbb{R}^{n-1} \times \{0\}$ maximal $45°$ ist. Dies ist äquivalent zu der Aussage, dass der Winkel zwischen $\mathcal{C}$ und der

Hyperebene $\mathbb{R}^{n-1} \times \{0\}$ mindestens 45° beträgt. Das würde bedeuten, dass für ein Element $v = (c \cdot \mathbf{x}_1, \ldots, c \cdot \mathbf{x}_n)^\top \in \mathcal{C}$, mit $c \cdot \mathbf{x}_n = 1$, folgen müsste, dass der übrige Vektor $(c \cdot \mathbf{x}_1, \ldots, c \cdot \mathbf{x}_{n-1})$ eine Länge von maximal 1 hat. In dem Fall ist jedoch

$$(c \cdot \mathbf{x}_n)^2 = 1 \geq \|(c \cdot \mathbf{x}_1, \ldots, c \cdot \mathbf{x}_{n-1})\| = \sum_{i=1}^{n-1} (c \cdot \mathbf{x}_i)^2.$$

Dies ist ein Widerspruch zur Voraussetzung (2.9) und es folgt $b_n > b_i$.
Es folgt nun direkt, dass das Lemma auch für $a > 0$ gilt, da wir diesen Fall sehr einfach auf den Fall mit $a = 0$ reduzieren können. Sei also $a > 0$. Dann erweitern wir die gegebene Designe-Matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$ mit den p Dummy-Objekten $\sqrt{a}e_i \in \mathbb{R}^p$, $i = 1, \ldots, p$, welche alle Kategorie (bzw. lable) 0 haben.

□

Lemma 2.10. *(Vovk u. a. 2022, Lemma 2.17) Der Fall $b_n \leq b_i$ für $i < n$ ist fast sicher ausgeschlossen für hinreichend große n unter den Annahmen (A1)-(A4).*

Beweis. Wir zeigen, dass (2.9) ab einem bestimmten n gilt. Wir setzen, ohne Beschränkung der Allgemeinheit, $a := 0$. Sei

$$\Sigma_l = \frac{1}{l} \sum_{i=1}^{l} \mathbf{x}_i \mathbf{x}_i^\top.$$

Dann gilt wegen dem starken Gesetz der großen Zahlen $\lim_{l \to \infty} \Sigma_l = \Sigma$ *f.s.* und es folgt die Konvergenz der kleinsten Eigenwerte

$$|\lambda_{min}(\Sigma_l) - \lambda_{min}(\Sigma)| \underset{l \to \infty}{\longrightarrow} 0 \quad f.s.,$$

wobei $\lambda_{min}(\cdot)$ der kleinste Eigenwert einer Matrix ist. Daher existiert fast sicher ein $n_1 \in \mathbb{N}$, so dass für alle $l \geq n_1$

$$\lambda_{min}(\Sigma_l) > \frac{1}{2} \lambda_{min}(\Sigma) > 0.$$

Da außerdem gilt $\|\mathbf{x}_n\|^2 / n \to 0 \quad f.s.$ für $n \to \infty$, wegen (A1) und (A2), existiert auch ein $n_2 \in \mathbb{N}$ für das alle $n \geq n_2$ die Ungleichung

$$\frac{\|\mathbf{x}_n\|^2}{n-1} < \frac{1}{2} \lambda_{min}(\Sigma)$$

erfüllen. Damit erhalten wir die Abschätzung

$$\begin{aligned}
\frac{1}{n-1} \sum_{i=1}^{n-1} (c \cdot \mathbf{x}_i)^2 &= c^T \Sigma_{n-1} c \\
&\geq \lambda_{min}(\Sigma_{n-1}) \|c\|^2 \\
&> \frac{1}{2} \lambda_{min}(\Sigma) \|c\|^2 \\
&> \frac{\|c\|^2 \|\mathbf{x}_n\|^2}{n-1} \\
&\geq \frac{(c \cdot \mathbf{x}_n)^2}{n-1} \quad \text{(Cauchy–Schwarz-Ungleichung)},
\end{aligned}$$

welche für $c \neq 0$ und $n \geq \max(n_1, n_2)$ gilt. □

Proposition 2.8 garantiert ein robustes Prädiktionsintervall und liefert unter den Voraussetzungen (A1)-(A4) immer ein valides Ergebnis. Allerdings sind dies sehr starke Vor-

aussetzungen und der große Vorteil von conformal prediction war bisher, dass wir mit der Austauschbarkeit nur sehr geringe Voraussetzungen benötigt haben. Somit ist das Intervall aus Proposition 2.8, eine gute Lösung wenn die Bedingungen gegeben sind, aber nicht so allgemein gültig, wie es für uns wünschenswert wäre. Aus diesem Grund haben wir diesen Ansatz nur beispielhaft für die Ridge Regression betrachtet und keinen allgemeinen Ansatz gewählt. Ein weiteres Problem, welches grundsätzlich bei conformal prediction auftritt, ist der hohe Rechenaufwand (vgl. Bemerkung 1.7). Wie wir im folgenden Abschnitt sehen werden gibt es eine effizientere Methode, um ein valides Intervall zu konstruieren. Zudem sind wir nicht auf so starke Voraussetzungen angewiesen wie es bei Proposition 2.8 der Fall ist.

2.3 Split conformal predictor

Wie wir nun erkannt haben, ist conformal prediction ein sehr rechenaufwendiges Verfahren. Daher ist es von großem Interesse möglichst viel Rechenaufwand einzusparen, ohne an Genauigkeit zu verlieren. Eine schnellere Methode, die das wiederholte Trainieren der Regressionsfunktion vermeidet, wird als *split conformal prediction* (oder auch als inductive conformal prediction) bezeichnet und wurde erstmalig von *Papadopoulos u. a. (2002)* erwähnt. Wir orientieren uns bei dieser Einführung an Algorithm 2 von *Chernozhukov u. a. (2021)* und Algorithm 2 von *Lei u. a. (2018)*. Dafür teilen wir den Datensatz $\mathcal{D} = \{z_1, \dots, z_{n-1}\}$ in zwei Datensätze $\mathcal{D}_1 = \{z_1 \dots, z_k\}$ und $\mathcal{D}_2 = \{z_{k+1}, \dots, z_{n-1}\}$ auf (wie bisher können Elemente in diesen Mengen mehrfach vorkommen). Da die Daten, nach Annahme, austauschbar verteilt sein sollten, spielt es keine Rolle welche Daten wir für die Regressionsfunktion verwenden. Nun können wir die Regressionsfunktion $\hat{f}_0$ basierend auf $\mathcal{D}_1$ bestimmen.

Beispiel 2.11. Für die lineare Regression wäre die Regressionsfunktion dann

$$\hat{f}_0(x) = x^\top \hat{\beta}^{KQ}_{\mathcal{D}_1} = x^\top (\mathbb{X}_k^\top \mathbb{X}_k)^{-1} \mathbb{X}_k^\top \mathbf{Y},$$

wobei $\mathbb{X}_k = (x_1, \dots, x_k)^\top$ und $\mathbf{Y} = (y_1, \dots, y_k)^\top$ nur auf $\mathcal{D}_1$ basieren und nicht verändert werden.

Für den Datensatz $\mathcal{D}_2$ berechnen wir nun die nonconformity scores (für $R_i = |y_i - \hat{y}_i|$) mit der festen Funktion $\hat{f}_0$, welche nur durch $\mathcal{D}_1$ bestimmt ist,

$$R_{k+j} = |y_{k+j} - \hat{f}_0(x_{k+j})|, \; j = 1, \dots, n - (k+1).$$

Mit den nonconformity scores $R_{k+1}, \dots, R_{n-1}$ können wir nun das empirische $(1-\alpha)$-Quantil dieser Werte bestimmen, welches wir hier mit $q_{1-\alpha}$ bezeichnen.

Bemerkung 2.12. Für eine nach Größe geordnete Stichprobe $x_1, \dots, x_n$ ist das empirische p-Quantil, für $p \in (0,1)$ definiert durch

$$\hat{q}_p := \begin{cases} \frac{1}{2}(x_{np} + x_{np+1}) & \text{wenn } np \in \mathbb{N} \\ x_{\lceil np \rceil} & \text{wenn } np \notin \mathbb{N} \end{cases}.$$

Dann ist $q_{1-\alpha}$ größer oder gleich als $\lceil (n-k)(1-\alpha) \rceil$ der nonconformity scores $R_{k+1}, \dots, R_n$. Wenn nun $z_{k+1}, \dots, z_n$ austauschbar verteilt sind, gilt unabhängig von gegebenen $z_1, \dots, z_k$,

$$\mathbb{P}(R_n \leq q_{1-\alpha}) \geq 1 - \alpha,$$

nach Definition des Quantils (siehe auch Definition 3.2) und im Fall von $R_n = |y_n - \hat{f}_0(x_n)|$ folgt

$$\begin{aligned} R_n \leq q_{1-\alpha} &\iff |y_n - \hat{f}_0(x_n)| \leq q_{1-\alpha} \\ &\iff -q_{1-\alpha} \leq y_n - \hat{f}_0(x_n) \leq q_{1-\alpha} \\ &\iff \hat{f}_0(x_n) - q_{1-\alpha} \leq y_n \leq \hat{f}_0(x_n) + q_{1-\alpha}. \end{aligned}$$

Das Konfidenzintervall für y_n zum Niveau $1 - \alpha$ ist damit

$$[\hat{f}_0(x_n) \pm q_{1-\alpha}]. \tag{2.10}$$

Für eine genauere Betrachtung siehe Abschnitt 3.4. Dort wird in Proposition 3.17 formal bewiesen, dass (2.10) ein valider Vorhersagebereich ist.

Der Vorteil dieses Intervalls ist offensichtlich. Wir können nun sehr einfach einen Vorhersagebereich für jedes weitere y_{n+j} mit $j \in \mathbb{N}$ bestimmen, indem wir nur $\hat{f}_0(x_{n+j})$ neu berechnen, wobei $\hat{f}_0$ bereits gegeben ist und wir weiterhin das Quantil $q_{1-\alpha}$ der $R_{k+1}, \ldots, R_{n-1}$ nutzen. Außerdem müssen wir nur die Austauschbarkeit der Daten und die Unabhängigkeit der beiden aufgeteilten Datensätze voraussetzen und keine weiteren Annahmen an die genaue Verteilung der Daten stellen. Der Nachteil diese Verfahrens ist, dass wir die neuen Daten nicht nutzen, um möglicherweise genauere Ergebnisse, durch einen größeren Datensatz zu erhalten.
Im nächsten Kapitel werden wir weiter mit Quantilen arbeiten und diese für einen neuen Ansatz nutzen, um den p-Wert zu ersetzen. Außerdem werden wir, am Ende des Kapitels, wieder auf den split conformal predictor zu sprechen kommen, allerdings im Zusammenhang mit covariate shift.

3 Gewichtete conformal prediction

Bemerkung 3.1. Zur einfacheren Notation schreiben wir in diesem Kapitel für den Vorhersagebereich

$$\Gamma^\alpha(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n) := \Gamma^\alpha_n(x_n).$$

3.1 Conformal prediction über Quantile

Wir wollen nun einen anderen Zugang zu conformal prediction betrachten. Dieser ist durchaus ähnlich zu der Variante in Kapitel 1, allerdings wollen wir nun mit Quantilen arbeiten und nutzen dafür die allgemeine Definition des Quantils.

Definition 3.2. Das β Quantil der Verteilung F, für $Z \sim F$, ist definiert als

$$\text{Quantil}(\beta; F) = \inf\{z \in \mathbb{R} : \mathbb{P}(Z \leq z) \geq \beta\}.$$

Bemerkung 3.3. Im folgenden werden wir erlauben, dass Verteilungsfunktionen F auf $\mathbb{R} \cup \{\infty\}$ definiert sind. Außerdem schreiben wir für Zufallsvariablen $R_1, \ldots, R_n$ als Abkürzung $R_{1:n} = \{R_1, \ldots, R_n\}$, wobei dies die Menge der ungeordneten R_i ist. Wir fassen $R_{1:n}$ dabei nicht als klassische Menge auf, da es Werte auch doppelt enthalten soll, wenn $R_i = R_j$ für $i \neq j$ gilt. Für das Quantil der empirischen Verteilung der Zufallsvariablen $R_1, \ldots, R_n$ schreiben wir

$$\text{Quantil}(\beta; R_{1:n}) = \text{Quantil}\left(\beta; \frac{1}{n}\sum_{i=1}^{n} \delta_{R_i}\right),$$

wobei δ_{R_i} die Punktmasse von R_i bezeichnet (also die Dirac-Verteilung). Wir werden in diesem Zusammenhang den Ausdruck $R_{1:n-1} \cup \{\infty\}$ nutzen, welcher die Verteilung

$$\frac{1}{n}\sum_{i=1}^{n-1} \delta_{R_i} + \frac{1}{n}\delta_\infty$$

bezeichnet.
Im Fall von conformal prediction werden für die Zufallsvariablen $R_1, \ldots, R_n$, wieder die nonconformity scores verwendet.

Zum besseren Verständnis werden wir direkt beweisen, dass wir mithilfe von Quantilen einen Vorhersagebereich erhalten können, der konservativ valide ist. Wie sich allerdings herausstellen wird, ist dieser äquivalent zu der Menge (2.2), welchen wir mittels des p-Wertes ermittelt haben. Außerdem werden wir die Beweisstruktur der folgenden Beweise nutzen, um ähnlich dazu einen gewichteten conformal predictor zu ermitteln und zu beweisen. Dafür ist das folgende Lemma ein zentraler Bestandteil.

Lemma 3.4. *(Tibshirani u. a. 2019, Lemma 1) Seien $R_1, \ldots, R_n$ austauschbare Zufallsvariablen. Dann gilt für jedes $\beta \in (0, 1)$, dass*

$$\mathbb{P}\big(R_n \leq \textit{Quantil}(\beta; R_{1:n-1} \cup \{\infty\})\big) \geq \beta.$$

Beweis. Für den Beweis benötigen wir zuerst eine nützliche Eigenschaft von Quantilen von diskreten Verteilungsfunktionen F mit Träger $a_1, \ldots, a_n \in \mathbb{R}$. Sei $q = \textit{Quantile}(\beta; F)$, dann können wir alle $a_i > q$ austauschen durch Werte, die ebenfalls strikt größer als q sind, und

erhalten eine neue Verteilungsfunktion $\tilde{F}$. Für das β Quantil dieser neuen Funktion gilt dann

$$\text{Quantil}(\beta; \tilde{F}) = \text{Quantil}(\beta; F).$$

Daraus folgt

$$R_n > \text{Quantil}(\beta; R_{1:n-1} \cup \{\infty\}) \iff R_n > \text{Quantil}(\beta; R_{1:n}),$$

da wir wegen $\infty > R_n > \text{Quantil}(\beta; R_{1:n})$, den Wert R_n durch ∞ ersetzen können. Die Äquivalenz können wir nun einfach umschreiben zu

$$R_n \leq \text{Quantil}(\beta; R_{1:n-1} \cup \{\infty\}) \iff R_n \leq \text{Quantil}(\beta; R_{1:n}). \tag{3.1}$$

Nach der Definition des Quantils ist $\text{Quantil}(\beta; R_{1:n})$ größer als $\lceil \beta n \rceil$ der R_i, $i = 1, \ldots, n$. Damit gilt

$$R_n \leq \text{Quantil}(\beta; R_{1:n}) \iff R_n \text{ ist unter den } \lceil \beta n \rceil \text{ kleinsten Werten von } R_1, \ldots, R_n. \tag{3.2}$$

Wegen der Austauschbarkeit der $R_1, \ldots, R_n$ folgt für die Wahrscheinlichkeit

$$\mathbb{P}(R_n \text{ ist unter den } \lceil \beta n \rceil \text{ kleinsten } R_1, \ldots, R_n) \geq \frac{\lceil \beta n \rceil}{n} \geq \beta$$

und damit folgt die Behauptung. □

Mit diesem Lemma können wir nun beweisen, dass die Menge der $y_n \in \mathbb{R}$, welche

$$R_n \leq \text{Quantil}(1 - \alpha; R_{1:n-1} \cup \{\infty\})$$

erfüllen, konservativ valide ist. Damit haben wir eine Möglichkeit gefunden conformal prediction über Quantile einzuführen.

Satz 3.5. *(Tibshirani u. a. 2019, Theorem 1) Angenommen $Z_i = (X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$ sind austauschbar verteilt. Für $\alpha \in (0, 1)$ und nonconformity scores R_i, $i = 1, \ldots, n$, wie in (1.3) ist der Vorhersagebereich, für $x_n \in \mathbb{R}^d$, definiert als*

$$\hat{\Gamma}_n^\alpha(x_n) = \big\{ y_n \in \mathbb{R} : R_n \leq \textit{Quantil}(1 - \alpha; R_{1:n-1} \cup \{\infty\}) \big\}. \tag{3.3}$$

Dann erfüllt $\hat{\Gamma}_n^\alpha$ die Bedingung

$$\mathbb{P}\left(Y_n \in \hat{\Gamma}_n^\alpha(X_n) \right) \geq 1 - \alpha$$

Beweis. Es ist klar, dass gilt

$$\mathbb{P}\left(Y_n \in \hat{\Gamma}_n^\alpha(X_n) \right) = \mathbb{P}\big(R_n \leq \text{Quantil}(1 - \alpha; R_{1:n-1} \cup \{\infty\}) \big).$$

Da $Z_1, \ldots, Z_n$ austauschbar sind, folgt mit Lemma 1.11, dass die nonconformity scores $R_1, \ldots, R_n$ ebenfalls austauschbar sind. Nun können wir Lemma 3.4 anwenden und es folgt direkt die Behauptung. □

Dieser Satz erscheint sehr ähnlich zu Satz 1.9, welcher unsere erste wichtige Aussage über conformal prediction darstellt. Aus diesem Grund interessiert es uns, inwiefern diese Sätze

zusammenhängen.

Bemerkung 3.6. Zur Vereinfachung nehmen wir wieder an, dass $R_1, \ldots, R_n$ fast sicher unterschiedlich sind.

Fall 1: $\alpha n \in \mathbb{N}$.
Mit (3.1) und (3.2) folgt die Äquivalenz

$$R_n \leq \text{Quantil}(1-\alpha; R_{1:n-1} \cup \{\infty\}) \iff R_n \text{ ist unter den } (1-\alpha)n \text{ kleinsten } R_1, \ldots, R_n.$$

Für den in (2.2) definierten Vorhersagebereich,

$$\Gamma_n^\alpha(x_n) = \{y_n \in \mathbb{R} : p_{(x_n,y_n)} > \alpha\}$$

mit

$$p_{(x_n,y_n)} = \frac{|\{i = 1, \ldots, n : R_i \geq R_n\}|}{n}$$

gelten dann die Äquivalenzen

$$\begin{aligned} p_{(x_n,y_n)} > \alpha &\iff p_{(x_n,y_n)} > \frac{\alpha n}{n} \\ &\iff |\{i = 1, \ldots, n : R_i \geq R_n\}| > \alpha n \\ &\iff \text{Es gibt mindestens } \alpha n + 1 \text{ der } R_i s, \text{ die größer oder gleich sind als } R_n \\ &\iff R_n \text{ ist unter den } (1-\alpha)n \text{ kleinsten } R_1, \ldots, R_n. \end{aligned}$$

Und somit folgt die Gleichheit der beiden Mengen

$$\big\{y_n \in \mathbb{R} : p_{(x_n,y_n)} > \alpha\big\} = \big\{y_n \in \mathbb{R} : R_n \leq \text{Quantil}(1-\alpha; R_{1:n-1} \cup \{\infty\})\big\}.$$

Fall 2: $\alpha n \notin \mathbb{N}$.
Wir werden analog zum 1. Fall vorgehen, wobei wir nun Gaußklammern verwenden müssen. Daher gelten die Äquivalenzen

$$R_n \leq \text{Quantil}(1-\alpha; R_{1:n-1} \cup \{\infty\}) \iff R_n \text{ ist unter den } \lceil (1-\alpha)n \rceil \text{ kleinsten } R_1, \ldots, R_n$$

und

$$\begin{aligned} p_{(x_n,y_n)} > \alpha &\iff p_{(x_n,y_n)} > \frac{\alpha n}{n} \\ &\iff |\{i = 1, \ldots, n : R_i \geq R_n\}| > \alpha n \\ &\iff \text{Es gibt mindestens } \lfloor \alpha n \rfloor + 1 \text{ der } R_i s, \text{ die größer oder gleich sind als } R_n \\ &\iff R_n \text{ ist unter den } \lfloor (1-\alpha)n \rfloor + 1 \text{ kleinsten } R_1, \ldots, R_n. \end{aligned}$$

Wobei die letzte Äquivalenz sich aus folgender Gleichung ergibt

$$n - \lfloor \alpha n \rfloor = n + \lfloor -\alpha n \rfloor + 1 = \lfloor (1-\alpha)n \rfloor + 1.$$

Wegen $\lfloor (1-\alpha)n \rfloor + 1 = \lceil (1-\alpha)n \rceil$ folgt dann ebenfalls die Gleichheit der Mengen.
Somit ist Satz 3.5 eine andere Variante von Satz 1.9.

3.2 Gewichteter conformal predictor

Nun wollen wir einen gewichteten conformal predictor konstruieren. Dieser soll ähnlich zu Satz 3.5 sein, weshalb wir für den Beweis eine gewichtete Version von Lemma 3.4 benötigen. Um dies zu erreichen, präsentieren wir zunächst einen alternativen Beweis von Lemma 3.4. Diese Beweisstruktur wird uns helfen, mit der selben Strategie, die gewichtete Version des Lemmas zu zeigen.

Alternativer Beweis von Lemma 3.4. (*vgl. Tibshirani u. a. 2019*): Zur Vereinfachung nehmen wir an, dass die Zufallsvariablen $R_1, \dots, R_n$ fast sicher verschieden sind. Sei E_r das Ereignis $\{R_1, \dots, R_n\} = \{r_1, \dots, r_n\}$. Damit ist gemeint, dass wir alle möglichen Realisierungen der Zufallsvariable R_i kennen, aber nicht wissen welche konkret für R_i zutrifft. Außerdem sei f die Dichtefunktion der gemeinsamen Verteilung von $R_1, \dots, R_n$. Wegen der Austauschbarkeit gilt dann für jede Permutation π über $1, \dots, n$

$$f(r_1, \dots, r_n) = f(r_{\pi(1)}, \dots, r_{\pi(n)}).$$

Für jedes $i = 1, \dots, n$ gilt für die bedingte Wahrscheinlichkeit

$$\begin{aligned}
\mathbb{P}\left(R_n = r_i | E_r\right) &= \frac{\sum_{\{\pi:\pi(n)=i\}} f(r_{\pi(1)}, \dots, r_{\pi(n)})}{\sum_{\pi} f(r_{\pi(1)}, \dots, r_{\pi(n)})} \\
&= \frac{\sum_{\{\pi:\pi(n)=i\}} f(r_1, \dots, r_n)}{\sum_{\pi} f(r_1, \dots, r_n)} \\
&= \frac{(n-1)!}{n!} \\
&= \frac{1}{n}
\end{aligned}$$

Somit ist $R_n | E_r$ uniform auf $\{r_1, \dots, r_n\}$ verteilt, also

$$R_n | E_r \sim \frac{1}{n} \sum_{i=1}^{n} \delta_{r_i},$$

und es folgt mit der Definition 3.2 des Quantils

$$\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \frac{1}{n} \sum_{i=1}^{n} \delta_{r_i}\right) \middle| E_r \right) \geq \beta.$$

Das Ereignis E_r besagt $\{R_1, \dots, R_n\} = \{r_1, \dots, r_n\}$, daher gilt auch

$$\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \frac{1}{n} \sum_{i=1}^{n} \delta_{R_i}\right) \middle| E_r \right) \geq \beta.$$

Da dies wahr ist, für alle r, können wir marginalisieren und erhalten

$$\begin{aligned}
\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \frac{1}{n} \sum_{i=1}^{n} \delta_{R_i}\right)\right) &= \mathbb{E}\left[\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \frac{1}{n} \sum_{i=1}^{n} \delta_{R_i}\right) \middle| E_r \right)\right] \\
&\geq \mathbb{E}[\beta] \\
&= \beta.
\end{aligned}$$

Wegen (3.1) ist das äquivalent zu der Behauptung im Lemma. □

Wie auch in diesem Lemma ist Austauschbarkeit die wichtigste Voraussetzung für conformal prediction. Diese Voraussetzung müssen wir im gewichteten Fall weiter verallgemeinern. Das ist auch der Grund warum wir den Beweis von Lemma 3.4 auf andere Weise aufgeschrieben haben. In diesem alternativen Beweis nutzen wir die Austauschbarkeit nur bei der Dichte Funktion, um die bedingte Wahrscheinlichkeit zu bestimmen. Dieses Vorgehen wollen wir genau so wieder nutzen, mit der sogenannten *gewichteten Austauschbarkeit*.

Definition 3.7. *(Tibshirani u. a. 2019, Definition 1)* Zufallsvariablen $Z_1, \ldots, Z_n$ sind *gewichtet austauschbar* mit Gewichtsfunktionen $w_1, \ldots, w_n$, wenn die Dichte f der Verteilung geschrieben werden kann als

$$f(z_1, \ldots, z_n) = \prod_{i=1}^{n} w_i(z_i) \cdot g(z_1, \ldots, z_n),$$

wobei g eine Funktion mit der Eigenschaft

$$g(z_1, \ldots, z_n) = g(z_{\pi(1)}, \ldots, z_{\pi(n)})$$

ist, für jede Permutation π auf $1, \ldots, n$. Die Dichte f sei die Radon-Nikodym Ableitung bezüglich eines beliebigen Basis-Maßes.

Ein Beispiel, für gewichtet austauschbar verteilte Zufallsvariablen, ist durch das folgende Lemma gegeben. Dieses wird im Fall von covariate shift nochmals aufgreifen.

Lemma 3.8. *(Tibshirani u. a. 2019, Lemma 2) Seien $Z_i \sim P_i$, $i = 1, \ldots, n$ unabhängige Zufallsvariablen, wobei P_i, für $i \geq 2$, absolut stetig bezüglich P_1 ist. Dann sind $Z_1, \ldots, Z_n$ gewichtet austauschbar verteilt mit Gewichtsfunktionen $w_1 \equiv 1$ und $w_i = dP_i/dP_1$, $i \geq 2$.*

Beweis. Die Unabhängigkeit der Z_i impliziert

$$f(z_1, \ldots, z_n) = \prod_{i=1}^{n} f_i(z_i),$$

wobei f_i die Dichte von P_i, bzgl. eines Maßes μ, ist. Wegen $P_i \ll P_1$, $i \geq 2$, existieren die Radon-Nikodym-Dichten

$$w_i(z) = \frac{dP_i}{dP_1}(z), \ i \geq 2.$$

Da außerdem $w_1 \equiv 1$, gilt für alle $i = 1, \ldots, n$

$$f_i(z) = w_i(z) \cdot f_1(z)$$

und wir können die gemeinsame Dichte schreiben als

$$f(z_1, \ldots, z_n) = \prod_{i=1}^{n} w_i(z_i) \cdot f_1(z_i) = \prod_{i=1}^{n} w_i(z_i) \cdot \prod_{i=1}^{n} f_1(z_i).$$

Um Definition 3.7 zu erfüllen, setzen wir also

$$g(z_1, \ldots, z_n) = \prod_{i=1}^{n} f_1(z_i),$$

wobei diese Funktion, aufgrund der Multiplikation, offensichtlich die Symmetrieeigenschaft von g erfüllt. □

Bemerkung 3.9. Die bisherige Definition von Austauschbarkeit nach Definition 6.1 entspricht gerade der gewichteten Austauschbarkeit, wenn für alle $i = 1, \dots, n$ die Gewichtsfunktionen $w_i \equiv 1$ erfüllen.

Nun kommen wir allerdings zu der gewichteten Version von Lemma 3.4, welche das zentrale Element im Beweis für einen gewichteten conformal predictor ist.

Lemma 3.10. *(Tibshirani u. a. 2019, Lemma 3) Seien $Z_1, \dots, Z_n$ gewichtet austauschbare Zufallsvariablen mit Gewichtsfunktionen $w_1, \dots, w_n$. Sei $R_i = A((Z_1, \dots, Z_n)^\top, Z_i)$, für $i = 1, \dots, n$, und A ein beliebiges nonconformity measure. Definiere*

$$p_i^w(z_1, \dots, z_n) = \frac{\sum_{\{\pi:\pi(n)=i\}} \prod_{j=1}^n w_j(z_{\pi(j)})}{\sum_\pi \prod_{j=1}^n w_j(z_{\pi(j)})}, i = 1, \dots, n, \tag{3.4}$$

wobei über die Permutationen π, der Zahlen $1, \dots, n$, summiert wird. Dann gilt für jedes $\beta \in (0, 1)$

$$\mathbb{P}\left(R_n \leq \mathit{Quantil}\left(\beta; \sum_{i=1}^{n-1} p_i^w(Z_1, \dots, Z_n)\delta_{R_i} + p_n^w(Z_1, \dots, Z_n)\delta_\infty\right)\right) \geq \beta.$$

Beweis. Zur Vereinfachung nehmen wir an, dass die nonconformity scores $R_1, \dots, R_n$ fast sicher eindeutig sind. Sei E_z das Ereignis $\{Z_1, \dots, Z_n\} = \{z_1, \dots, z_n\}$ und $r_i = A(z_{1:n}, z_i)$. Außerdem sei f die Dichtefunktion der gemeinsamen Verteilung der $Z_1, \dots, Z_n$. Dann gilt für die bedingte Wahrscheinlichkeit aller $i = 1, \dots, n$,

$$\mathbb{P}(R_n = r_i | E_z) = \mathbb{P}(Z_n = z_i | E_z) = \frac{\sum_{\{\pi:\pi(n)=i\}} f(z_{\pi(1)}, \dots, z_{\pi(n)})}{\sum_\pi f(z_{\pi(1)}, \dots, z_{\pi(n)})}.$$

Wegen der gewichteten Austauschbarkeit von $Z_1, \dots, Z_n$ folgt

$$\begin{aligned}
\frac{\sum_{\{\pi:\pi(n)=i\}} f(z_{\pi(1)}, \dots, z_{\pi(n)})}{\sum_\pi f(z_{\pi(1)}, \dots, f_{\pi(n)})} &= \frac{\sum_{\{\pi:\pi(n)=1\}} \prod_{j=1}^n w_j(z_{\pi(j)}) \cdot g(z_{\pi(1)}, \dots, z_{\pi(n)})}{\sum_\pi \prod_{j=1}^n w_j(z_{\pi(j)}) \cdot g(z_{\pi(1)}, \dots, z_{\pi(n)})} \\
&= \frac{\sum_{\{\pi:\pi(n)=1\}} \prod_{j=1}^n w_j(z_{\pi(j)}) \cdot g(z_1, \dots, z_n)}{\sum_\pi \prod_{j=1}^n w_j(z_{\pi(j)}) \cdot g(z_1, \dots, z_n)} \\
&= \frac{\sum_{\{\pi:\pi(n)=1\}} \prod_{j=1}^n w_j(z_{\pi(j)})}{\sum_\pi \prod_{j=1}^n w_j(z_{\pi(j)})} \\
&= p_i^w(z_1, \dots, z_n).
\end{aligned} \tag{3.5}$$

Wir können also über die bedingte Verteilung sagen

$$R_n | E_z \sim \sum_{i=1}^n p_i^w(z_1, \dots, z_n)\delta_{r_i}.$$

Diese Verteilung impliziert die Aussage

$$\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \sum_{i=1}^n p_i^w(z_1, \dots, z_n)\delta_{r_i}\right) \middle| E_z\right) \geq \beta,$$

und weil das Ereignis E_z besagt, dass $\{Z_1, \dots, Z_n\} = \{z_1, \dots, z_n\}$, folgt auch $R_i = r_i$, und

obiges ist äquivalent zu

$$\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \sum_{i=1}^{n} p_i^w(Z_1,\ldots,Z_n)\delta_{R_i}\right)\middle| E_z\right) \geq \beta.$$

Da das vorherige Argument für alle z wahr ist, und somit auch für alle r, können wir marginalisieren und erhalten

$$\mathbb{P}\left(R_n \leq \text{Quantil}\left(\beta; \sum_{i=1}^{n} p_i^w(Z_1,\ldots,Z_n)\delta_{R_i}\right)\right) \geq \beta.$$

Dies ist äquivalent zu der Aussage, die wir zeigen wollten, da nach (3.1):

$$R_n \leq \text{Quantil}(\beta; R_{1:(n-1)} \cup \{\infty\}) \Leftrightarrow R_n \leq \text{Quantil}(\beta; R_{1:n}).$$

□

Bemerkung 3.11. Der Unterschied zum alternativen Beweis von Lemma 3.4 ist, dass wir durch die Definition der gewichteten Austauschbarkeit in der Gleichung (3.5) nur noch das Verhältnis der Gewichtsfunktionen betrachten und damit für jedes r_i eine individuelle Gewichtung $p_i^w(z_1,\ldots,z_n)$ erhalten. Wenn nur Austauschbarkeit gegeben ist ergibt sich bei dieser Rechnung, allgemein für alle r_i, das Gewicht $1/n$, da die Gewichte dann immer $w_i \equiv 1$ erfüllen und es somit nur auf die Anzahl der Summanden ankommt.

Mit Hilfe dieses Lemmas können wir nun eine allgemeine Aussage für gewichtet austauschbare Zufallsvariablen treffen und erhalten damit einen gewichteten Vorhersagebereich.

Satz 3.12. *(Tibshirani u. a. 2019, Theorem 2) Angenommen $Z_i = (X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}, i = 1,\ldots,n$ ist gewichtet austauschbar mit Gewichtsfunktion $w_1,\ldots,w_n$. Für jedes nonconformity measure A, und jedes $\alpha \in (0,1)$, sei der gewichtete Conformal Predictor an einem Punkt $x_n \in \mathbb{R}^d$ definiert durch*

$$\Gamma_n^\alpha(x_n) = \left\{ y \in \mathbb{R} : R_n \leq \mathit{Quantil}\left(1-\alpha; \sum_{i=1}^{n-1} p_i^w(Z_1,\ldots,Z_{n-1},(x_n,y_n))\delta_{R_i} + p_n^w(Z_1,\ldots,Z_{n-1},(x_n,y_n))\delta_\infty\right)\right\},$$

wobei R_i, $i = 1,\ldots,n$ wie in Definition 1.3 (ggf. ohne i-ten Wert) und $p_i^w, i = 1,\ldots,n$ wie in (3.4)*. Dann erfüllt Γ_n^α*

$$\mathbb{P}\big(Y_n \in \Gamma_n^\alpha(X_n)\big) \geq 1-\alpha$$

Beweis. Es gilt die Äquivalenz

$$Y_n \in \Gamma_n^\alpha(X_n) \Leftrightarrow R_n \leq \text{Quantil}\left(1-\alpha; \sum_{i=1}^{n-1} p_i^w(Z_1,\ldots,Z_n)\delta_{R_i} + p_n^w(Z_1,\ldots,Z_n)\delta_\infty\right),$$

wegen der Definition des Vorhersagebereiches $\Gamma_n^\alpha(x_n)$. Mit Lemma 3.10 folgt dann direkt die Aussage. □

Wie *Hore und Barber (2025) (Appendix B)* betonen, ist es für die gewichtete Variante ebenfalls möglich einen geglätteten conformal predictor zu erhalten, wie es in Satz 1.18 der Fall ist, und kann über eine Randomisierung des Quantils erreicht werden. Dies ist für den gewichteten Fall aus dem Grund deutlich relevanter, da die Wahrscheinlichkeit des ungewichteten conformal predictors durch $1-\alpha+1/n$ nach oben beschränkt ist (siehe

Proposition 1.20), während beim gewichteten conformal predictor die obere Schranke durch das größte Gewicht bestimmt ist, welches deutlich größer seien kann, als $1/n$.

3.3 Covariate shift

In diesem Kapitel wollen wir uns mit dem Fall auseinandersetzen, wenn die Kovariablen der Trainingsdaten und der Testdaten unterschiedlichen Verteilungen folgen. Dieses Ereignis tritt in der Realität öfter auf und wird als *covariate shift* bezeichnet. Mathematisch formulieren Tibshirani u. a. (2019) dies wie folgt.

Definition 3.13. Wir sprechen von covariate shift, im Bezug auf die Daten $(X_i, Y_i), i = 1, \ldots, n$, wenn die Trainings- und Testdaten wie folgt verteilt sind:

$$\begin{aligned} (X_i, Y_i) \overset{u.i.v.}{\sim} P = P_X \times P_{Y|X},\ i = 1, \ldots, n-1, \\ (X_n, Y_n) \sim \tilde{P} = \tilde{P}_X \times P_{Y|X},\ \text{unabhängig.} \end{aligned} \tag{3.6}$$

Bemerkung 3.14. Es ist wichtig zu beachten, dass die Daten nicht mehr allgemein austauschbar verteilt sind, wie es bisher meistens vorausgesetzt wurde. Das ist bereits daran zu erkennen, dass (X_n, Y_n) nicht identisch zu den anderen (X_i, Y_i) verteilt ist. Allerdings genügt es, dass die Daten unabhängig verteilt sind, da dies bereits gewichtete Austauschbarkeit impliziert. Daher benötigen wir diese allgemeinere Version der Austauschbarkeit, um überhaupt mit Daten der Form (3.6) arbeiten zu können. Wir werden sehen, dass Lemma 3.8 die gewichtete Austauschbarkeit dieser Daten garantiert.

Die einzige weitere Voraussetzung, die wir benötigen, um einen Vorhersagebereich zu konstruieren, ist die Likelihood-Ratio

$$\frac{d\tilde{P}_X}{dP_X}.$$

Damit können wir die nonconformity scores R_i, proportional zur Likelihood-Ratio der Verteilungen unserer Trainingsdaten P_X und unserer Testdaten $\tilde{P}_X$, gewichten. Es gilt dann für die Gewichtsfunktion w von X_i

$$w(X_i) = \frac{d\tilde{P}_X(X_i)}{dP_X(X_i)}.$$

Wir werden im folgenden die gewichtete empirische Verteilung

$$\sum_{i=1}^{n-1} p_i^w(x)\delta_{R_i} + p_n^w(x)\delta_\infty$$

betrachten, wobei die Gewichte p_i^w für $i = 1, \ldots, n-1$ und p_n^w sich als

$$\begin{aligned} p_i^w(x) &= \frac{w(X_i)}{\sum_{j=1}^{n-1} w(X_j) + w(x)}, \\ p_n^w(x) &= \frac{w(x)}{\sum_{j=1}^{n-1} w(X_j) + w(x)}, \end{aligned} \tag{3.7}$$

ergeben, wie aus dem Beweis des folgenden Korollars klar wird. Durch diese Gewichtung können wir nun einen Vorhersagebereich unter covariate shift bestimmen, als Spezialfall des gewichteten Vorhersagebereiches nach Satz 3.12.

Korollar 3.15. *(Tibshirani u. a. 2019, Corollary 1) Seien Daten gegeben, wie in* (3.6)*. Angenommen $\tilde{P}_X$ ist absolut stetig bezüglich P_X und $w = d\tilde{P}_X/dP_X$. Für jedes nonconformity*

measure A und jedes $\alpha \in (0,1)$, definiere für $x_n \in \mathbb{R}^d$

$$\Gamma_n^\alpha(x_n) = \left\{ y_n \in \mathbb{R} : R_n \leq \mathit{Quantil}\left(1-\alpha; \sum_{i=1}^{n-1} p_i^w(x_n)\delta_{R_i} + p_n^w(x_n)\delta_\infty\right)\right\},$$

wobei $R_i = A((z_1,\dots,z_n)^\top, z_i)$, $i = 1,\dots,n$, mit $z_i = (x_i, y_i)$, und $p_i^w(x)$, $i = 1,\dots,n$ wie in (3.7). Dann ist der Vorhersagebereich $\Gamma_n^\alpha(x_n)$ konservativ valide, also

$$\mathbb{P}\big(y_n \in \Gamma_n^\alpha(x_n)\big) \geq 1-\alpha.$$

Beweis. Mit Lemma 3.8 folgt, dass die $Z_i = (X_i, Y_i)$ aus (3.6) gewichtet austauschbar sind mit $w_i \equiv 1$ für $i = 1,\dots,n-1$ und $w_n((x_n,y_n)) = w(x_n)$. Dies gilt, da Lemma 3.8, mit (3.6) für $i = 1$, $w_1 \equiv 1$ und für $i = 2,\dots,n-1$,

$$w_i = \frac{dP_i}{dP_1} = \frac{d(P_X \times P_{Y|X})}{d(P_X \times P_{Y|X})} \equiv 1$$

impliziert. Genauso gilt in diesem Fall für w_n

$$w_n = \frac{d(\tilde{P}_X \times P_{Y|X})}{d(P_X \times P_{Y|X})} = \frac{d\tilde{P}_X}{dP_X} = w.$$

Damit vereinfacht sich der Ausdruck (3.4) zu

$$p_i^w(z_1,\dots,z_n) = \frac{\sum_{\{\pi:\pi(n)=i\}} w(x_i)}{\sum_\pi w(x_{\pi(n)})} = \frac{w(x_i)}{\sum_{j=1}^n w(x_j)},\ i = 1,\dots,n.$$

Wir können also folgern, dass

$$p_i^w(Z_1,\dots,Z_{n-1},(x_n,y_n)) = p_i^w(x),$$

für $i = 1,\dots,n$, wobei $p_i^w(x_n)$ wie in (3.7). Durch Anwendung von Satz 3.12 folgt dann die Aussage des Korollars. □

3.4 Split conformal prediction unter covariate shift

Wir betrachten nun erneut den split conformal predictor, den wir bereits in Abschnitt 2.3 eingeführt haben.
Sei $\mathcal{D} = \{Z_1,\dots,Z_{n-1}\}$ ein Datensatz aus $n-1$ Beobachtungen. Wir teilen diesen in Trainingsmenge $\mathcal{D}_1 = \{Z_1,\dots,Z_k\}$ und Kalibrierungsmenge $\mathcal{D}_2 = \{Z_{k+1},\dots,Z_{n-1}\}$. Gegeben X_n wollen wir die zugehörige Kategorie Y_n vorhersagen. Der split conformal predictor liefert dazu den Vorhersagebereich *(vgl. Chernozhukov u. a. 2021, Algorithm 2 und Fontana u. a. 2023)*

$$\Gamma_{n-k}^\alpha(X_n) = \{Y_n \in \mathbb{R} : R_n \leq \text{Quantil}(1-\alpha; R_{k+1:n-1} \cup \{\infty\})\}. \tag{3.8}$$

Bemerkung 3.16. Wir schreiben bei der Menge $\Gamma_{n-k}^\alpha(X_n)$ als Index $n-k$, da dieser Vorhersagebereich auf den $n-(k+1)$ Werten aus $\mathcal{D}_1$ und dem Wert X_n basiert und wir die Kategorie Y_n von X_n Vorhersagen wollen. Damit haben wir $n-k$ gegebene Werte, um den

Vorhersagebereich zu bestimmen. Also

$$\Gamma^{\alpha}_{n-k}(X_n) = \Gamma^{\alpha}(Z_{k+1}, \ldots, Z_{n-1}, X_n).$$

Wir wollen nun zeigen, dass der Vorhersagebereich (3.8) auch wirklich konservativ valide ist und damit auch eine nützliche Alternative zu unserem bisherigen Verfahren. Da diese Version von conformal prediction ein Spezialfall ist, müssen wir nur zeigen, dass die Voraussetzungen für den klassische conformal predictor erfüllt sind, um die Behauptung mit Satz 3.5 folgern zu können. Die folgenden Aussagen basieren wieder auf denen von *Tibshirani u. a. 2019, Kapitel 2.2*, allerdings wollen wir dies formaler und etwas ausführlicher darstellen. Zudem unterscheiden sich einige Notationen, aufgrund unterschiedlicher Schreibweisen und Definitionen.

Proposition 3.17. *Seien $\mathcal{D} = \{Z_1, \ldots, Z_{n-1}\}$ austauschbar verteilt, sowie eine weitere austauschbar verteilte Zufallsvariable Z_{n+m} für $m \geq 0$. Können wir $\mathcal{D}$ aufteilen, in $\mathcal{D}_1 = \{Z_1, \ldots, Z_k\}$ und $\mathcal{D}_2 = \{Z_{k+1}, \ldots, Z_{n-1}\}$, mit $k < n$, wobei $\mathcal{D}_1$ und $\mathcal{D}_2$ unabhängig sind, dann gilt, dass die Menge* (3.8) *die Bedingung*

$$\mathbb{P}\big(Y_{n+m} \in \Gamma^{\alpha}_{n-k}(X_{n+m})\big) \geq 1 - \alpha$$

für jedes $m \geq 0$ erfüllt.

Beweis. (vgl. Jin und Ren 2025, S. 6 und Tibshirani u. a. 2019, S. 5) Da $\hat{f}_0$ fixiert ist, durch $\mathcal{D}_1$, sind die nonconformity scores $R_i = |Y_i - \hat{f}_0(X_i)|$ für $i = k+1, \ldots, n-1, n+m$, bedingt auf $\mathcal{D}_1$, austauschbar verteilt. Dies gilt, weil $Z_{k+1}, \ldots, Z_{n-1}, Z_{n+m}$, bedingt auf $\mathcal{D}_1$, fast sicher austauschbar verteilt sind. Mit Satz 3.5 folgt daher

$$\mathbb{P}\big(Y_{n+m} \in \Gamma^{\alpha}_{n-k}(X_{n+m}) | \mathcal{D}_1\big) \geq 1 - \alpha.$$

Mit der Turmeigenschaft folgt dann

$$\mathbb{P}\big(Y_{n+m} \in \Gamma^{\alpha}_{n-k}(X_{n+m})\big) = \mathbb{E}_{\mathcal{D}_1}\big[\mathbb{P}\big(Y_{n+m} \in \Gamma^{\alpha}_{n-k}(X_{n+m}) | \mathcal{D}_1\big)\big] \geq 1 - \alpha.$$

□

Bemerkung 3.18. Die Gleichung in (3.8) ist allgemein gültig für beliebige nonconformity scores. Wählen wir jedoch $R_i = |Y_i - \hat{f}_0(X_i)|$ als nonconformity score, so können wir die Menge noch genauer angeben und direkt als Intervall schreiben

$$\begin{aligned}
\Gamma^{\alpha}_{n-k}(X_n) &= \{Y_n \in \mathbb{R} : R_n \leq \text{Quantil}(1-\alpha; R_{k+1:n-1} \cup \{\infty\})\} \\
&= \Big\{Y_n \in \mathbb{R} : |y - \hat{f}_0(X_n)| \leq \text{Quantil}(1-\alpha; R_{k+1:n-1} \cup \{\infty\})\Big\} \qquad (3.9) \\
&= [\hat{f}_0(X_n) \pm \text{Quantil}(1-\alpha; R_{k+1:n-1} \cup \{\infty\})].
\end{aligned}$$

Dies können wir nun auf jeden weiteren Punkt X_{n+m} mit $m \geq 1$ anwenden und nutzen dabei immer die feste Funktion $\hat{f}_0$, wodurch wir viel effizienter den entsprechenden Vorhersagebereich bestimmen können. Dieses Intervall entspricht dem Intervall (2.10) mit einer leichten Anpassung des Quantils $q_{1-\alpha}$.

Split conformal prediction ist also offensichtlich ein Spezialfall von unserer bisherigen conformal prediction Methode. Somit lässt sich nicht nur Satz 3.5 anwenden, sondern auch

Korollar 3.15 und als Vorhersageintervall, unter den Voraussetzungen von covariate shift (3.6), erhalten wir für nonconformity scores $R_i = |Y_i - \hat{f}_0(X_i)|$ mit Proposition 3.19

$$\begin{aligned}\Gamma^{\alpha}_{n-k}(X_n) &= \left\{ Y_n \in \mathbb{R} : R_n \leq \text{Quantil}\left(1-\alpha; \sum_{i=k+1}^{n-1} p_i^w(X_n)\delta_{R_i} + p_n^w(X_n)\delta_{\infty}\right)\right\} \\ &= \left[\hat{f}_0(X_n) \pm \text{Quantil}\left(1-\alpha; \sum_{i=k+1}^{n-1} p_i^w(X_n)\delta_{|Y_i - f_0(X_i)|} + p_n^w(X_n)\delta_{\infty}\right)\right],\end{aligned} \tag{3.10}$$

wobei die p_i für $i = k+1, \ldots, n$ wie in (3.7) definiert sind (*vgl.* Tibshirani u. a. 2019, *S. 5*).

Diese Aussage formulieren wir formal wie folgt:

Proposition 3.19. *Seien Daten unter covariate shift* (3.6) *gegeben, also* $\mathcal{D} = \{Z_1, \ldots, Z_{n-1}\}$ *mit* $Z_i \overset{u.i.v.}{\sim} P = P_X \times P_{Y|X}$ *für* $i = 1 \ldots, n-1$ *und eine weitere unabhängige Zufallsvariable* $Z_{n+m} \sim \tilde{P} = \tilde{P}_X \times P_{Y|X}$ *für* $m \geq 0$. *Angenommen* $\tilde{P}_X$ *ist absolut stetig bezüglich* P_X *und* $w = d\tilde{P}_X / dP_X$. *Für jedes nonconformity measure* A *und jedes* $\alpha \in (0,1)$, *ist der split conformal predictor für den Trainingsdatensatz* $\mathcal{D}_1 = \{Z_1, \ldots, Z_k\}$ *und den Kalibrierungsdatensatz* $\mathcal{D}_2 = \{Z_{k+1}, \ldots, Z_{n-1}\}$ *gegeben als*

$$\Gamma^{\alpha}_{n-k}(X_{n+m}) = \left\{ Y_{n+m} \in \mathbb{R} : R_{n+m} \leq \mathit{Quantil}\left(1-\alpha; \sum_{i=k+1}^{n-1} p_i^w(X_{n+m})\delta_{R_i} + p_{n+m}^w(X_{n+m})\delta_{\infty}\right)\right\}$$

und konservativ valide.

Beweis. (vgl. Tibshirani u. a. 2019, S. 5) Da $\hat{f}_0$ wieder durch $\mathcal{D}_1$ fixiert ist, erfüllen $\mathcal{D}_2$ und Z_{n+m} für $m \geq 0$, gegeben $\mathcal{D}_1$, die Voraussetzungen für Korollar 3.15. Es gilt also

$$\mathbb{P}\big(Y_{n+m} \in \Gamma^{\alpha}_{n-k}(X_{n+m}) | \mathcal{D}_1\big) \geq 1 - \alpha$$

und mit der Turmeigenschaft folgt analog zum Beweis von Proposition 3.17 die Behauptung. □

3.5 Verallgemeinerter covariate shift

In diesem Abschnitt wollen wir nun eine allgemeinere Variante von covariate shift betrachten. Diese ist eine Erweiterung von (3.6) und wird von *Wang und Qiao (2025)* für Klassifikation eingeführt. Wir wollen zeigen wie diese Variante im Fall von Regression angewendet werden kann.

Wir wollen also wieder einen Vorhersagebereich für den Wert Y_n bestimmen, für gegebene $Z_1, \ldots, Z_{n-1}, X_n$. Dabei seien diese Daten wie folgt verteilt:

$$\begin{aligned}(X_i, Y_i) &\overset{u.i.v.}{\sim} P = P_X \times P_{Y|X},\ i = 1, \ldots, l, \\ (X_j, Y_j) &\overset{u.i.v.}{\sim} \tilde{P} = \tilde{P}_X \times P_{Y|X},\ j = l+1, \ldots, n,\end{aligned} \tag{3.11}$$

wobei alle Daten unabhängig voneinander sind (wir schreiben auch $Z_i \sim P_i$). Wir haben also l Datenpunkte aus der Quellverteilung (Source) P und $n - l - 1$ Datenpunkte aus der Zielverteilung (Target) $\tilde{P}$ gegeben (in der Realität ist $n - l - 1$ meistens klein, da wir sonst direkt den ungewichteten conformal predictor nutzen könnten). Der Unterschied zu unserer bisherigen Definition von covariate shift ist, dass wir nun auch Daten, mit Kategorie, aus

der selben Verteilung wie (X_n, Y_n) haben und vorher nur die aus der Quellverteilung zur Verfügung hatten. Die obige Definition von covariate shift ist offensichtlich eine Verallgemeinerung von (3.6), da diese, für $l = n - 1$, direkt aus (3.11) folgt.

Unser Ziel ist es nun einen Vorhersagebereich für diese allgemeinere Version zu bestimmen, wie wir es in Korollar 3.15 für (3.6) getan haben. Dafür benötigen wir allerdings folgende Definition:

Definition 3.20. *(Wang und Qiao 2025, Lemma 2)* Seien $a_1, \dots, a_n$ Zahlen. Für $0 \leq k \leq n$ ist $e_k(a_1, \dots, a_n)$ das k-te *elementarsymmetrische Polynom*:

$$e_k(a_1, \dots, a_n) = \sum_{\substack{S \subseteq \{1,\dots,n\} \\ |S|=k}} \prod_{i \in S} a_i.$$

Damit ist also

$$\begin{aligned} e_0(a_1, \dots, a_n) &= 1, \\ e_1(a_1, \dots, a_n) &= a_1 + a_2 + \cdots + a_n, \\ e_2(a_1, \dots, a_n) &= \sum_{1 \leq i < j \leq n} a_i a_j, \\ &\vdots \\ e_n(a_1, \dots, a_n) &= a_1 a_2 \dots a_n. \end{aligned}$$

(Für eine effizientere Berechnung siehe Abschnitt 6.2.)

Mit den elementarsymmetrischen Polynomen können wir nun die allgemeinere Version der p_i^w, aus (3.4), für den Fall (3.11) umschreiben. Dabei orientieren wir uns an dem Vorgehen von *Wang und Qiao (2025)* für die Klassifikation.

Lemma 3.21. *Der Ausdruck* (3.4) *vereinfacht sich im Fall von covariate shift nach* (3.11) *zu*

$$p_i^w(z_1, \dots, z_n) = \frac{1}{n-l} \frac{w(x_i) e_{n-l-1}\big(w(x_1), \dots, w(x_{i-1}), w(x_{i+1}), \dots, w(x_n)\big)}{e_{n-l}\big(w(x_1), \dots, w(x_n)\big)}, \tag{3.12}$$

wobei $w = d\tilde{P}_X / dP_X$.

Beweis. Für die Indizes $i = 1, \dots, l$ gilt

$$w_i = \frac{dP_i}{dP_1} = \frac{d(P_X \times P_{Y|X})}{d(P_X \times P_{Y|X})} \equiv 1.$$

Und analog folgt für $j = l + 1, \dots, n$ die Gleichheit

$$w_j = \frac{d\tilde{P}_X}{dP_X} = w.$$

Damit gilt für (3.4)

$$p_i^w(z_1, \dots, z_n) = \frac{\sum_{\{\pi : \pi(n) = i\}} \prod_{j=1}^n w_j(z_{\pi(j)})}{\sum_\pi \prod_{j=1}^n w_j(z_{\pi(j)})} = \frac{\sum_{\{\pi : \pi(n) = i\}} \prod_{j=l+1}^n w(x_{\pi(j)})}{\sum_\pi \prod_{j=l+1}^n w(x_{\pi(j)})}.$$

Wir wollen nun Zähler und Nenner einzeln umformen. Dafür fixieren wir eine Permutation π. Die Menge der Indizes, welche den Positionen $l+1, \dots, n$ zugeordnet werden, sei S_π. Dann

ist $S_\pi \subset \{1,\dots,n\}$ und es gilt $|S_\pi| = n-l$. Die Menge S_π tritt dann für genau $(n-l)!\cdot l!$ Permutationen auf. Wir können daher die Summe über alle Permutationen schreiben als

$$\sum_\pi \prod_{j=l+1}^{n} w(x_{\pi(j)}) = \sum_{\substack{S\subseteq\{1,\dots,n\}\\|S|=n-l}} \left(l!\,(n-l)! \prod_{c\in S} w(x_c) \right) = l!\,(n-l)!\, e_{n-l}(w(x_1),\dots,w(x_n)).$$

Nun wollen wir die Summe über alle Permutationen mit $\pi(n) = i$ betrachten. In diesem Fall gilt immer $i \in S_\pi$ nach Konstruktion von S_π. Die Anzahl der Anordnungen auf die restlichen $n-l-1$ Plätze in S_π entspricht $(n-l-1)!$. Somit gibt es $(n-l-1)!\cdot l!$ verschiedene Permutationen, die zu S_π gehören. Es folgt also

$$\sum_{\{\pi:\pi(n)=i\}} \prod_{j=1}^{n} w(x_{\pi(j)}) = l!\,(n-l-1)! \sum_{\substack{S\subseteq\{1,\dots,n\}\\|S|=n-l, i\in S}} \prod_{c\in S} w(x_c).$$

Da $i \in S$ für alle S können wir schreiben $S = U \cap \{i\}$ für $U \subset \{1,\dots,n\}\setminus\{i\}$ und $|U| = n-l-1$. Daher gilt

$$\sum_{\substack{S\subseteq\{1,\dots,n\}\\|S|=n-l, i\in S}} \prod_{c\in S} w(x_c) = \sum_{\substack{U\subseteq\{1,\dots,n\}\setminus\{i\}\\|U|=n-l-1}} w(x_i) \prod_{c\in U} w(x_c) = w(x_i) e_{n-l-1}\big(w(x_1),\dots,w(x_{i-1}),w(x_{i+1}),\dots,w(x_n)\big).$$

Einsetzen in p_i^w liefert uns

$$p_i^w(z_1,\dots,z_n) = \frac{1}{n-l} \frac{w(x_i) e_{n-l-1}\big(w(x_1),\dots,w(x_{i-1}),w(x_{i+1}),\dots,w(x_n)\big)}{e_{n-l}\big(w(x_1),\dots,w(x_n)\big)}$$

□

Bemerkung 3.22. Im Fall von $n-l=1$ folgt mit Definition 3.20

$$\begin{aligned} p_i^w(z_1,\dots,z_n) &= \frac{w(x_i) e_0\big(w(x_1),\dots,w(x_{i-1}),w(x_{i+1}),\dots,w(x_n)\big)}{e_1\big(w(x_1),\dots,w(x_n)\big)} \\ &= \frac{w(x_i)}{\sum_{j=1}^n w(x_j)}, \end{aligned}$$

was gerade (3.7), für den Fall von covariate shift nach (3.6), entspricht.

Wir wollen nun eine allgemeinere Version von Korollar 3.15 notieren.

Korollar 3.23. *Seien Daten gegeben wie in* (3.11). *Angenommen $\tilde{P}_X$ ist absolut stetig bezüglich P_X und $w = d\tilde{P}_X/dP_X$. Für jedes nonconformity measure A und jedes $\alpha \in (0,1)$, definiere für $x_n \in \mathbb{R}^d$*

$$\Gamma_n^\alpha(x_n) = \left\{ y_n \in \mathbb{R} : R_n \leq \mathit{Quantil}\left(1-\alpha; \sum_{i=1}^{n-1} p_i^w(x_n)\delta_{R_i} + p_n^w(x_n)\delta_\infty\right)\right\},$$

wobei $R_i = A((z_1,\dots,z_n)^\top, z_i)$, $i=1,\dots,n$, mit $z_i = (x_i,y_i)$, und $p_i^w(x_n)$, $i=1,\dots,n$ wie in (3.12). *Dann ist der Vorhersagebereich $\Gamma_n^\alpha(x_n)$ konservativ valide, also*

$$\mathbb{P}\big(y_n \in \Gamma_n^\alpha(x_n)\big) \geq 1-\alpha.$$

Beweis. Da alle Daten aus (3.11) unabhängig sind und $\tilde{P}_X$ absolut stetig bezüglich P_X ist,

folgt mit Lemma 3.8, dass $Z_1, \ldots, Z_n$ gewichtet austauschbar sind. Wegen Lemma 3.21 ist

$$p_i^w(Z_1, \ldots, Z_{n-1}, (x_n, y_n)) = \frac{1}{n-l} \frac{w(x_i) e_{n-l-1}\big(w(x_1), \ldots, w(x_{i-1}), w(x_{i+1}), \ldots, w(x_n)\big)}{e_{n-l}\big(w(x_1), \ldots, w(x_n)\big)} = p_i^w(x_n)$$

und damit ergibt sich die Behauptung aus Satz 3.12. □

Analog zu Proposition 3.19 können wir Korollar 3.23 nutzen, um einen Vorhersagebereich für den Fall (3.11) zu konstruieren. Dafür seien nun die Daten $Z_1, \ldots, Z_{n-1}$ gegeben, wobei einige Daten der Zielverteilung $\tilde{P}$ folgen. Wollen wir nun die Kategorien für X_{n+m}, mit $m \geq 0$, bestimmen, so müssen wir $Z_1, \ldots, Z_n$ nur in Trainingsdatensatz $\mathcal{D}_1$ und Kalibrierungsdatensatz $\mathcal{D}_2$ aufteilen und da $\mathcal{D}_1$ die Regressionsfunktion $\hat{f}_0$ fixiert, erfüllt $\mathcal{D}_2$ die Voraussetzungen für Korollar 3.23. Damit entspricht der split conformal predictor für den verallgemeinerten Fall gerade (3.10) mit p_i^w wie in (3.12) für nonconformity scores $R_i = |Y_i - \hat{f}_0(X_i)|$.

4 Simulation

In diesem Kapitel wollen wir den Nutzen von conformal prediction anhand eines empirischen Beispiels demonstrieren. Dazu rekonstruieren wir einen Teil des Datenbeispieles von *Tibshirani u. a. (2019)*, wobei inhaltliche Veränderungen auftreten, da *Tibshirani u. a.* das gelöschte nonconformity measure (siehe Bemerkung 1.5) nutzen und wir eine andere Herangehensweise bei der Aufteilung der Daten wählen. Denn da wir nur einen Datensatz ohne covariate shift nutzen, müssen wir diesen künstlich erzeugen. *Tibshirani u. a. (2019)* tun dies nur in der Testmenge, während wir die Testmenge fixieren und stattdessen die Trainings- und Kalibrierungsmenge verändern. Dies hat den Vorteil, dass die Testmengen von gewichteten und ungewichteten conformal predictor gleich sind. Der Nachteil ist bei dieser Methode, dass wir den covariate shift für deutlich mehr Daten erzeugen müssen, was zu Verzerrungen in den Ergebnissen führen kann. Außerdem ist unser Code etwas spezialisierter als die Vorlage. Unser Ziel ist es dabei zu zeigen, dass der gewichtete conformal predictor für covariate shift auch praktisch anwendbar ist und die theoretische Validität einhält. Dazu wollen wir zuerst, ähnlich wie *Tibshirani u. a. (2019)*, den split conformal predictor aus Proposition 3.19 betrachten. Diesen vergleichen wir mit dem ungewichteten split conformal predictor aus Proposition 3.17. Als zweiten Teil dieser Simulation wollen wir überprüfen, ob der conformal predictor für die verallgemeinerte Version von covariate shift (3.11), ebenfalls die richtigen Ergebnisse liefert.

Dazu nutzen wir den selben Datensatz, welchen auch *Tibshirani u. a. (2019)* verwenden. Dieser enthält $N = 1503$ Beobachtungen für die Kategorie Y (skalierter Schalldruckpegel von NASA Tragflächen) mit 5-dimensionale Kovariablen X (log Frequenz, Anstellwinkel, Akkordlänge, free-stream-Geschwindigkeit, log Stärke der saugseitigen Verdrängung). Der Datensatz (airfoil.txt), sowie der R Code von *Tibshirani u. a. (2019)*, kann auf der Seite `http://www.github.com/ryantibs/conformal/` gefunden werden.

Simulation 1: Für unsere erste Simulation wollen wir den Datensatz in Trainings-, Kalibrierungs-, und Testmenge aufteilen, um dann zu überprüfen, ob die resultierenden Vorhersageintervalle, die wahren Werte aus der Testmenge enthalten. Dieses Experiment wiederholen wir 5000 Mal. Der Theorie dieser Arbeit zur Folge müssten die Intervalle mindestens 90% (für $\alpha = 0,1$) der wahren Daten erfassen (Überdeckung). Die Aufteilung der Daten $\{(X_i, Y_i)\}_{i=1}^N$ erfolgt wie folgt:

- $\mathcal{D}_1$: 25% der Daten, um die feste Regressionsfunktion $\hat{f}_0$, mittels linearer Regression (siehe Bemerkung 2.11), zu bestimmen.
- $\mathcal{D}_2$: 25% der Daten zur Bestimmung des Quantils der nonconformity scores.
- $\mathcal{D}_1^{\text{shift}}$: 25% der Daten, wie in $\mathcal{D}_1$, aber durch ziehen mit zurücklegen mit Wahrscheinlichkeit proportional zu

$$g(x) = \exp(x^\top \beta), \text{ mit } \beta = (-1.3, 0, 0, 0, 1.3).$$

- $\mathcal{D}_2^{\text{shift}}$: 25% der Daten zur Bestimmung des Quantils. Werden mit der selben Gewichtung, wie $\mathcal{D}_1^{\text{shift}}$, gezogen.
- $\mathcal{D}_{\text{test}}$: 30% der Daten, zum testen der Vorhersagebereiche aus Trainings- und Kalibrierungsdatensatz (mit und ohne covariate shift).

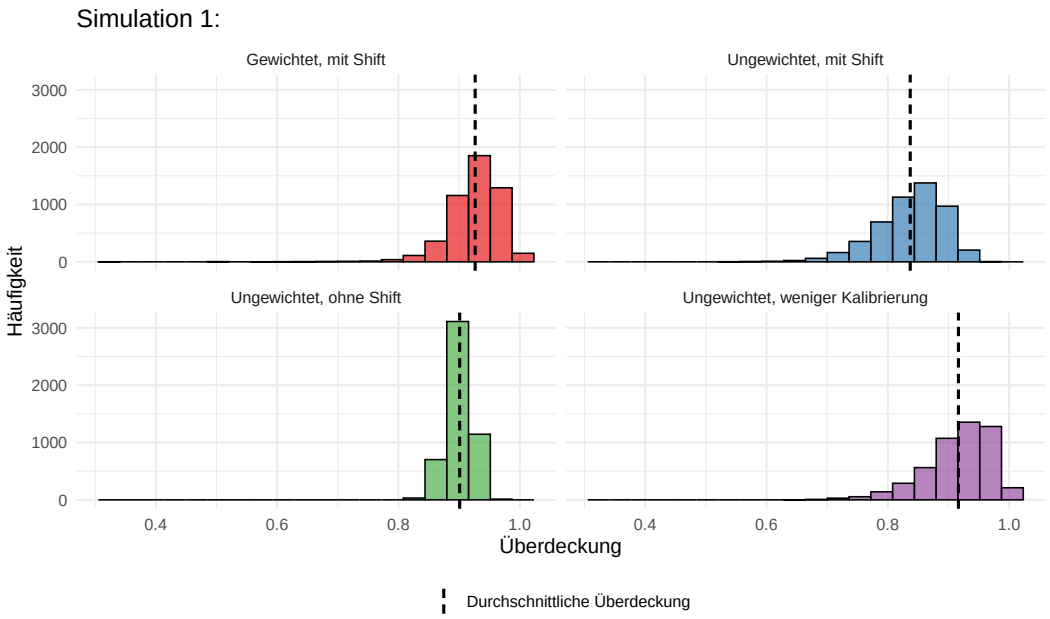

Abbildung 1: Überdeckungen bei Simulation 1. Oben links ist der Fall mit covariate shift, bei dem wir den gewichteten conformal predictor nutzen. Oben rechts testen wir wie sich der ungewichtete conformal predictor bei covariate shift verhält. Unten links ist der ungewichtete conformal predictor, ohne covariate shift abgebildet und unten rechts ebenfalls, allerdings mit einer kleineren Kalibrierungsmenge.

Wir verwenden nicht 100% der Daten, damit die Daten für den covariate shift aus einer größeren Stichprobe gezogen werden, um zu viele mehrfach gezogene Daten zu vermeiden.

Da $\mathcal{D}_2$ und $\mathcal{D}_{\text{test}}$ zufällig aus dem selben Datensatz gezogen wurden, können wir annehmen, dass bei beiden die selbe Verteilung zugrunde liegt. Wenn wir die Verteilung dieser Kovariablen nun als die Zielverteilung $\tilde{P}_X$ bezeichnen, dann sind die Kovariablen von $\mathcal{D}_1^{\text{shift}}$ und $\mathcal{D}_2^{\text{shift}}$ mit der Quellenverteilung P_X verteilt und es gilt die Aussage

$$dP_X \propto \exp(x^\top \beta) d\tilde{P}_X,$$

wobei $g(x) = \exp(x^\top \beta)$ die Likelihood-Ratio der beiden Verteilungen ist. Es folgt somit auch die Umkehrung

$$d\tilde{P}_X \propto \exp(-x^\top \beta) dP_X,$$

welche für den gewichteten conformal predictor relevant ist. Dabei werden wir die Gewichtsfunktion $w(x) = \exp(-x^\top \beta)$ nutzen.

Wenn wir also die Datensätze $\mathcal{D}_1^{\text{shift}}, \mathcal{D}_2^{\text{shift}}$ und $\mathcal{D}_{\text{test}}$ verwenden und $\alpha = 0,1$ wählen, dann können wir Proposition 3.19 anwenden und erhalten den gewichteten split conformal predictor für diese Daten mit covariate shift. Die Überdeckungen für diesen ist in Abbildung 1, oben links, dargestellt und die durchschnittliche Überdeckung ist mit 92,65% deutlich über dem 90%-Niveau. Die Notwendigkeit eines gewichteten conformal predictors wird durch das Histogram oben rechts verdeutlicht. Dieses zeigt die Überdeckungen für den ungewichteten conformal predictor aus Proposition 3.17, allerdings wurden als Trainings- und Kalibrierungsmengen $\mathcal{D}_1^{\text{shift}}$ und $\mathcal{D}_2^{\text{shift}}$ genutzt. Die durchschnittliche Überdeckung von 83,69% zeigt, dass die ungewichtete Variante nicht die richtigen Quantile für den covariate

shift erstellt. Wir wollen den gewichteten conformal predictor allerdings auch mit der klassischen, ungewichteten Variante vergleichen, wenn $\mathcal{D}_1$ und $\mathcal{D}_2$ gegeben sind. In diesem Fall liefern die Intervalle eine durchschnittliche Überdeckung von 90,09%, was genau der Theorie entspricht. Vergleicht man allerdings die Streuung der Überdeckung, dann fällt auf, dass der gewichtete conformal predictor (oben links) eine deutlich weitere Streuung aufweist als der ungewichtete Fall (Abbildung 1, unten links). Der Grund dafür liegt aber nicht an der Konstruktion des gewichteten conformal predictors, sondern an der Stichprobengröße (*vgl. Tibshirani u. a. 2019*). Denn durch die Gewichtung des Quantils, verringert sich die effektive Stichprobengröße. Wir bräuchten also mehr Daten, um die höhere Genauigkeit des ungewichteten conformal predictors zu erreichen. Wie effektiv die aktuelle Stichprobe ist lässt sich sogar direkt berechnen, mit (*Tibshirani u. a. 2019*)

$$\hat{n} = \frac{[\sum_{i=1}^{n} |w(X_i)|]^2}{\sum_{i=1}^{n} |w(X_i)|^2} = \frac{\|w(X_{1:n})\|_1^2}{\|w(X_{1:n})\|_2^2}.$$

Wenn wir nun den ungewichteten conformal predictor berechnen und die Größe von $\mathcal{D}_2$ auf $\hat{n}$ reduzieren, dann können wir in Abbildung 1 (unten rechts) sehen, dass die Streuung der Überdeckung sehr ähnlich zu der, des gewichteten conformal predictors ist.

Wir wollen zudem noch die Größe der Intervalle betrachten. Denn dort fällt ein Unterschied zwischen dem gewichteten und ungewichteten Fall auf. Während die mittlere Intervalllänge für den ungewichteten conformal predictor (ohne covariate shift) bei 16,07 liegt, ist diese bei der gewichteten Variante mit 20,19 etwas größer. Dies ist aber nicht weiter verwunderlich, da die Daten durch den covariate shift anders gestreut sind. Dadurch müssen die Intervalle größer werden, um die 90% Überdeckung zu erfüllen. Dies lässt sich auch durch den ungewichteten conformal predictor, bei dem wir die Daten mit covariate shift einsetzen, bestätigen. Dort liegt die Überdeckung (bei gleicher Regressionsfunktion) unter 90% und die mittlere Intervalllänge ist mit 17,93 schon näher an der des gewichteten conformal predictors. Außerdem bedingt eine kleinere (effektive) Stichprobe ebenfalls größere Intervalle, wie der conformal predictor mit weniger Kalibrierungsdaten zeigt. Hierbei ist die mittlere Intervalllänge 18,38. Es ist allerdings auffällig, dass die Überdeckung des gewichteten conformal predictors mit 92,65% um einiges größer ist als notwendig. Dies kann zwei Gründe haben. Zum einen haben wir bereits erwähnt, dass die obere Schranke für die Überdeckung bei dem gewichteten conformal predictor größer sein kann als bei der ungewichteten Variante. In dem Fall könnte das Problem mit einem geglätteten conformal predictor gelöst werden. Andererseits tritt dieses Problem bei *Tibshirani u. a. (2019)* überhaupt nicht auf. Daher besteht die Möglichkeit, dass unsere stärkere Einflussnahme auf die Daten, um den covariate shift zu erzeugen, dazu geführt hat, dass die Stichprobe aus weniger individuellen Datenpunkten besteht, welche durch die Gewichtung häufiger gezogen wurden. Diese Theorie wird auch durch das Ergebnis des ungewichteten conformal predictors, mit weniger Kalibrierungsdaten, gestützt. Dieser hat ebenfalls einen etwas höhere durchschnittliche Überdeckung mit 91,64%.

Simulation 2: Nachdem wir die Simulation von *Tibshirani u. a. (2019)* einmal selbst durchgeführt haben, werden wir nun überprüfen, ob der conformal predictor aus Abschnitt 3.5, für eine verallgemeinerte Form von covariate shift, ebenfalls die richtigen Ergebnisse liefert. Wir führen daher ein sehr ähnliches Experiment durch wie in Simulation 1, allerdings mit 500 Wiederholungen. Dazu müssen wir die Daten wie folgt aufteilen:

- $\mathcal{D}_1$: 25% der Daten für die Regressionsfunktion $\hat{f}_0$, wobei 90% davon proportional zu $g(x) = \exp(x^\top \beta)$, mit $\beta = (-1, 0, 0, 0, 1)$, gezogen werden.
- $\mathcal{D}_2$: 25% der Daten zur Bestimmung des Quantils, wobei 90% davon proportional zu $g(x)$ gezogen werden.
- $\mathcal{D}_{\text{test}}$: 30% der Daten zum testen direkt aus dem Datensatz.

Die Daten entsprechen somit dem Fall von covariate shift nach (3.11), bei dem wir Daten aus der Quellverteilung und der Zielverteilung zur Verfügung haben. Damit können wir Korollar 3.23 als split conformal predictor anwenden, wobei $w(x) = \exp(-x^\top \beta)$. Die Überdeckung der Intervalle, durch diesen Algorithmus, ist in Abbildung 2 (oben links) dargestellt.

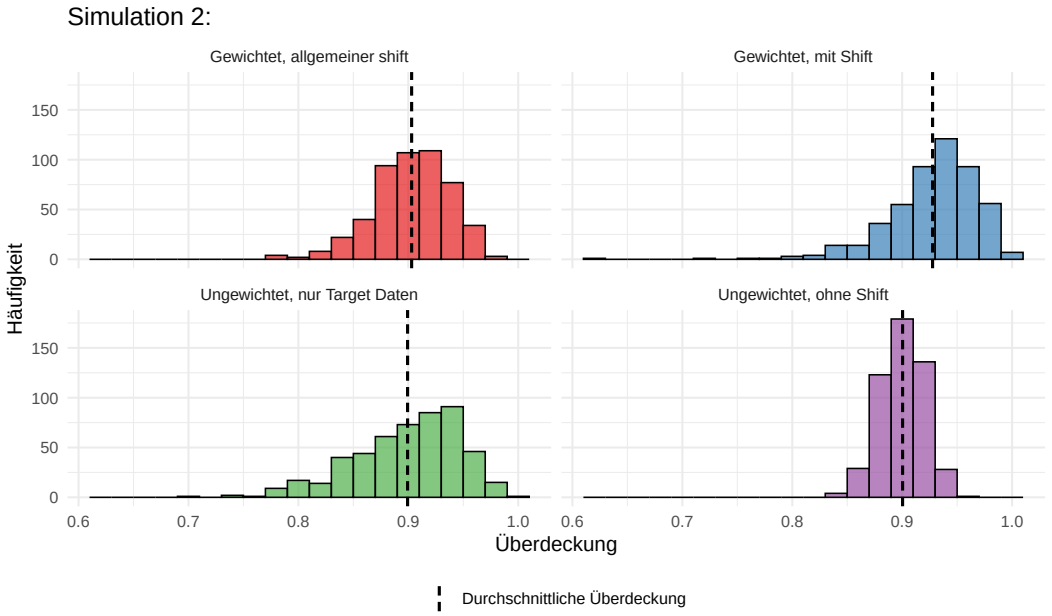

Abbildung 2: Oben links ist die Überdeckung für den Fall von covariate shift nach (3.11) und den conformal predictor aus 3.23. Unten links ist der ungewichtete conformal predictor, wobei $\mathcal{D}_1, \mathcal{D}_2$ nur die Daten enthält, die nicht nach $g(x)$ gezogen wurden. Auf der rechten Seite sind der gewichtete und ungewichtete conformal predictor aus Simulation 1, mit nur 500 Wiederholungen, nochmal abgebildet.

Die durchschnittliche Überdeckung von 90,3% erfüllt die geforderte Validität des conformal predictors. Dabei ist die Streuung der Überdeckung ähnlich zu der gewichteten Variante aus Simulation1, welche in Abbildung 2 nochmal oben recht abgebildet ist. Allerdings scheint die Streuung bei der allgemeineren Variante ein wenig geringer zu sein. Im Vergleich zu der ungewichteten Variante aus Simulation 1 (Abbildung 2 unten rechts) ist die Streuung jedoch deutlich stärker. Da die Daten bei dieser Simulation eine Mischung aus Source und Target Daten ist, war dies zu erwarten. Außerdem ist die Intervalllänge bei diesem verallgemeinerten conformal predictor mit 17,89 kleiner als bei dem gewichteten aus der vorherigen Simulation. Dies kann dadurch begründet werden, dass wir in Simulation 2 bessere Daten gegeben haben als bei Simulation 1 und dadurch die Regressionsfunktion $\hat{f}_0$ bereits etwas besser für $\mathcal{D}_{\text{shift}}$ geschätzt wird und nicht mehr so ein großer Bereich durch die Quantile abdecken müssen. Es wäre allerdings noch interessant diesen Fall zu überprüfen, wenn man geschätzte Gewichte verwendet und nicht direkt das wahre Gewicht nutzt. Denn in der Praxis müssen die Gewichte immer geschätzt werden. *Tibshirani u. a. (2019)* schätzen, bei ihrer

Simulation, die Gewichte mittels logistischer Regression oder random forest.
Es ist für uns allerdings ebenfalls von Interesse, inwiefern es überhaupt sinnvoll ist, bei einer Mischung der Daten aus Source und Target, einen gewichteten conformal predictor zu nutzen. Daher haben wir ebenfalls getestet, wie sich die Überdeckung für den ungewichteten conformal predictor verhält, wenn wir nur die 10% Target Daten von $\mathcal{D}_1$ und $\mathcal{D}_2$ verwenden, die aus der selben Verteilung wie $\mathcal{D}_{\text{test}}$ stammen. Dies ist in Abbildung 2 unten links abgebildet. Die Durchschnittliche Überdeckung ist mit 89,93% nur leicht unter dem 90% Niveau. Jedoch ist die Streuung der Überdeckung deutlich stärker als bei allen anderen Varianten. Außerdem kommt hinzu, dass die mittlere Intervalllänge, bei schlechterer Überdeckung, mit 18,51, größer ist als bei dem gewichteten conformal predictor mit gemischten Daten. Wir können daher festhalten, dass wir bei nur wenig Daten aus der Zielverteilung, mit zusätzlichen Daten aus einer anderen Verteilung bessere Ergebnisse erzielen können. Es gibt jedoch ein zentrales Problem, wenn wir Vorhersagebereiche für diese allgemeinere Form von covariate shift berechnen wollen.

Denn bei dieser Variante tritt, im Gegensatz zu den Varianten aus Simulation 1, folgendes Problem auf. So führt die Verallgemeinerung der Gewichtung in Lemma 3.21 zu einem starken Anstieg des Rechenaufwandes, wodurch die Berechnung der Vorhersagebereiche deutlich zeitaufwändiger ist. Aus diesem Grund haben wir in Simulation 2 auch nur 500 Wiederholungen durchgeführt. Außerdem ist die Berechnung des gewichteten Quantils komplizierter und sorgt bei der Ausführung des Codes für numerische Probleme. Aus diesem Grund mussten wir ein anderes β als in Simulation 1 wählen. In der aktuellen Form ist es fraglich wie nützlich diese Variante wirklich ist, wenn wir bereits gute Ergebnisse mit dem ungewichteten conformal predictor erzielen. Der R Code, zu den beiden Simulationen, ist in Abschnitt 6.3 zu finden.

5 Fazit und Ausblick

Das Ziel dieser Arbeit war es einen Vorhersagebereich mittels conformal prediction zu entwickeln und dies auf covariate shift auszuweiten. Dazu haben wir im ersten Kapitel den Grundstein gelegt, indem wir conformal prediction und alle relevanten Begriffe eingeführt haben. Daraus haben wir in Satz 1.9 den ersten konservativen Vorhersagebereich mit den Methoden von conformal prediction konstruiert. Im Anschluss sind wir im zweiten Kapitel auf den Fall eingegangen, bei dem eine Regressionsfunktion im nonconformity measure genutzt wird. Im Zuge dessen wurde uns die Problematik der effizienten Berechnung bei conformal prediction bewusst, welche wir mit dem split conformal predictor gelöst haben. Daraufhin sind wir auf einen anderen Ansatz, zur Konstruktion von Vorhersagebereichen, aufmerksam geworden und konnten mit diesem Ansatz einen gewichteten conformal predictor, mit Satz 3.12, präsentieren. Diesen konnten wir spezialisieren, um conformal prediction auf Daten unter covariate shift anzuwenden (Korollare 3.15 und 3.23). Zudem konnten wir zeigen, dass sich diese Vorhersagebereiche ebenfalls mit dem split conformal predictor berechnen lassen (Proposition 3.19), wodurch eine effizientere Berechnung möglich wird. Zum Abschluss haben wir unsere theoretischen Ergebnisse mit einer Simulation überprüft und konnten zeigen, dass diese auf echte Datensätze anwendbar sind.

Bei der Simulation ist allerdings zu beachten, dass wir keinen Datensatz mit echtem covariate shift verwenden. Dies hatte den Vorteil, dass wir die Gewichte selber bestimmt haben und diese uns damit bekannt waren. Es ist empfehlenswert noch weitere Untersuchungen bei realen Problemen vorzunehmen, um das Verhalten des Algorithmus besser zu analysieren und diesen genauer anzupassen. Insbesondere wäre es interessant zu sehen wie sicher der Algorithmus sich bei geschätzten Gewichten verhält. Ein weiteres Problem, welches für conformal prediction auch in Zukunft relevant sein wird, ist der hohe Rechenaufwand. Denn wenn conformal prediction in der Zukunft eine Rolle spielen sollte, muss es Algorithmen geben, welche die Vorhersagebereiche, auch bei komplizierten Fällen, effizient berechnen können.

Da conformal prediction noch ein relativ junges Forschungsgebiet ist, wird es spannend sein die Entwicklungen der nächsten Jahre zu verfolgen. So kann es auch sein, dass dann einige Aussagen dieser Arbeit bereits überholt sind. Es existieren jetzt schon Erweiterungen zu dem Ansatz von Tibshirani u. a. (2019), wie etwa Yang u. a. (2024), bei dem stabilere Algorithmen für die gewichtete Variante konstruiert werden. Allerdings ist dies alles nicht nur für Regression möglich. So gibt es bereits ähnliche Arbeiten zur Klassifikation, siehe Wang und Qiao (2025). Conformal prediction ist aber ein noch viel größeres Feld und es gibt viele weitere Aussagen, Ansätze und Ideen, welche hier noch nicht betrachtet wurden.

6 Anhang

6.1 Austauschbarkeit

Die einzige und somit auch wichtigste Voraussetzung, um unseren conformal prediction Algorithmus anzuwenden, ist die Annahme der Austauschbarkeit, der Werte $z_1, \dots, z_n$. Diese ist essentiell für die grundlegenden Aussagen, die wir im über conformal prediction treffen. Dafür beginnen wir mit einer sehr allgemeinen Definition von Austauschbarkeit, werden diese aber noch genauer untersuchen und ebenfalls ein einfaches Beispiel betrachten.

Definition 6.1. *(vgl. Vovk u. a. 2022, Kapitel 2.1.1)* Sei $\mathbb{P}$ eine Wahrscheinlichkeitsverteilung auf $\mathcal{Z}^\infty$. Dann ist $\mathbb{P}$ *austauschbar*, falls für jede messbare Teilmenge $E \subseteq \mathcal{Z}^n$ und Permutation π auf $\{1, \dots, n\}$ gilt

$$\mathbb{P}(\{(z_1, z_2, \dots) \in \mathcal{Z}^\infty : (z_1, \dots, z_n) \in E\}) = \mathbb{P}(\{(z_1, z_2, \dots) \in \mathcal{Z}^\infty : (z_{\pi(1)}, \dots, z_{\pi(n)}) \in E\}). \tag{6.1}$$

Für eine Verteilung $\mathbb{P}$ auf $\mathcal{Z}^N$ ist $\mathbb{P}$ *austauschbar*, falls für jede Teilmenge $E \subseteq \mathcal{Z}^N$ gilt

$$\mathbb{P}(E) = \mathbb{P}(\{(z_1, \dots, z_N) \in \mathcal{Z}^N : (z_{\pi(1)}, \dots, z_{\pi(n)}) \in E\}). \tag{6.2}$$

Wir schreiben auch

$$(Z_1, \dots, Z_n) \overset{d}{=} (Z_{\pi(1)}, \dots, Z_{\pi(n)})$$

für austauschbare Zufallsvariablen $Z_1, \dots, Z_n$.

Austauschbare Zufallsvariablen sind immer identisch verteilt. Allerdings ist die Annahme der Austauschbarkeit eine schwächere, als die der u.i.v. Zufallsvariablen, wie wir im folgenden kurz erörtern wollen.

Bemerkung 6.2. *(Klenke 2020, Bemerkung 12.2.)* Sind die Zufallsvariablen $(X_i)_{i\in\mathcal{I}}$ austauschbar verteilt. Dann ist, für den diskreten Fall, eine äquivalente Aussage:
Für paarweise unterschiedliche $i_1, \dots, i_n \in \mathcal{I}$ und $j_1, \dots, j_n \in \mathcal{I}$ gilt

$$\mathbb{P}(X_{i_1} = x_1, \dots, X_{i_n} = x_n) = \mathbb{P}(X_{j_1} = x_1, \dots, X_{j_n} = x_n).$$

Betrachten wir diese Aussage für $n = 1$, so folgt

$$\mathbb{P}(X_i = x) = \mathbb{P}(X_j = x)$$

für beliebige $i, j \in \mathcal{I}$. Die Zufallsvariablen $(X_i)_{i\in\mathcal{I}}$ sind also identisch verteilt.

Bemerkung 6.3. Für unabhängig und identisch verteilte Zufallsvariablen $(X_i)_{i\in\mathcal{I}}$ und paarweise verschiedene $i_1, \dots, i_n \in \mathcal{I}$ sowie $j_1, \dots, j_n \in \mathcal{I}$ folgt

$$\mathbb{P}(X_{i_1} = x_1, \dots, X_{i_n} = x_n) = \prod_{k=1}^{n} \mathbb{P}(X_{i_k} = x_k) = \prod_{k=1}^{n} \mathbb{P}(X_{j_k} = x_k) = \mathbb{P}(X_{j_1} = x_1, \dots, X_{j_n} = x_n).$$

Die Zufallsvariablen $(X_i)_{i\in\mathcal{I}}$ sind also austauschbar.

Es ist also klar, dass eine unabhängig und identische Verteilung, Austauschbarkeit implizieren. Das bedeutet, dass alle Ergebnisse in dieser Arbeit, welche Austauschbarkeit voraussetzen, ebenfalls für u.i.v. Zufallsvariablen gelten. Wie das folgende Beispiel verdeutlicht gilt die Umkehrung im Allgemeinen nicht.

Beispiel 6.4. Wir ziehen aus einer Urne mit 3 Kugeln dreimal ohne zurücklegen. Es gibt 2 rote Kugeln und wir definieren die Zufallsvariablen für $i = 1, 2, 3$,

$$X_i = \begin{cases} 1, \text{ falls die i-te Kugel rot ist,} \\ 0, \text{ sonst.} \end{cases}$$

Dann ist $(X_i)_{i=1,2,3}$ austauschbar wegen

$$\mathbb{P}(X_1 = 1, X_2 = 1, X_3 = 0) = \mathbb{P}(X_1 = 1, X_2 = 0, X_3 = 1) = \mathbb{P}(X_1 = 0, X_2 = 1, X_3 = 1) = \frac{1}{3},$$

allerdings nicht unabhängig, da

$$\begin{aligned} \mathbb{P}(X_1 = 1) \cdot \mathbb{P}(X_2 = 1) \cdot \mathbb{P}(X_3 = 0) &= \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} \\ &= \frac{4}{27} \neq \mathbb{P}(X_1 = 1, X_2 = 1, X_3 = 0). \end{aligned}$$

6.2 Rekursive Berechnung von elementarsymmetrischen Polynomen

Für eine effiziente Berechnung von elementarsymmetrischen Polynomen, nach Definition 3.20, sollte die Rekursive Formel

$$\begin{aligned} e_k(a_1, \ldots, a_n) &= \sum_{\substack{S \subseteq \{1,\ldots,n\} \\ |S|=k}} \prod_{i \in S} a_i \\ &= \sum_{\substack{S \subseteq \{1,\ldots,n\} \\ |S|=k,\, n \notin S}} \prod_{i \in S} a_i + \sum_{\substack{S \subseteq \{1,\ldots,n\} \\ |S|=k,\, n \in S}} \prod_{i \in S} a_i \\ &= \sum_{\substack{S \subseteq \{1,\ldots,n-1\} \\ |S|=k}} \prod_{i \in S} a_i + a_n \sum_{\substack{S \subseteq \{1,\ldots,n-1\} \\ |S|=k-1}} \prod_{i \in S} a_i \\ &= e_k(a_1, \ldots, a_{n-1}) + a_n \cdot e_{k-1}(a_1, \ldots, a_{n-1}) \end{aligned} \tag{6.3}$$

für $k < n$ genutzt werden. Wir benötigen allerdings auch eine effizientere Formel für $e_k(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. Dafür definieren wir

$$\begin{aligned} F_{j,r} &:= e_r(a_1, \ldots, a_j), \\ B_{j,r} &:= e_r(a_j, \ldots, a_n). \end{aligned}$$

Dann folgt mit

$$\begin{aligned} e_k(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) &= \sum_{\substack{S \subseteq \{1,\ldots,n\} \setminus \{i\} \\ |S|=k}} \prod_{c \in S} a_c \\ &= \sum_{t=0}^{k} \left(\sum_{\substack{S \subseteq \{1,\ldots,i-1\} \\ |S|=t}} \prod_{c \in S} a_c \cdot \sum_{\substack{S \subseteq \{i+1,\ldots,n\} \\ |S|=k-t}} \prod_{c \in S} a_c \right) \\ &= \sum_{t=0}^{k} e_t(a_1, \ldots, a_{i-1}) e_{k-t}(a_{i+1}, \ldots, a_n) \end{aligned}$$

die Gleichung

$$e_k(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) = \sum_{t=0}^{k} F_{i-1,t} B_{i+1,k-t}. \tag{6.4}$$

Diese Formeln werden im R-Code, für die allgemeinere Version des gewichteten Quantils aus Korollar 3.23, verwendet.

6.3 R Code

```r
##### Split conformal predictor
# Split CP fuer ungewichteten Fall und effizienter fuer gewichteten Spezialfall
split.cp = function(D_1, D_2, D_test, alpha, w_cal, w_test=NULL, gewichtet=FALSE) {
    # Trainingsdatensatz D_1, Kalibrierungsdatensatz D_2, Testdatensatz D_test,
    # Signifikanzniveau alpha, Gewichte w_cal und w_test, gewichtet Wahr/Falsch
    lm_mod = lm(Y ~ ., data = D_1)
    f_hut = predict(lm_mod, newdata = D_2)
    R_i = abs(D_2$Y - f_hut)
    n_cal = nrow(D_2)
    n_test = nrow(D_test)

    if(gewichtet) {
        q = numeric(n_test)
        for(i in 1:n_test) {
            q[i] = gew.quantil(c(R_i, Inf),
            c(w_cal, ifelse(is.null(w_test),1,w_test[i])),
            1 - alpha)
        }
    } else {
        k = ceiling((n_cal + 1) * (1 - alpha))
        q = sort(R_i)[k]
    }

    pred = predict(lm_mod, newdata = D_test)
    lower = pred - q
    upper = pred + q
    return(list(lower = lower, upper = upper))
}

# gewichteter split CP fuer den verallgemeinerten Fall
split.cp.general = function(D_1, D_2, D_test, alpha, w_func, is_target_cal) {
    # D_1: Trainingsdaten
    # D_2: Kalibrierungsdaten
    # D_test: Testdaten
    # is_target_cal: logical Vektor, Laenge n_cal, TRUE wenn ein Punkt von Target ist
    lm_mod = lm(Y ~ ., data = D_1)
    f_hat_cal = predict(lm_mod, newdata = D_2)
    R_cal = abs(D_2$Y - f_hat_cal)
    n_cal = nrow(D_2)
    n_test = nrow(D_test)
    if (length(is_target_cal) != n_cal) stop("is_target_cal length mismatch")
    # raw w fuer cal und test
    w_cal_raw  = w_func(as.matrix(D_2[, 1:(ncol(D_2)-1), drop = FALSE]))
```

```
                w_test_raw = w_func(as.matrix(D_test[, 1:(ncol(D_test)-1), drop =
                    FALSE]))

                l_cal = sum(!is_target_cal)
                # Fuer jeden Punkt gewichtetes Quantil berechnen
                q = numeric(n_test)
                for (i in seq_len(n_test)) {
                        v_all = c(w_cal_raw, w_test_raw[i])
                        R_all = c(R_cal, Inf)

                        l_total = l_cal
                        q[i] = gew.quantil.general(R_all, v_all, prob = 1 - alpha, l
                            = l_total)
                }
                pred = predict(lm_mod, newdata = D_test)
                list(lower = pred - q, upper = pred + q)
        }

        #### Gewichtetes Quantil
        # Fuer Spezialfall (effizienter)
        gew.quantil = function(R, weight, prob) {
                # nonconformity scores R, Gewichte weight, Quantil zur
                    Wahrscheinlichkeit prob
                # Sortiere R
                ord = order(R)
                R_sorted = R[ord]
                w_sorted = weight[ord]
                # Kumulierte Gewichte
                cum_w = cumsum(w_sorted / sum(w_sorted))
                # Finde kleinstes x, so dass cum_w >= 1-alpha
                i = which(cum_w >= prob)
                q = R_sorted[min(i)]
                return(q)
        }

        # Verallgemeinerter Fall
        gew.quantil.general = function(R, v, prob, l) {
                # R: Vektor Laenge n (nonconformity scores)
                # v: Vektor Laenge n fuer gewichte w(x_i)
                # l: Anzahl Source unter den n Punkten
                # prob: z.B. 1 - alpha

                n = length(R)
                k = n - l
                if (k <= 0) stop("Mindestens ein Target notwendig (k = n-l >= 1)")

                ## numerische Stabilisierung: skaliere v
                if (any(!is.finite(v))) stop("Gewichte enthalten nicht-finite Werte")
                s = max(v)
                if (s <= 0) stop("Ungueltige Gewichte (max(v) <= 0)")
                v = v / s
```

```
92
93              ## Forward-DP: fwd[j+1, m+1] = e_m(v[1:j])
94              fwd = matrix(0, n+1, k+1)
95              fwd[1,1] = 1
96              for (j in 1:n) {
97                      fwd[j+1, ] = fwd[j, ]
98                      for (m in 1:min(k, j)) {
99                              fwd[j+1, m+1] = fwd[j+1, m+1] + v[j] * fwd[j, m] #
                                    Gleichung (6.3)
100                     }
101             }
102             ## Backward-DP: bwd[j, m+1] = e_m(v[j:n])
103             bwd = matrix(0, n+2, k+1)
104             bwd[n+1,1] = 1
105             for (j in n:1) {
106                     bwd[j, ] = bwd[j+1, ]
107                     for (m in 1:min(k, n-j+1)) {
108                             bwd[j, m+1] = bwd[j, m+1] + v[j] * bwd[j+1, m] #
                                    Gleichung (6.3)
109                     }
110             }
111
112             den = fwd[n+1, k+1]   # e_k(v)
113             if (den <= 0 || !is.finite(den)) stop("Ungueltiger Nenner e_k(v)")
114
115             ## p_i^w berechnen mit Gleichung (6.4)
116             p = numeric(n)
117             for (i in 1:n) {
118                     num_e = 0
119                     for (s in 0:(k-1)) {
120                             num_e = num_e + fwd[i, s+1] * bwd[i+1, (k-1-s)+1]
121                     }
122                     num = v[i] * num_e
123                     p[i] = (1/k) * num / den
124             }
125
126             ## Normalisieren
127             p[p < 0] = 0
128             p = p / sum(p)
129
130             ## Quantil bestimmen
131             ord = order(R)
132             cs = cumsum(p[ord])
133             j = which(cs >= prob)[1]
134             if (is.na(j)) j = length(R)
135             R[ord][j]
136     }
137     #### Simulation
138
139     ### Datenvorbereitung
140     dat = read.table("airfoil.txt")
```

```
        colnames(dat) = c("Frequency","Angle","Chord","Velocity","Suction","Sound")

        # log Transform
        dat$Frequency = log(dat$Frequency)
        dat$Suction = log(dat$Suction)

        # Trennen von Objekt X und Lable Y
        dat.x = as.matrix(dat[, 1:5])
        dat.y = as.numeric(dat[, 6])

        N = nrow(dat.x)
        p = ncol(dat.x)

        # Gewichtsfunktion fuer covariate shift mit exponentialen Tilting
        w = function(x) {
                exp(x[, c(1,5)] %*% c(-1.3, 1.3))
        }

        # Gegengewicht, fuer split cp
        gw = function(x) {
                exp(x[, c(1,5)] %*% c(1.3, -1.3))
        }
        ### Simulation 1
        set.seed(0)
        B = 5000
        alpha = 0.1

        # Ueberdeckung Speichern
        cover_unweighted = numeric(B)
        cover_unweighted_shift = numeric(B)
        cover_weighted_shift = numeric(B)
        cover_unweighted_fewersamples =numeric(B)

        # Intervalllaenge speichern
        len_unweighted = numeric(B)
        len_unweighted_shift = numeric(B)
        len_weighted_shift = numeric(B)
        len_unweighted_fewersamples = numeric(B)

        # Groesse von Trainings- und Kalibrierungsmenge
        n_pre = floor(0.25 * N)
        n_cal = floor(0.25 * N)
        n_test = floor(0.30 * N)

        # Begin der Schleife
        for(b in 1:B) {
                idx = sample(1:N)

                # Erstellen von Trainings-, Kalibrierungs und Testmenge
                idx_pre  = idx[1:n_pre]
                idx_cal  = idx[(n_pre + 1):(n_pre + n_cal)]
```

```r
                idx_test = idx[(n_pre + n_cal + 1):N]

                # Format der Daten auf split.cp() anpassen
                D_1 = data.frame(dat.x[idx_pre, ]); colnames(D_1) = paste0("X", 1:5); D_1$Y = dat.y[idx_pre]
                D_2 = data.frame(dat.x[idx_cal, ]); colnames(D_2) = paste0("X", 1:5); D_2$Y = dat.y[idx_cal]
                D_test = data.frame(dat.x[idx_test, ]); colnames(D_test) = paste0("X", 1:5); D_test$Y = dat.y[idx_test]

                # kuenstlichen covariate shift in D_1 und D_2
                rest = setdiff(1:N, idx_test)

                idx_rest_perm = sample(rest, size = (n_pre + n_cal), replace = TRUE, prob = w(dat.x[rest, ]))
                idx_pre_shift = idx_rest_perm[1:n_pre]
                idx_cal_shift = idx_rest_perm[(n_pre + 1):(n_pre + n_cal)]

                D_1_shift = data.frame(dat.x[idx_pre_shift, ]); colnames(D_1_shift) = paste0("X",1:5); D_1_shift$Y = dat.y[idx_pre_shift]
                D_2_shift = data.frame(dat.x[idx_cal_shift, ]); colnames(D_2_shift) = paste0("X",1:5); D_2_shift$Y = dat.y[idx_cal_shift]

                # Ungewichteter CP (ohne covariate shift)
                res_unweighted = split.cp(D_1, D_2, D_test, alpha, gewichtet = FALSE)
                cover_unweighted[b] = mean(D_test$Y >= res_unweighted$lower & D_test$Y <= res_unweighted$upper)
                len_unweighted[b] = median(res_unweighted$upper - res_unweighted$lower)

                # Ungewichteter CP mit covariate shift
                res_unweighted_shift = split.cp(D_1_shift, D_2_shift, D_test, alpha, gewichtet = FALSE)
                cover_unweighted_shift[b] = mean(D_test$Y >= res_unweighted_shift$lower & D_test$Y <= res_unweighted_shift$upper)
                len_unweighted_shift[b] = median(res_unweighted_shift$upper - res_unweighted_shift$lower)

                # Gewichteter CP mit bekannten Gewicht
                gw_cal = gw(as.matrix(D_2_shift[, 1:5]))
                gw_test = gw(as.matrix(D_test[,1:5]))
                res_weighted_shift = split.cp(D_1_shift, D_2_shift, D_test, alpha, gw_cal, gw_test, TRUE)
                cover_weighted_shift[b] = mean(D_test$Y >= res_weighted_shift$lower & D_test$Y <= res_weighted_shift$upper)
                len_weighted_shift[b] = median(res_weighted_shift$upper - res_weighted_shift$lower)

                # Ungewichteter CP ohne covariate shift mit kleinerer Kalibrierungsstichprobe
```

```
227                 w_cal = gw(dat.x[idx_cal, ])
228                 n_hat = round((sum(abs(w_cal))^2)/sum(w_cal^2))  # Effektive
                        Stichprobengroesse
229                 idx_few = sample(1:nrow(D_2), size = n_hat, replace =FALSE)
230                 D_2_fewer = D_2[idx_few, ]
231                 res_unweighted_fewersamples = split.cp(D_1, D_2_fewer, D_test, alpha,
                         w_unif, w_unif_test, FALSE)
232                 cover_unweighted_fewersamples[b] = mean(D_test$Y >=
                        res_unweighted_fewersamples$lower & D_test$Y <=
                        res_unweighted_fewersamples$upper)
233                 len_unweighted_fewersamples[b] = median(
                        res_unweighted_fewersamples$upper -
                        res_unweighted_fewersamples$lower)
234         }
235         ### Ergebnisse
236         # Ueberdeckung
237         cat("Durchschnittliche Ueberdeckung (ungewichtet ohne covariate shift):",
                mean(cover_unweighted), "\n")
238         cat("Durchschnittliche Ueberdeckung (ungewichtet mit covariate shift):", mean
                (cover_unweighted_shift), "\n")
239         cat("Durchschnittliche Ueberdeckung (gewichtet mit covariate shift):", mean(
                cover_weighted_shift), "\n")
240         cat("Durchschnittliche Ueberdeckung (ungewichtet ohne covariate shift mit
                kleinerem Kalibrierungsdatensatz):", mean(cover_unweighted_fewersamples),
                 "\n")
241
242         # Intervalllaenge
243         cat("Mittlere Intervalllaenge (ungewichtet ohne Shift):", mean(len_unweighted
                ), "\n")
244         cat("Mittlere Intervalllaenge (ungewichtet mit Shift):", mean(
                len_unweighted_shift), "\n")
245         cat("Mittlere Intervalllaenge (gewichtet mit Shift):", mean(
                len_weighted_shift), "\n")
246         cat("Mittlere Intervalllaenge (ungewichtet, kleinerer Kalibrierungsdatensatz)
                :", mean(len_unweighted_fewersamples), "\n")
247
248         # Gewichtsfunktion mit anderem beta
249         w2 = function(x) {
250                 exp(x[, c(1,5)] %*% c(-1, 1))
251         }
252
253         gw2 = function(x) {
254                 exp(x[, c(1,5)] %*% c(1, -1))
255         }
256
257         ### Simulation 2
258         set.seed(0)
259         C = 500
260         # Speicher
261         cover_weighted_gen = numeric(C)
262         len_weighted_gen   = numeric(C)
```

```
263         cover_target = numeric(C)
264         len_target = numeric(C)
265 
266         for (b in 1:C) {
267                  # D_test Testdaten
268                 idx_test = sample(1:N, size = n_test, replace = FALSE)
269 
270                 # Erstellen von gemischten D_1 und D_2
271                 rest = setdiff(1:N, idx_test)
272 
273                 n_T = round(0.10 * n_pre) # 10% Anteil
274                 n_S = n_pre - n_T
275 
276                 idx_Source = sample(rest, size = 2*n_S, replace = TRUE, prob = w2(dat
                        .x[rest, ]))
277                 idx_Source_pre = idx_Source[1:n_S]
278                 idx_Source_cal = idx_Source[(n_S+1):(2*n_S)]
279 
280                 rest_T = setdiff(rest, idx_Source)
281                 idx_Target = sample(rest_T, size = 2*n_T, replace = FALSE)
282                 idx_Target_pre = idx_Target[1:n_T]
283                 idx_Target_cal = idx_Target[(n_T+1):(2*n_T)]
284 
285                 idx_pre = c(idx_Source_pre, idx_Target_pre)
286                 idx_cal = c(idx_Source_cal, idx_Target_cal)
287 
288                 # Markiert ob Target oder Source
289                 is_target_cal = c(rep(FALSE, n_S), rep(TRUE, n_T))
290 
291                 # Data Frames
292                 D_1 = data.frame(dat.x[idx_pre, , drop = FALSE]); colnames(D_1) =
                        paste0("X",1:p); D_1$Y = dat.y[idx_pre]
293                 D_2 = data.frame(dat.x[idx_cal, , drop = FALSE]); colnames(D_2) =
                        paste0("X",1:p); D_2$Y = dat.y[idx_cal]
294                 D_test = data.frame(dat.x[idx_test, , drop = FALSE]); colnames(D_test
                        ) = paste0("X",1:p); D_test$Y = dat.y[idx_test]
295                 D_1_Target = data.frame(dat.x[idx_Target_pre, , drop = FALSE]);
                        colnames(D_1_Target) = paste0("X",1:p); D_1_Target$Y = dat.y[
                        idx_Target_pre]
296                 D_2_Target = data.frame(dat.x[idx_Target_cal, , drop = FALSE]);
                        colnames(D_2_Target) = paste0("X",1:p); D_2_Target$Y = dat.y[
                        idx_Target_cal]
297 
298                 # gewichteter split CP (verallgemeinert)
299                 res_w = split.cp.general(D_1, D_2, D_test, alpha = alpha, w_func =
                        gw2, is_target_cal = is_target_cal)
300                 lo_w = res_w$lower; up_w = res_w$upper
301                 cover_weighted_gen[b] = mean(D_test$Y >= lo_w & D_test$Y <= up_w)
302                 len_weighted_gen[b] = median(up_w - lo_w)
303 
304                 # ungewichteter split cp nur mit Target Daten
```

```
                res_T = split.cp(D_1_Target, D_2_Target, D_test, alpha, gewichtet =
                    FALSE)
                lo_T = res_T$lower; up_T = res_T$upper
                cover_target[b] = mean(D_test$Y >= lo_T & D_test$Y <= up_T)
                len_target[b] = median(up_T - lo_T)
                }
        cat("Durchschnittliche Ueberdeckung (gewichtet mit covariate shift,
            verallgemeinert):", mean(cover_weighted_gen, na.rm=TRUE), "\n")
        cat("Mittlere Intervalllaenge (gewichtet mit Shift, verallgemeinert):", mean(
            len_weighted_gen, na.rm=TRUE), "\n")
        cat("Durschnittliche Ueberdeckung (ungewichtet ohne covariate shift, geringer
             Datensatz):", mean(cover_target, na.rm=TRUE), "\n")
        cat("Mittlere Intervalllaenge (ungewichtet ohne covariate shift, geringer
            Datensatz):", mean(len_target, na.rm=TRUE), "\n")
        #### Plots
        library(ggplot2)
        library(gridExtra)

        ### Ergebnisse aus Simulation 1 zusammenfassen
        df1_cover = data.frame(
        Wert = c(cover_unweighted,
        cover_unweighted_shift,
        cover_weighted_shift,
        cover_unweighted_fewersamples),
        Methode = factor(rep(c("Ungewichtet, ohne Shift",
        "Ungewichtet, mit Shift",
        "Gewichtet, mit Shift",
        "Ungewichtet, weniger Kalibrierung"),
        each = B))
        )
        means_df = aggregate(Wert ~ Methode, df1_cover, mean)

        ## Histogramme Simulation 1
        p1 = ggplot(df1_cover, aes(x = Wert)) +
        geom_histogram(aes(fill = Methode),
        alpha = 0.7, position = "identity", bins = 20, color = "black") +
        facet_wrap(~Methode, scales = "fixed") +
        geom_vline(data = means_df,
        aes(xintercept = Wert, linetype = "Durchschnittliche Ueberdeckung"),
        color = "black", size = 1) +
        scale_linetype_manual(name = NULL,
        values = c("Durchschnittliche Ueberdeckung" = "32")) +
        scale_fill_manual(values = RColorBrewer::brewer.pal(length(unique(
            df1_cover$Methode)), "Set2"),
        guide = "none") +
        scale_fill_brewer(palette = "Set1", guide = "none") +
        theme_minimal(base_size = 14) +
        labs(title = "Simulation 1:", x = "Ueberdeckung", y = "Haeufigkeit") +
        theme(legend.position = "bottom")

        grid.arrange(p1, ncol = 1)
```

```
350
351         ### Ergebnisse aus Simulation 2 zusammenfassen
352         # Reduziere Daten aus Simulation 1 auf C=500
353         cover_unweighted_C = cover_unweighted[1:C]
354         cover_weighted_shift_C = cover_weighted_shift[1:C]
355
356         df2_cover = data.frame(
357         Wert = c(cover_unweighted_C,
358         cover_weighted_gen,
359         cover_weighted_shift_C,
360         cover_target),
361         Methode = factor(rep(c("Ungewichtet, ohne Shift",
362         "Gewichtet, allgemeiner shift",
363         "Gewichtet, mit Shift",
364         "Ungewichtet, nur Target Daten"),
365         each = C))
366         )
367         means_df2 = aggregate(Wert ~ Methode, df2_cover, mean)
368
369         ### Histogramme Simulation 2
370         p2 = ggplot(df2_cover, aes(x = Wert)) +
371         geom_histogram(aes(fill = Methode),
372         alpha = 0.7, position = "identity", bins = 20, color = "black") +
373         facet_wrap(~Methode, scales = "fixed") +
374         geom_vline(data = means_df2,
375         aes(xintercept = Wert, linetype = "Durchschnittliche Ueberdeckung"),
376         color = "black", size = 1) +
377         scale_linetype_manual(name = NULL,
378         values = c("Durchschnittliche Ueberdeckung" = "32")) +
379         scale_fill_manual(values = RColorBrewer::brewer.pal(length(unique(
              df2_cover$Methode)), "Set2"),
380         guide = "none") +
381         scale_fill_brewer(palette = "Set1", guide = "none") +
382         theme_minimal(base_size = 14) +
383         labs(title = "Simulation 2:", x = "Ueberdeckung", y = "Haeufigkeit") +
384         theme(legend.position = "bottom")
385
386         grid.arrange(p2, ncol = 1)
```

Literatur

Balasubramanian, Vineeth, Shen-Shyang Ho und Vladimir Vovk, Hrsg. (2014). *Conformal prediction for reliable machine learning. Theory, adaptations and applications*. English. Amsterdam: Elsevier. ISBN: 978-0-12-398537-8. URL: www.sciencedirect.com/science/book/9780123985378.

Chernozhukov, Victor, Kaspar Wüthrich und Yinchu Zhu (2021). "Distributional conformal prediction". English. In: *Proc. Natl. Acad. Sci. USA* 118.48. Id/No e2107794118, S. 9. ISSN: 0027-8424. DOI: 10.1073/pnas.2107794118.

Fontana, Matteo, Gianluca Zeni und Simone Vantini (2023). "Conformal prediction: a unified review of theory and new challenges". English. In: *Bernoulli* 29.1, S. 1–23. ISSN: 1350-7265. DOI: 10.3150/21-BEJ1447. URL: projecteuclid.org/journals/bernoulli/volume-29/issue-1/Conformal-prediction-A-unified-review-of-theory-and-new/10.3150/21-BEJ1447.full.

Gammerman, Alexander, Vladimir Vovk und Vladimir Vapnik (1998). "Learning by transduction". In: *Proceedings of the Conference on Machine Learning*. Morgan Kaufmann, S. 148–155.

Hore, Rohan und Rina Foygel Barber (2025). "Conformal prediction with local weights: randomization enables robust guarantees". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 87.2, S. 549–578.

Jin, Ying und Zhimei Ren (2025). "Confidence on the focal: Conformal prediction with selection-conditional coverage". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology*, qkaf016.

Klenke, Achim (2020). *Wahrscheinlichkeitstheorie*. German. 4th revised and supplemented edition. Masterclass. Berlin: Springer Spektrum. ISBN: 978-3-662-62088-5; 978-3-662-62089-2. DOI: 10.1007/978-3-662-62089-2.

Lei, Jing, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani und Larry Wasserman (2018). "Distribution-free predictive inference for regression". English. In: *J. Am. Stat. Assoc.* 113.523, S. 1094–1111. ISSN: 0162-1459. DOI: 10.1080/01621459.2017.1307116. URL: figshare.com/articles/journal_contribution/Distribution-Free_Predictive_Inference_For_Regression/4812304.

Papadopoulos, Harris, Kostas Proedrou, Volodya Vovk und Alex Gammerman (2002). "Inductive confidence machines for regression". English. In: *Machine learning: ECML 2002. 13th European conference, Helsinki, Finland, August 19–23, 2002. Proceedings*. Berlin: Springer, S. 345–356. ISBN: 3-540-44036-4. DOI: 10.1007/3-540-36755-1_29.

Saleh, A. K. Md. Ehsanes, Mohammad Arashi und B. M. Golam Kibria (2019). *Theory of ridge regression estimation with applications*. English. Wiley Ser. Probab. Stat. Hoboken, NJ: John Wiley & Sons. ISBN: 978-1-118-64461-4; 978-1-118-64447-8. DOI: 10.1002/9781118644478.

Shafer, Glenn und Vladimir Vovk (2008). "A tutorial on conformal prediction". English. In: *J. Mach. Learn. Res.* 9, S. 371–421. ISSN: 1532-4435. URL: www.jmlr.org/papers/v9/shafer08a.html.

Tibshirani, Ryan J, Rina Foygel Barber, Emmanuel Candes und Aaditya Ramdas (2019). "Conformal prediction under covariate shift". In: *Advances in neural information processing systems* 32.

Vovk, Vladimir, Alexander Gammerman und Glenn Shafer (2022). *Algorithmic learning in a random world*. English. 2nd revised, updated, and expanded edition. Cham: Springer.

ISBN: 978-3-031-06648-1; 978-3-031-06651-1; 978-3-031-06649-8. DOI: 10.1007/978-3-031-06649-8.

Wang, Baozhen und Xingye Qiao (2025). "Conformal prediction under generalized covariate shift with posterior drift". In: *arXiv preprint arXiv:2502.17744*.

Yang, Yachong, Arun Kumar Kuchibhotla und Eric Tchetgen Tchetgen (2024). "Doubly robust calibration of prediction sets under covariate shift". English. In: *J. R. Stat. Soc., Ser. B, Stat. Methodol.* 86.4, S. 943–965. ISSN: 1369-7412. DOI: 10.1093/jrsssb/qkae009.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich, Till Jakob Kaiser, die vorliegende Bachelorarbeit selbstständig und ohne unzulässige Hilfe angefertigt habe. Alle verwendeten Quellen, Hilfsmittel und fremden Inhalte sind als solche kenntlich gemacht. Die Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und nicht veröffentlicht. Die eingereichte elektronische Fassung entspricht inhaltlich der gedruckten Fassung.

Ort, Datum	Unterschrift