technische universität
dortmund

Masterarbeit

Certifiable Active Class Selection in Multi-Class Classification

Martin Senz
Juli 2022

Gutachter:
Prof. Dr. Katharina Morik
Mirko Bunse

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Künstliche Intelligenz (LS-8)
https://www-ai.cs.uni-dortmund.de/

Inhaltsverzeichnis

Kapitel 1

Einleitung

Im Allgemeinen ist es ratsam, maschinelle Lernverfahren auf einem möglichst großen Datensatz zu trainieren, um die bestmögliche Modell-Performanz zu erreichen. Allerdings existieren Anwendungsszenarien, in denen die Beschaffung von Trainingsdaten aufwändig und teuer ist. Falls die Datenakquise hohe Kosten verursacht, besteht die Herausforderung, ein Modell mit möglichst wenigen Daten performant zu trainieren.

Im Zuge dessen gewann Active Class Selection (ACS) [13] [11] bei überwachten Klassifikationsproblemen stetig an Bedeutung. In ACS gilt die Voraussetzung, dass ein Datengenerator existiert, der in Abhängigkeit zum Klassenlabel Datenbeispiele unter Kosten produziert. Folglich kann, unter Berücksichtigung der Akquisekosten, das Klassenverhältnis der Trainingsdaten vom Anwender beziehungsweise vom Lernalgorithmus aktiv gewählt werden. ACS-Strategien versuchen Klassenverhältnisse für Trainingsdaten zu finden, welche die Performanz von maschinellen Lernmethoden maximieren. Dabei entscheidet eine ACS-Strategie in Iterationen, aus welchen Klassen die nächsten zu lernenden Trainingsbeispiele stammen sollten, die voraussichtlich die größte Verbesserung hinsichtlich der Modell-Performanz versprechen.

Active Class Selection kann in Anwendungen eingesetzt werden, die einen klassenabhängigen Datengenerator besitzen. Ein häufiges Anwendungsfeld sind generierte Daten, die beispielsweise aus aufwendigen Physiksimulationen stammen [4]. ACS wird auch in komplexen Versuchsaufbauten wie der Gehirn-Computer-Interaktion [19] eingesetzt, wo einzelne Trainingsinstanzen nur mit großem Aufwand erzeugt werden können und das Trainingsset individuell am Probanden aufgebaut werden muss.

1.1 Certifiable Active Class Selection

Die Analyse von ACS beschränkte sich in der Vergangenheit meistens auf die Evaluation der Modell-Performanz unter einer ACS-Strategie [8] [19] [25] [24]. Ein oft vernachlässigter Umstand ist, dass durch die aktive Wahl der Klassenverhältnisse der Trainingsdaten

eine Domänenlücke (engl. *domain-gap*) gebildet wird, die sich auf das Modellverhalten auswirken kann. Diese Domänenlücke resultiert aus Klassenverschiebung zwischen ACS-generierten Trainingsdaten und den Deploymentdaten aus der Anwendung.

Nachdem die *domain-gap* qualitativ untersucht wurde [7], konnte in [6] unter Einbettung der Statistischen Lerntheorie, der durch divergierende Klassenverteilungen entstehende Fehler erfolgreich quantifiziert werden. Basierend auf der Theorie der Quantifizierung des Einflusses der Domänenlücke wurde eine Methode [6] vorgestellt, die trainierte binäre Klassifizierungsprobleme gegen Label-Verschiebungen zertifiziert. Dieses Zertifikat basiert auf PAC-Schranken und besitzt eine formal statistische Aussagekraft, die durch den Anwender als erlaubten Fehler ϵ und erlaubte Fehlerwahrscheinlichkeit δ spezifiziert werden kann. Ein solches Zertifikat, dass die Robustheit eines Modells bezüglich Klassenverschiebungen im Deployment adressiert, ist in der Modellvalidierung gefragt.

Das Validieren von Modellen gewinnt durch den verstärkten Einsatz von maschinellen Modellen stetig an Bedeutung. Eingesetzte Modelle, deren Leistungsverhalten nicht ausgiebig erprobt wurden, können großen Schaden verursachen. Folglich werden im Vorfeld Modelle durch umfangreiche Evaluierungen, in Form von *model reports* [16] [1] [20], getestet, bevor diese in Anwendungen implementiert werden. Die Robustheit von Modellen gegenüber Klassenverschiebungen ist in den Evaluations-Berichten der Modelle ein vertrauenskritischer Punkt im Kontext der Modellsicherheit.

Dementsprechend kann unter der Verwendung des ACS-Frameworks eine zentrale Robustheitseigenschaft analysiert und bewertet werden. Allerdings konnte die Zertifizierung in [6] bislang nur für binäre Klassifikationsprobleme angewendet werden. Da sich viele Anwendungsprobleme nicht auf eine binäre Klassifikation beschränken lassen, stellt dies eine starke Einschränkung bezüglich einer breiteren Anwendung dar.

Die Verallgemeinerung der binären Zertifizierung auf Mehrklassenprobleme ist nicht trivial. Beispielsweise würde eine *one-vs-all* Dekomposition erfordern, dass ein Anwender die formale Aussagekraft des Zertifikats für jede Klasse individuell vorentscheiden müsste, statt die Aussagekraft global zu spezifizieren.

Ebenfalls lassen sich auf Grundlage der quantitativen Betrachtungen Akquirierungsstrategien herleiten, die die *domain-gap* beachten und minimieren. Eine solche ACS-Strategie wird in [5] vorgestellt und evaluiert. Damit die Domänenlücke durch die sukzessive Akquise von Trainingsdaten minimiert werden kann, benötigt die Strategie einen Prior über die Klassenverteilung im Deployment, die durch den Anwender spezifiziert werden muss. Zwar sind in den meisten Anwendungsszenarien die Klassenverteilung der Deployment-Daten nicht exakt bekannt, jedoch können diese häufig hinreichend genau von Domänenexperten geschätzt werden. Die Strategie in [5] ist ebenfalls nur für binäre Klassifikationsprobleme anwendbar, was die Einsetzbarkeit gleichermaßen einschränkt.

1.1.1 Aufbau und Ziele der Arbeit

Das primäre Ziel dieser Arbeit ist es, die Anwendbarkeit der Zertifizierung auf Klassifizierungsprobleme mit beliebig vielen Klassen zu erweitern. Zu diesem Zweck werden mehrere Lösungsstrategien vorgestellt, die die Theorie gezielt erweitern und Einblicke in die Robustheit von Modellen in einem ACS-Framework geben. Eine erfolgreiche Verallgemeinerung der Theorie wäre ein wichtiger Schritt hin zu einer universelleren Anwendbarkeit von zertifizierbaren Klassifizierungsmodellen unter ACS.

Aufbauend auf den Erkenntnissen über die Mehrklassen-Zertifizierung wird auch ein Ausblick auf das Leistungsverhalten von ACS-Strategien bei Label-Verschiebungen gegeben. Ähnlich wie bei der in [5] vorgestellten Methode besteht das Ziel darin, eine Strategie für Multiklassenprobleme zu finden, die die Domänenlücke durch Optimierung der Datenakquise sukzessive minimiert.

Entsprechend der angestrebten Ziele ist die Arbeit in zwei Teile gegliedert. Der erste Teil der Arbeit thematisiert die Mehrklassen-Zertifizierung. In Kapitel 2 werden zunächst die notwendigen Grundlagen bezüglich der Zertifizierung gelegt. Darunter zählen die relevanten Grundbegriffe der Statistischen Lerntheorie (2.1), sowie einige grundlegenden Prinzipien zur Validierung von Modellen im Maschinellen Lernen (2.2). Abgeschlossen wird das Kapitel mit der Zusammenfassung der Theorie aus [6], die das Fundament für die Mehrklassen-Zertifizierung bildet. Im anschließenden Kapitel 3 werden die Methoden zur Mehrklassen-Zertifizierung vorgestellt. In der abschließenden Evaluation werden die entwickelten Zertifizierungen in einer vergleichenden Evaluation analysiert und bewertet.

Im zweiten Teil der Arbeit werden zu Beginn populäre heuristische ACS-Strategien aufgezeigt (Kapitel 6). Anschließend werden in Kapitel 7 Strategien vorgestellt, die die Domänenlücke beachten und minimieren. In der nachfolgenden Evaluation wird die Leistungsfähigkeit der diskutierten ACS-Strategien unter wechselnden Deployment-Bedingungen analysiert und abschließend eingeschätzt.

Teil I

Mehrklassen-Zertifizierung

Kapitel 2

Modell-Zertifizierung

Dieses Kapitel befasst sich weitergehend mit dem Themenfeld der Zertifizierung von maschinellen Lernmodellen. Da die hier vorgestellten Zertifizierungen auf den Grundlagen der Statistischen Lerntheorie gründen, werden zunächst die relevanten Konzepte der Lerntheorie betrachtet und Teile der Notation eingeführt. Anschließend werden Grundprinzipien der Modellvalidierung im maschinellen Lernen skizziert, wobei die Modell-Robustheit in einem ACS-Framework formalisiert wird. In 2.4 und 2.5 wird die auf PAC-Schranken basierende Theorie zur Zertifizierung von Lernmodellen [6] dargestellt, die bereits Zertifizierungen für binäre Klassifikationsaufgaben ermöglicht.

2.1 Statistische Lerntheorie

Die Statistische Lerntheorie ist ein beliebtes Framework im maschinellen Lernen, das eingesetzt wird, um Lernmodelle zu formalisieren und zu verstehen [22]. Insbesondere das überwachte Lernen ist gut erforscht und kann aus der Perspektive der Lerntheorie analysiert werden [3] [15]. Im Nachfolgenden wird ein Klassifikationsproblem mit N Klassen durch die Statistische Lerntheorie formalisiert. Die hier verwendete Notation orientiert sich an [21].

Sei $\mathcal{X}$ die Menge der möglichen Datenbeispiele und $\mathcal{Y} = \{1, ..., N\}$ die möglichen Klassenlabel. Eine Hypothese $h : \mathcal{X} \to \mathcal{Y}$ ist eine Funktion, die für jedes Datenbeispiel $x \in \mathcal{X}$ eine Vorhersage $h(x)$ liefert. Die Begriffe Hypothese, Modell und Klassifikator werden im weiteren Verlauf gleichbedeutend für h verwendet. Die Trainingsmenge D besteht aus m gelabelten Trainingsbeispielen $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. Es wird angenommen, dass die Trainingsinstanzen aus einem datenerzeugenden Prozess stammen, der eine Wahrscheinlichkeitsdichtefunktion $\mathcal{S}$ über den gelabelten Datenraum $\mathcal{X} \times \mathcal{Y}$ induziert. Demnach wird die Trainingsmenge D durch das Ziehen von m Datenbeispielen aus $\mathcal{S}$ generiert, das heißt:

$$D = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq m\} \sim \mathcal{S}^m \tag{2.1}$$

Eine zentrale Annahme bei der Ziehung der Trainingsbeispiele aus $\mathcal{S}$ ist, dass diese unabhängig und identisch verteilt **(i.i.d.)** sind. Für ein Datentupel $(x, y) \in \mathcal{X} \times \mathcal{Y}$ bewertet eine Verlustfunktion (engl. *loss*) $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ das durch die Hypothese h prognostizierte Label $h(x)$ mit dem tatsächlichen (wahren) Label y. Eine elementare Verlustfunktion ist das *zero-one loss* definiert durch

$$\ell_{01}(h(x), y)) = \begin{cases} 0 & \text{Wenn } h(x) = y \\ 1 & \text{Wenn } h(x) \neq y \end{cases} \tag{2.2}$$

Eine Risikofunktion 2.3 beschreibt den erwarteten Verlust eines Modells h unter einer *loss*-Funktion ℓ über $\mathcal{X} \times \mathcal{Y}$.

$$L_{\mathcal{S}}(h) = \mathbb{E}_{(x,y)\sim\mathbb{P}_{\mathcal{S}}}[\ell(y, h(x))] \tag{2.3}$$

Im Allgemeinen ist das erwartete Risiko unbekannt und kann nur durch das empirische Risiko $L_D(h)$ mit

$$L_D(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(y_i, h(x_i)) \tag{2.4}$$

approximiert werden, wobei auf Grundlage der bekannten Trainingsdaten der Verlust des Modells unter ℓ evaluiert wird.

Die grundlegende Lernaufgabe besteht im Finden eines Modells h, das die Risikofunktion 2.3 minimiert. Da das tatsächliche Risiko nur durch das empirische Risiko geschätzt werden kann, wird bei der empirischen Risikominimierung (*ERM*) eine Funktion h aus einer Hyphothesenklasse $\mathcal{H}$ gesucht, die das empirische Risiko 2.4 minimiert:

$$ERM_{\mathcal{H}}(D) \in \arg\min_{h\in\mathcal{H}} L_D(h) \tag{2.5}$$

Das durch die empirische Risikominimierung gefundene Modell $h^* = ERM_{\mathcal{H}}(D)$ wurde ausschließlich auf Grundlage der Trainingsmenge D selektiert und evaluiert. Allerdings wird an h^* die Anforderung gestellt für beliebige Dateninstanzen aus $\mathcal{X} \times \mathcal{Y}$ akkurate Prognosen zu liefern, das bedeutet 2.3 zu minimieren. Folglich ist die Fähigkeit zur Generalisierung das zentrale Leistungsmerkmal von Modellen im maschinellen Lernen.

Die Statistische Lerntheorie kann Aussagen über die zu erwartende Generalisierbarbeit von Modellen liefern, die über *ERM* gefunden wurden. Auf Grundlage der **(i.i.d.)** Annahme bei der Ziehung der Trainingsbeispiele sind *PAC*-Fehlerschranken herleitbar.

2.1.1 Proposition (i.i.d. PAC-Fehlerschranke [21]). *Für jedes $\epsilon > 0$ und jedes feste $h \in \mathcal{H}$ gilt mit einer Wahrscheinlichkeit von mindestens $1 - \delta$, wobei $\delta = 2e^{-2m\epsilon^2}$, dass:*

$$|L_D(h) - L_{\mathcal{S}}(h)| \leq \epsilon$$

Nach 2.1.1 lässt sich die maximale absolute Abweichung zwischen dem empirischen Risiko L_D und dem wahren Risiko $L_{\mathcal{S}}$ durch ϵ begrenzen, wobei die Aussage mit einer Wahrscheinlichkeit von $1 - \delta$ gilt. Neben der Wahl von δ wird die Größe von ϵ durch die Anzahl der

Trainingsbeispiele m bestimmt. Somit schränkt ϵ den erwarteten Schätzfehler ein, der sich aus der Ziehung $D \sim \mathcal{S}^m$ ergibt. Die PAC-Schranke in 2.1.1 lässt sich, wie in [6] gezeigt, ebenfalls in asymmetrische PAC-Schranken 2.1.2 zerlegen.

2.1.2 Korollar (asymmetrische i.i.d PAC-Fehlerschranke [6]). *Für jedes* $\epsilon^{(l)}, \epsilon^{(u)} > 0$ *und für beliebiges aber fester* $h \in \mathcal{H}$*, jeder der folgenden Schranken gilt mit einer Wahrscheinlichkeit von mindestens* $1 - \delta^{(i)}$*, wobei* $\delta^{(i)} = e^{-2m(\epsilon^{(i)})^2}$ *und* $i \in \{l, u\}$*:*

a) $L_D(h) - L_{\mathcal{S}}(h) \leq \epsilon^{(l)}$

b) $L_{\mathcal{S}}(h) - L_D(h) \leq \epsilon^{(u)}$

2.2 Prinzipien der Modellvalidierung

Die Bedeutsamkeit das Leistungsverhalten von trainierten Modellen im Kontext der Modellsicherheit zu evaluieren, wurde bereits diskutiert. Das Ziel der Validierung ist es, die Generalisierungsfähigkeit eines trainierten Modells unter bestimmten Anwendungsbedingungen zu bewerten. Im Rückbezug zur Statistische Lerntheorie entspricht dies dem Szenario, dass für eine Hypothese h die unbekannte Risikofunktion $L_{\mathcal{S}}(h)$ geschätzt werden muss. Im maschinellen Lernen gründet die Modellvalidierung auf einigen etablierten Prinzipien, die im Folgendem vorgestellt werden.

2.2.1 Aufteilung des Datensatzes

Sei $D = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq m\} \sim \mathcal{S}^m$ der gegebene Datensatz. Ein zentrales Prinzip bei der Validierung von Modellen lautet, dass Dateninstanzen die für das Optimieren von h^* verwendet wurden, nicht für die Testung des Generalisierungsfehlers benutzt werden dürfen. Daraus folgt, dass D mindestens in zwei disjunkte Datenmengen D_{trn} und D_{tst} aufgeteilt werden muss. Während der Trainingsphase wird ein Modell h auf Basis der Trainingsmenge D_{trn} optimiert. Der Testfehler wird unter der Evaluation einer Güte anhand der gelabelten Testdaten approximiert.

Die Plausibilität der Aufteilung von D in sich nicht überschneidende Datensätze wird deutlich, wenn man das Problem der Überanpassung von Modellen an Daten (engl. *overfitting*) betrachtet. Beim *overfitting* werden Modelle zu stark an den gegebenen Daten angepasst, wodurch die Generalisierungsfähigkeit stark beeinträchtigt wird. Es ist charakteristisch für überangepasste Modelle, dass der ermittelte Trainingsfehler deutlich kleiner ist als der Testfehler. Wird D nicht in disjunkte Teilmengen separiert, ist es nicht möglich *overfitting* zu erkennen, wodurch die tatsächliche Leistungsfähigkeit eines Modells stark überschätzt wird.

Neben einer Aufteilung von D in Trainings- und Testmengen, ist eine zusätzliche Teilung in eine Validierungsmenge populär, so dass $D_{\text{trn}} \cap D_{\text{val}} \cap D_{tst} = \emptyset$ gilt. Dies bietet

sich im Kontext der Modell-Selektion an, wenn Modelle auf Basis von Hyperparameteroptimierungen ausgewählt werden. Hierbei wird die Güte einer Modellkonfiguration auf Grundlage der Validierungsdaten bestimmt. Auf diesem Weg, kann das leistungsstärkste Modell aus einer Auswahl von Hyperparameterkonfigurationen selektiert werden. Abschließend wird die Leistungsfähigkeit des ausgewählten Modells einmalig auf den Testdaten evaluiert. Durch das zusätzliche Aufteilen in D_{val} und das einmalige Testen auf D_{tst} wird die Wahrscheinlichkeit reduziert, dass *overfitting* während der Validierung nicht erkannt wird beziehungsweise dass die Optimierung der Hyperparameter zur Überanpassung führt.

Eine Erweiterung dieses Vorgehens stellt die Kreuzvalidierung dar. Bei einer k-fachen Kreuzvalidierung wird D in k Datenblöcken mit $D = \{D_1, \ldots, D_k\}$ eingeteilt, die jeweils ungefähr die gleiche Anzahl an Dateninstanzen umfassen. Insgesamt werden k Durchläufe gestartet, in denen ein Modell auf $k-1$ Datenblöcken trainiert wird, während die Leistungsfähigkeit durch den verbliebenen Datenblock geschätzt wird. Es werden demnach k Modelle erzeugt, die auf unterschiedlich zusammengesetzten Datenblöcken trainiert werden. Der approximative Generalisierungsfehler wird durch die Mittelung der gesammelten Testfehler ermittelt.

Die Kreuzvalidierung existiert in einigen Varianten, die sich hauptsächlich durch die Zusammensetzungen der Datenblöcke unterscheiden. So bestehen bei der stratifizierten k-fachen Kreuzvalidierung die k Datenblöcke aus der annähernd gleichen Klassen-Verteilung. Dies hat den Vorteil, dass die Varianz der Fehlerschätzungen verringert wird. Um die Robustheit der Ergebnisse weiter zu erhöhen, können die Kreuzvalidierungsverfahren wiederholt angewendet werden.

2.2.2 Modellgüte

Der Testfehler muss unter einer Modellgüte evaluiert werden. Klassifikationsmodelle werden häufig nach der *accuracy* (*acc*) bewertet, die den Anteil von korrekt vorhergesagten Datenbeispielen abbildet. Zwischen der *accuracy* und der Evaluation unter einer *zero-one-loss* Funktion besteht durch

$$acc(\hat{y}, y) = 1 - \frac{1}{n} \sum_{i=1}^{n} \ell_{01}(\hat{y}, y)) \tag{2.6}$$

ein direkter Zusammenhang.

Die *accuracy* als alleinige Kennzahl der Generalisierungsfähigkeit kann zu verfälschten Einschätzungen führen. So hängt der erreichte *acc*-Fehler stark von der Klassenzusammensetzung der Testdaten ab. Wird die Anzahl an Klasseninstanzen durch eine Klasse dominiert, so dominiert die Erkennungsrate der Klasse den erzielten *acc*-Wert. Das Setzen von Gewichten, die die Anzahl an Klasseninstanzen reflektieren, kann eine Lösung bei der Fehlerevaluation von unbalancierten Klassifikationsproblemen sein. Ebenfalls kann durch inspizieren von Konfusionsmatrizen die Fehlerrate nach Klassen aufgeschlüsselt werden.

2.2.3 Robustheit

Damit valide Aussagen über das Leistungsverhalten eines Modelles gewonnen werden kann, ist eine strukturierter experimenteller Aufbau notwendig. Das Design der Experimente entscheidet über die Aufteilung der Daten, sowie über eine angemessene Wahl der Modellgüte. Des weiteren muss eine Datenauswahl stattfinden, die das Anwendungsumfeld repräsentativ abbildet.

Die vorgestellten Prinzipien zur Validierung beruhen auf der Annahme, dass die Bedingungen zwischen dem Training eines Modells, sowie der Anwendung identisch sind. Allerdings ist in vielen Anwendungen diese Annahme in ihrer Absolutheit nicht haltbar, da Anwendungsbedingungen Veränderungen unterworfen sind beziehungsweise nicht voll umfänglich bekannt sind. Folglich ist das Modellverhalten unter wechselnden Bedingungen ein wichtiger Leistungsaspekt, der die Robustheit eines Modells adressiert.

2.3 Modell-Robustheit im Kontext von ACS

Die *domain adaption* [10] ist ein mit dem *transfer learning* assoziierter Forschungsbereich [18]. Eine Domäne (engl. *domain*) kann als Wahrscheinlichkeitsdichtefunktion über einen gelabelten Datenraum $\mathcal{X} \times \mathcal{Y}$ aufgefasst werden. Entsprechend beschreibt die *domain* eine Umgebung die durch einen datengenerierenden Prozess induziert wurde. Die *domain adaption* adressiert das Szenario, dass ein trainiertes Modell in einer Anwendungsumgebung (*Deployment*) eingesetzt wird, welches sich von der Trainingsumgebung unterscheidet. Die divergierenden Verteilungen der Deploymentdaten (*domain shift*, *distributional shift*) kann eine Herausforderung für die Generalisierbarbeit von Modellen darstellen. Zudem stellen sich Fragen zum Leistungsverhalten von Modellen, die unter wechselnden oder unbekannten Deployment-Bedingungen eingesetzt werden.

Die Statistische Lerntheorie liefert mit PAC-Schranken hilfreiche Instrumente um die Leistungsfähigkeit von Modellen einzuschätzen. Allerdings wird bei der Fehlerabschätzung wie in 2.1.1 davon ausgegangen, dass der zugrundeliegende datengeneriende Prozess jeweils für das Training und dem Deployment identisch ist. Sei $\mathcal{S}$ die *domain* in der ein Modell h^* auf Grundlage von $D \sim \mathcal{S}^m$ trainiert wurde und $\mathcal{T}$ die *domain* in der h^* eingesetzt wird. Falls die Deploymentumgebung gegeben durch $\mathcal{T}$ von der Trainingsumgebung abweicht, dass heißt $\mathcal{S} \neq \mathcal{T}$ gilt, wird der absolute Fehler mit

$$|L_{\mathcal{D}}(h) - L_{\mathcal{T}}(h)| \tag{2.7}$$

durch die absolute Abweichung zwischen den empirischen Risiko unter $\mathcal{S}$ und dem wahren Risiko unter $\mathcal{T}$ beschrieben. Da D nicht aus der $\mathcal{T}$ gezogen wurde, ist die PAC-Schranke in 2.1.1 nicht für die Fehlerabschätzung von 2.7 anwendbar.

Werden maschinelle Lernprobleme aus dem ACS-Framework betrachtet, so existiert ein klassenbedingter Datengenerator, der Dateninstanzen nach der Vorschrift $\mathcal{Y} \to \mathcal{X}$ unter

Kosten erzeugt. Hierdurch werden während des Trainings aktiv Klassenverhältnisse gewählt, die sich möglichst günstig auf die Akquisekosten und die Modellperformanz auswirken [13]. Im Folgendem werden ausschließlich Modelle betrachtet, die unter ACS trainiert wurden. Dabei gilt die Annahme, dass die Daten aus den Domänen $\mathcal{S}$ und $\mathcal{T}$ dem gleichen klassenbedingten Mechanismus folgen.

2.3.1 Definition (Annahme eines identischen Mechanismus [7]). Alle Daten in den Domänen $\mathcal{S}$ und $\mathcal{T}$ werden unabhängig voneinander durch denselben klassenbedingten Mechanismus erzeugt.

$$\mathbb{P}_{\mathcal{S}}(X = x \mid Y = y) = \mathbb{P}_{\mathcal{T}}(X = x \mid Y = y) \quad \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \tag{2.8}$$

Die Annahme des identischen Mechanismus ist in der Literatur ebenfalls als *prior probability shift* [17] beziehungsweise *target shift* [27] bekannt. Diese Annahme hat die Konsequenz, dass die Abweichung zwischen den Domänen $\mathcal{S}$ und $\mathcal{T}$ ausschließlich durch Verschiebungen der Klassenverteilungen $\mathbb{P}_{\mathcal{S}}(y) \neq \mathbb{P}_{\mathcal{T}}(y)$ resultiert. Die hier betrachtete Robustheitseigenschaft besteht in der Fragestellung, welche Konsequenzen eine divergierende Label-Verteilung auf die Perfomanz eines trainierten Modells besitzt. Das heißt, für eine gegebene Verschiebung $\mathbb{P}_{\mathcal{S}}(y) \neq \mathbb{P}_{\mathcal{T}}(y)$ ist die *domain*-induzierte abweichende Lücke in der Modell-Performanz das zentrale Untersuchungsobjekt.

Prinzipiell können Fragen zur Modell-Robustheit in ACS beantwortet werden, indem beliebig verteilte Testdaten durch den Generator produziert werden. Allerdings würde dies einen immensen Aufwand erfordern, der mit hohen Akquisekosten verbunden wäre. Dementsprechend stellt das produzieren von beliebigen Testmengen keinen praktikablen Weg dar, die Robustheit eines Modells zu evaluieren.

Auf Grundlage der Informationstheorie wurde die Domänenlücke (engl. *domain-gap*) unter dem ACS-Framework bereits qualitativ untersucht [7]. Die zentrale Beobachtung war, dass mit steigender Anzahl an ACS-generierten Trainingsdaten, jedes von $\mathbb{P}_{\mathcal{T}}(Y)$ divergierende Klassenverhältnis einen schwindenden Nutzen bietet. Dies liefert den Hinweis, dass ACS-Strategien die Klassenverhältnisse im Deployment reflektieren sollten. Allerdings sind die Erkenntnisse aus [7] aufgrund ihrer qualitativen Natur limitiert. So konnte ausschließlich das Grenzwertverhalten für $m \to \infty$ bezüglich der Datenakquisition analysiert werden. Folglich bestand die Motivation, die *domain-gap* aus einer quantitativen Perspektive zu untersuchen.

2.4 Quantitative Charakterisierung der Domänenlücke

Das Theorem 2.4.1 beschreibt die zentrale Beobachtung aus [6]. Dies besagt, dass der absolute Fehler $|L_{\mathcal{T}}(h) - L_{\mathcal{D}}(h)|$ durch die absolute Domänenübergreifende Lücke (*inter-domain-gap*) $|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)|$ unter Berücksichtigung von PAC-Schranken begrenzt wird.

Unter zunehmenden ACS-generierten Trainingsdaten, nähert sich der absolute Fehler kontinuierlich der *inter-domain-gap* an, sodass der absolute Fehler durch die Differenz der Modell-Perfomanz aus $\mathcal{S}$ und $\mathcal{T}$ dominiert wird. Ein Klassifikator h, der für die Trainingsdaten D optimiert wurde, kann für große Datenmengen m nur dann genaue Ergebnisse in $\mathcal{T}$ liefern, wenn der Abstand zwischen $\mathcal{S}$ und $\mathcal{T}$ klein ist. Dabei ist $|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)|$ konstant in Bezug auf die Trainingsmenge D und damit unabhängig von ϵ, δ und m.

2.4.1 Theorem (Identische Mechanismus Schranke [6]). *Für jedes $\epsilon > 0$ und für jedes feste $h \in \mathcal{H}$, hält mit einer Wahrscheinlichkeit von $1 - \delta$, wobei $\delta = 4e^{-2m\epsilon^2}$, dass:*

$$|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| - \epsilon \leq |L_{\mathcal{T}}(h) - L_{\mathcal{D}}(h)| \leq |L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| + \epsilon \tag{2.9}$$

Die Aussage aus dem Theorem 2.4.1 wird durch Theorem 2.4.2 auf endliche Hypothesenklassen erweitert [6].

2.4.2 Theorem (Identische Mechanismus Schranke; endliche Hypothesenklasse [6]). *Mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ mit $\delta = 4|\mathcal{H}|e^{-2m\epsilon^2}$, gelten die oberen und unteren Schranken von 2.4.1 für alle $h \in \mathcal{H}$.*

In [6] wurde der Fehler $L(h)$ in Hinblick auf die Domänenlücke weitergehend analysiert. So lässt sich unter der Annahme von 2.3.1 $L(h)$ in vom Label-abhängigen Verluste $\ell_X(h, y)$ faktorisieren, die über den gesamten Merkmalsraum $\mathcal{X}$ marginalisiert werden.

$$\begin{aligned} L(h) &= \int_{\mathcal{Y}} \int_{\mathcal{X}} \mathbb{P}(x, y)\ell(y, h(x))\ dx\ dy \\ &= \int_{\mathcal{Y}} \mathbb{P}(y) \underbrace{\int_{\mathcal{X}} \mathbb{P}(x \mid y)\ell(y, h(x))\ dx}_{=\ \ell_X(h,y)}\ dy \end{aligned} \tag{2.10}$$

Durch Einsetzten von $\ell_X(h, y)$ ergibt sich die *inter-domain-gap* zu

$$|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| = \left| \int_{\mathcal{Y}} \mathbb{P}_{\mathcal{T}}(y)\ell_X(h, y)dy - \int_{\mathcal{Y}} \mathbb{P}_{\mathcal{S}}(y)\ell_X(h, y)dy \right| \tag{2.11}$$

Für Klassifikationsprobleme mit N Klassen lässt sich die Darstellung von 2.11 vereinfachen. Seien $\mathbf{p}_{\mathcal{S}}$, $\mathbf{p}_{\mathcal{T}} \in [0, 1]^N$ mit den Komponenten $[\mathbf{p}_\bullet]_i = \mathbb{P}_\bullet(Y = i)$ Vektoren, die die Wahrscheinlichkeiten der Klassen in den jeweiligen Domänen $\mathcal{S}$ und $\mathcal{T}$ repräsentieren. Des weiteren wird mit $\boldsymbol{\ell}_h \in \mathbb{R}^N_+$ ein Vektor definiert, der durch $[\boldsymbol{\ell}_h]_i = \ell_X(h, i)$ die klassenbedingten Verluste repräsentiert. Folglich ergibt sich der Verlust unter h durch $L_\bullet(h) = \sum_{i \in \mathcal{Y}} [\mathbf{p}_\bullet]_i [\ell_h]_i = \langle \mathbf{p}_\bullet, \ell_h \rangle$ und somit ist durch

$$|L^{\text{clf}}_{\mathcal{T}}(h) - L^{\text{clf}}_{\mathcal{S}}(h)| = |\langle \mathbf{p}_{\mathcal{T}}, \boldsymbol{\ell}_h \rangle - \langle \mathbf{p}_{\mathcal{S}}, \boldsymbol{\ell}_h \rangle| \tag{2.12}$$

die Domänenlücke für Klassifikationsprobleme gegeben. Ausgehend von 2.12 wurde in [6] gezeigt, dass für binäre Klassifikationsprobleme die Darstellung der Domänenlücke weiter vereinfacht werden kann. Sei $\mathcal{Y} = \{1, 2\}$ mit $\mathbb{P}_\bullet(Y = 1) = p_\bullet$ und $\mathbb{P}_\bullet(Y = 2) = 1 - p_\bullet$, dann wird mit

$$|L^{\text{bin}}_{\mathcal{T}}(h) - L^{\text{bin}}_{\mathcal{S}}(h)| = |(p_{\mathcal{T}} - p_{\mathcal{S}}) \cdot (\ell_X(h, 2) - \ell_X(h, 1))| = \Delta p \cdot \Delta \ell_X \tag{2.13}$$

die Domänenlücke für binäre Klassifikationsprobleme repräsentiert. Dementsprechend hängt die Größe der Domänenlücke von zwei Faktoren Δp und $\Delta \ell_X$ ab, wobei Δp die absolute Differenz zwischen den binären Klassenverhältnissen aus $\mathcal{S}$ und $\mathcal{T}$ beschreibt und $\Delta \ell_X$ die absolute Differenz zwischen den Klassenverlusten $\Delta \ell_X$ reflektiert.

2.5 Zertifizierung von binären Klassifikationsproblemen

Die quantitativen Erkenntnisse bezüglich der Domänenlücke können verwendet werden, um ein trainiertes Modell für Klassenverhältnisse zu zertifizieren, die im Deployment einen beschränkten *domain*-induzierten Fehler aufweisen. Die Zertifikate werden in Abhängigkeit zu δ und zum tolerierten Fehler ϵ erstellt, wobei der Anwender δ und ϵ gemäß den eigenen Anforderungen wählen kann.

2.5.1 Definition (Zertifizierte Hypothese). Eine Hypothese $h \in \mathcal{H}$ ist (ϵ, δ)-zertifiziert für alle Klassenproportionen in der Menge $\mathcal{P} \subseteq [0, 1]^N$, falls mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ und $\epsilon, \delta > 0$:

$$|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| \leq \epsilon \quad \forall \mathbf{p}_{\mathcal{T}} \in \mathcal{P}$$

Nach 2.5.1 beschreibt $\mathcal{P}$ eine Menge von Klassenverhältnissen für die ein trainiertes Modell gemäß ϵ und δ zertifiziert wurde. Für ein (ϵ, δ)-zertifiziertes Modell bedeutet dies, dass für jedes Klassenverhältnis aus $\mathcal{P}$ der durch $\mathbb{P}_{\mathcal{S}}(y) \neq \mathbb{P}_{\mathcal{T}}(y)$ verursachte Fehler durch ϵ mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ beschränkt wird. Somit kann durch die Wahl von ϵ und δ Anforderungen an *sichere* Klassenverhältnisse durch den Anwender spezifiziert werden.

Für die binäre Klassifikation lässt sich $\mathcal{P}$ durch das Intervall $[p^{\min}_{\mathcal{T}}, p^{\max}_{\mathcal{T}}]$ von zulässigen Klassenverhältnissen beschreiben. Unter Berücksichtigung von 2.13 kann $[p^{\min}_{\mathcal{T}}, p^{\max}_{\mathcal{T}}]$ bestimmt werden durch das größte Δp^* für das gilt

$$\Delta p \cdot \Delta \ell_X \leq \epsilon \quad \forall \Delta p \leq \Delta p^* \tag{2.14}$$

Für ein gegebenes ϵ ergibt sich Δp^* zu $\frac{\epsilon}{\Delta \ell_X}$. Allerdings sind die wahren klassenweise-Verluste $\ell_X(h, y)$ eines Modells h in $\mathcal{X} \times \mathcal{Y}$ nicht bekannt und müssen empirisch durch

$$\hat{\ell}_X(h, y) = \frac{1}{m_y} \sum_{i: y_i = y} \ell(y, h(x_i)) \tag{2.15}$$

aus D geschätzt werden. Unter Einsetzung von 2.15 ergibt sich die empirische Schätzung von $\Delta\ell_X$ zu

$$\Delta\hat{\ell}_X = |\hat{\ell}_X(h,1) - \hat{\ell}_X(h,2)| \tag{2.16}$$

welche aufgrund des zufälligen Ziehens von D aus $\mathcal{S}$ jedoch einem Fehler unterliegt.

2.5.1 Berechnung von optimierten Schranken für $\Delta\hat{\ell}$

Damit Δp^* aus den empirischen Daten ermittelt werden kann, muss eine kleinste obere Schranke $\Delta\ell_X^*$ für $\Delta\hat{\ell}_X$ gefunden werden, sodass Δp^* mit $\frac{\epsilon}{\Delta\ell_X^*}$ abgeschätzt werden kann. Gemäß der Definition 2.5.1 muss $\Delta\ell_X^*$ mit einer Wahrscheinlichkeit von mindestens $1-\delta$ korrekt sein, das heißt $\Delta\hat{\ell}_X$ nach oben beschränken. Das Finden der kleinsten oberen Schranke wurde in [6] wie folgt gelöst.

Unter Ausnutzung der asymmetrischen PAC-Grenzen in 2.1.2, kann für jede Klasse $y \in \{1,2\}$ $\Delta\hat{\ell}_X = |\hat{\ell}_X(h,1) - \hat{\ell}_X(h,2)|$ mit einem maximalen unteren Fehler $\epsilon_y^{(l)} > 0$ und maximalen oberen Fehlern $\epsilon_y^{(u)} > 0$ assoziiert werden, die mit einer Wahrscheinlichkeit von $1-\delta_y^{(l)}$ beziehungsweise $1-\delta_y^{(u)}$ nicht überschritten werden. Für ein gegebenes Wahrscheinlichkeitsbudget von δ kann $\epsilon_y^{(l)}$ und $\epsilon_y^{(u)}$ gewählt werden, sodass obere Schranken gefunden werden, die mit einer Wahrscheinlichkeit von mindestens $1-\delta$ korrekt sind.

Im Folgendem wird angenommen, dass $\hat{\ell}_X(h,2) \geq \hat{\ell}_X(h,1)$ gilt. Da im Fall von $\hat{\ell}_X(h,2) \leq \hat{\ell}_X(h,1)$ ein vertauschen der beiden Klassen möglich ist, gilt diese Annahme ohne Beschränkung der Allgemeinheit. Das Finden der kleinsten oberen Schranke wird charakterisiert durch das gleichzeitige Minimieren von ϵ_1 und ϵ_2 innerhalb eines Wahrscheinlichkeitsbudget $\delta = \delta_1 + \delta_2 - \delta_1\delta_2$. Siehe dazu Abbildung 2.1. Die Werte von δ_1 und δ_2 können durch $\delta_y = \delta_y^{(l)} + \delta_y^{(u)} - \delta_y^{(l)}\delta_y^{(u)}$ und $\delta_y^{(i)} = e^{-2m_y(\epsilon_y^{(i)})^2}$ ermittelt werden, wobei gilt:

$$-\underbrace{(\hat{\ell}_X(h,2) - \hat{\ell}_X(h,1) + \epsilon_1)}_{=\epsilon_2^{(l)}} \leq \ell_X(h,2) - \hat{\ell}_X(h,2) \leq \underbrace{\epsilon_2}_{=\epsilon_2^{(u)}}$$

und

$$-\underbrace{(\hat{\ell}_X(h,2) - \hat{\ell}_X(h,1) + \epsilon_2)}_{=\epsilon_1^{(u)}} \leq \hat{\ell}_X(h,1) - \ell_X(h,1) \leq \underbrace{\epsilon_1}_{=\epsilon_1^{(l)}}$$

Im Rahmen der Optimierung werden strikte Ungleichungen durch nicht-strikte Ungleichungen mit einem hinreichend kleinen τ realisiert:

$$\min_{\epsilon_1,\epsilon_2\in\mathbb{R}} \epsilon_1 + \epsilon_2, \quad \text{u. d. N.} \begin{cases} \epsilon_1, \epsilon_2 & \geq \tau \\ \delta - (\delta_1 + \delta_2 - \delta_1\delta_2) & \geq 0 \end{cases} \tag{2.17}$$

Sei $(\epsilon_1^*, \epsilon_2^*)$ der Fehler, der das Optimierungsproblem in 2.17 minimiert. Dann bildet sich die kleinste obere Schranke durch $\Delta\ell_X^* = ((\hat{\ell}_X(h,2) + \epsilon_2^*) - (\hat{\ell}_X(h,1) + \epsilon_1^*))$, die mit einer Wahrscheinlichkeit von mindestens $1-\delta$ gilt.

Falls wenige Daten zur Schätzung von $\Delta\ell_X^*$ zur Verfügung stehen, kann die beidseitige Optimierung durch ϵ_1 und ϵ_2 fehlschlagen. Dieser Fall äußert sich, falls die Berechnung der unteren Schranke durch $\hat{\ell}_X(h,1) - \epsilon_1$ im negativen Wertebereich liegt, wodurch die grundlegende Eigenschaft $\ell(h,y) \geq 0$ verletzt wird. Falls die beidseitige Optimierung nicht realisierbar ist, kann auf eine einseitige Schätzung zurückgegriffen werden, in der nur die jeweiligen oberen Schranken minimiert werden, die von ϵ_2 abhängen. Das einseitige Optimieren der oberen Schranken $\epsilon_y^{(u)}$ ist somit für beliebig kleine Datensätze anwendbar. Allerdings fallen die Schätzungen durch das einseitige Optimieren, im Vergleich zur Optimierung in 2.17, gröber aus.

Analog können durch die asymmetrischen PAC-Schranken mit $\delta_y = e^{-2m_y\epsilon_y^2}$ ebenfalls die größte untere Schranke für $\Delta\ell_X$ durch $((\hat{\ell}_X(h,2) - \epsilon_2^*) - (\hat{\ell}_X(h,1) - \epsilon_1^*))$ ermittelt werden. Hierbei resultieren $(\epsilon_1^*, \epsilon_2^*)$ durch das Lösen der Optimierungsaufgabe in 2.18.

$$\min_{\epsilon_1,\epsilon_2\in\mathbb{R}} \epsilon_1 + \epsilon_2, \quad \text{u. d. N.} \begin{cases} \epsilon_1, \epsilon_2 & \geq \tau \\ \Delta\hat{\ell}_X - (\epsilon_1 + \epsilon_2) & \geq 2\tau \\ \delta - (\delta_1 + \delta_2 - \delta_1\delta_2) & \geq 0 \end{cases} \tag{2.18}$$

Für die binäre Zertifizierung nach [6] sind nur die kleinsten oberen Schranken relevant. Die maximale untere Schranke wird jedoch für die Mehrklassen-Zertifizierung in Kapitel 3.3 aufgegriffen.

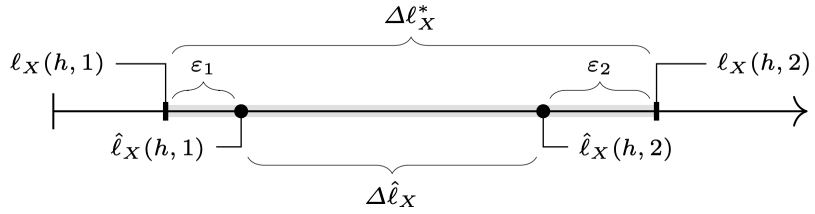


Abbildung 2.1: Abbildung entnommen aus [6]. In Abhängigkeit zu δ wird der Fehler ϵ_1 und ϵ_2 minimiert, damit die minimale obere Schranke $\Delta\ell_X^*$ bestimmt werden kann. Dies hat zur Folge, dass $\hat{\ell}_X(h,1)$ höchstens um ϵ_1 verringert wird, während $\hat{\ell}_X(h,2)$ maximal um ϵ_2 vergrößert wird.

Kapitel 3

Zertifizierung von Mehrklassen-Klassifikationsprobleme

Wie bereits dargelegt, können binäre Klassifikatoren für eine Menge an Klassenverhältnissen zertifiziert werden. Ausgestellte (ϵ, δ)-Zertifikate für binäre Klassifikatoren spiegeln PAC-Schranken wider, deren Anforderungen vom Benutzer individuell festgelegt werden. Die Beschränkung auf binäre Klassifikationsprobleme ist jedoch eine erhebliche Einschränkung im Hinblick auf eine universellere Anwendbarkeit der Zertifizierung, da in zahlreichen Anwendungsszenarien Klassifikationsprobleme betrachtet werden, die sich nicht auf binäre Klassen beschränken lassen. Auf Grundlage der Theorie zur Quantisierung der Domänenlücke [6], werden im Folgenden Methoden zur Zertifizierung von Mehrklassen-Klassifikationsproblemen entwickelt.

3.1 Problembeschreibung

Für binäre Klassifikationsproblemen kann die Domänenlücke auf $\Delta p \cdot \Delta \ell_X$ reduziert werden. Dies ermöglicht das Finden von kleinsten oberen Schranken, die gemäß den Zertifizierungsanforderungen spezifiziert wurden. Mit dem Erhalt von kleinsten oberen Schranken konnte die Zertifizierung erfolgreich durchgeführt werden und die Menge der sicheren Klassenverhältnisse durch ein Intervall beschrieben werden.

Zunächst muss bei der Mehrklassen Zertifizierung entschieden werden, wie die Menge $\mathcal{P}$ charakterisiert werden soll. Approximativ durch eine Menge von Datenbeispielen aus $\mathcal{P}$? Als Abweichung von $\mathbf{p}_{\mathcal{S}}$, gemäß einer bestimmten Norm? Welche Normen eignen sich? Diese unterschiedlichen Charakterisierungen führen zu verschiedenen Zertifizierungsstrategien, welche in 3.3 und 3.4 vorgestellt werden.

Die zentrale Herausforderung bei der Erweiterung der Zertifizierung auf Klassifikationsproblemen mit beliebig vielen Klassen besteht im Berechnen der *inter-domain-gap*. Analog zum binären Fall ist der tatsächliche Verlust von h unter den Domänen nicht bekannt und

kann nur empirisch aus der Trainingsmenge D geschätzt werden. Folglich muss eine Abschätzung für $|L^{\text{clf}}_{\mathcal{T}}(h) - L^{\text{clf}}_{\mathcal{S}}(h)| = |\langle \mathbf{p}_{\mathcal{T}}, \boldsymbol{\ell}_h \rangle - \langle \mathbf{p}_{\mathcal{S}}, \boldsymbol{\ell}_h \rangle|$ gefunden werden, die kompatibel zur Definition der Zertifizierung in 2.5.1 ist.

3.2 Dekomposition der Domänenlücke

In Anlehnung zur Vorgehensweise bei binären Klassifikationsproblemen, wird die Domänenlücke in 3.1 in eine Repräsentation transformiert, die den Grundstein für die hier vorgestellten Mehrklassen-Zertifizierungen legt. Sei $\mathbf{p}_{\mathcal{S}}$, $\mathbf{p}_{\mathcal{T}} \in [0,1]^N$, dann lässt sich die Domänenlücke umordnen durch

$$\begin{aligned} |L^{\text{clf}}_{\mathcal{T}}(h) - L^{\text{clf}}_{S}(h)| &= |\langle \mathbf{p}_{\mathcal{T}}, \boldsymbol{\ell}_h \rangle - \langle \mathbf{p}_{\mathcal{S}}, \boldsymbol{\ell}_h \rangle| \\ &= |\langle \mathbf{p}_{\mathcal{T}} - \mathbf{p}_{\mathcal{S}}, \boldsymbol{\ell}_h \rangle| \\ &= |\langle \mathbf{d}, \boldsymbol{\ell}_h \rangle|, \end{aligned} \tag{3.1}$$

Hierbei repräsentiert $\mathbf{d}$ einen durch $\mathbf{p}_{\mathcal{T}} - \mathbf{p}_{\mathcal{S}}$ induzierten *Verschiebungsvektor*. Für alle $i \in \{1, ..., N\}$ lässt sich das Vorzeichen des jeweiligen klassenweisen Verlustes $\ell_X(h, i)$ durch $[\mathbf{p}]_i = [\mathbf{p}_{\mathcal{T}}]_i - [\mathbf{p}_{\mathcal{S}}]_i$ bestimmen. Folglich ist die Domänenlücke für beliebige Label-Verschiebungen in positive und negative Verlustbestandteile dekomponierbar.

3.3 Vorzeichenbehaftete Zertifizierung

Bislang wurden Zertifizierungen nach der Definition 2.5.1 betrachtet, wonach für ein gegebenen Fehler ϵ Klassenverhältnisse zertifiziert werden, für die die betragsmäßige *inter-domain-gap* $|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| \leq \epsilon$ ist. Aus der Perspektive der Modellrobustheit interessiert bei der Zertifizierung nur der positive *domain*-induzierte Fehler ϵ, der zu einer Verschlechterung der Modell-Performanz im Deployment führt. Falls eine Klassenverschiebung im Deployment zu einer verbesserten Performanz des Klassifikators beiträgt, wird die Modellsicherheit nicht gefährdet. Entsprechend kann die Zertifikats-Definition um vorzeichenbehaftete Hypothesen erweitert werden.

3.3.1 Definition (Vorzeichenbehaftete zertifizierte Hypothese). Eine Hypothese $h \in \mathcal{H}$ ist (ϵ, δ)-zertifiziert für alle Klassenproportionen in der Menge $\mathcal{P} \subseteq [0,1]^N$, falls mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ und $\epsilon, \delta > 0$:

$$L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h) \leq \epsilon \quad \forall \mathbf{p}_{\mathcal{T}} \in \mathcal{P}$$

Eine Zertifizierung nach 3.3.1 kann vorteilhaft sein, falls für eine Domänenverschiebung $L_{\mathcal{S}}(h) > L_{\mathcal{T}}(h)$ gilt, sodass $\epsilon < 0$ ist. Durch das Wegfallen des Betrags kann der Anwender den negativen Fehler ϵ korrekt im Kontext der Modellsicherheit einordnen.

Signed Certificate

Eine Methode, die gemäß 3.3.1 vorzeichenbehaftet zertifizieren kann, ist das nachfolgend vorgestellte *Signed Certificate*. Gegeben sei $\mathbf{p}_{\mathcal{S}}$, $\mathbf{p}_{\mathcal{T}} \in [0,1]^N$, dann wird die *inter-domain-gap* nach 3.1 durch $\langle \mathbf{d}, \boldsymbol{\ell}_h \rangle$ dekomponiert. Die Zertifizierung folgt der Grundidee, dass durch $\langle \mathbf{d}, \boldsymbol{\ell}_h \rangle$ die einzelnen Verlustkomponenten $[\mathbf{d}]_i \ell_X(h,i)$ anhand ihrer Vorzeichen gruppiert und summiert werden können.

Sei $\ell_h^{(+)}$ der positive Verlustbestandteil mit $\ell_h^{(+)} = \sum_{i=1}^N \max(0, [\mathbf{d}]_i \ell_X(h,i))$ und $\ell_h^{(-)}$ der negative Verlustbestandteil mit $\ell_h^{(-)} = \sum_{i=1}^N \min(0, [\mathbf{d}]_i \ell_X(h,i)$. Die vorzeichenbehaftete Domänenlücke $L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)$ ergibt sich aus den jeweils summierten positiven und negativen Fehleranteil durch $L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h) = \ell_h^{(+)} + \ell_h^{(-)}$.

Analog zur binären Zertifizierung ist der tatsächliche klassenweise Verlust $\ell_X(h,i)$ nicht bekannt und muss wiederum aus den gegebenen Trainingsdaten geschätzt werden. Entsprechend wird der positive beziehungsweise negative Fehleranteil aus D durch $\hat{\ell}_h^{(+)} = \sum_{i=1}^N \max(0, [\mathbf{d}]_i \hat{\ell}_X(h,i))$ und $\hat{\ell}_h^{(-)} = \sum_{i=1}^N \min(0, [\mathbf{d}]_i \hat{\ell}_X(h,i))$ ermittelt.

Damit eine Zertifizierung nach 3.3.1 erfolgen kann, muss eine Abschätzung für $\hat{\ell}_h^{(+)} + \hat{\ell}_h^{(-)}$ gefunden werden, die mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ gilt.

Fall $\hat{\ell}_h^{(+)} \geq |\hat{\ell}_h^{(-)}|$: Falls der positive empirische Fehleranteil überwiegt, gilt $\hat{\ell}_h^{(+)} - |\hat{\ell}_h^{(-)}| = |\hat{\ell}_h^{(+)} - |\hat{\ell}_h^{(-)}|| = \Delta\hat{\ell}_X$. Eine obere Schranke für $\Delta\hat{\ell}_X$, die den Anforderungen entspricht, ist durch die in 2.17 formulierte Optimierungsaufgabe gegeben, die bereits im Kontext der binären Zertifizierung eingesetzt wird.

Fall $\hat{\ell}_h^{(+)} < |\hat{\ell}_h^{(-)}|$: In diesem Fall liegt der geschätzte Fehler, der durch die Klassenverschiebung verursacht wird, im negativen Wertebereich. Es gilt, $\hat{\ell}_h^{(+)} - |\hat{\ell}_h^{(-)}| = -|\hat{\ell}_h^{(+)} - |\hat{\ell}_h^{(-)}|| = -\Delta\hat{\ell}_X$. Damit -$\Delta\hat{\ell}_X$ eine kleinste obere Schranke bildet, muss eine größte untere Schranke für $\Delta\hat{\ell}_X$ hergeleitet werden. Diese wird durch das Lösen der Optimierungsaufgabe 2.18 realisiert.

Das Berechnen von unteren Schranken $\Delta\ell^*_{(-)}$ und oberen Schranken $\Delta\ell^*_{(+)}$ durch 2.18 beziehungsweise 2.17, gestattet die Bildung von Konfidenzintervallen. Für $\hat{\ell}_h^{(+)} \geq |\hat{\ell}_h^{(-)}|$ werden die Schätzintervalle durch $[\Delta\ell^*_{(-)}, \Delta\ell^*_{(+)}]$ gebildet. Falls der negative Fehleranteil dominiert, gilt für das Vertrauensintervall $[-\Delta\ell^*_{(+)}, -\Delta\ell^*_{(-)}]$.

Somit liefert das *Signed Certificate* eine optionale Intervallschätzung des *domain*-induzierten Fehlers. Hierbei wird der tatsächliche Fehler durch die Grenzen des Intervalls umschlossen, die jeweils mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ gelten.

Da die Aufteilung in $\ell_h^{(-)}$ und $\ell_h^{(+)}$ den Vektor $\mathbf{d}$, und somit den Vektor $\mathbf{p}_{\mathcal{T}}$, als gegeben annehmen muss, lässt sich das Zertifikat nur für bestimmte Datenbeispielen aus $\mathcal{P}$ anwen-

den. Damit kann $\mathcal{P}$ nur approximativ durch eine Menge an Datenbeispielen charakterisiert werden.

3.4 Norm Certification

Die Höldersche Ungleichung [26] gehört zu den fundamentalen Ungleichungen für *Lebesgue*-Räume (L^p-Räume). Die *inter-domain-gap* kann durch die Höldersche Ungleichung nach oben abgeschätzt werden. Diese Abschätzung legt das Fundament für eine Vielzahl von Methoden, die es erlauben Zertifizierungen für Mehrklassen-Klassifikationsprobleme auszustellen. Im Gegensatz zum *Signed Certificate* können Zertifikate, die auf die Hölder-Ungleichung basieren, nur die absolute Domänenlücke $|L_{\mathcal{T}}(h) - L_S(h)|$ adressieren. Allerdings kann die Menge $\mathcal{P}$ als Abweichung von $\mathbf{p}_{\mathcal{S}}$ unter einer Norm charakterisiert werden.

3.4.1 Theorem (Höldersche Ungleichung [26]). *Sei (S, Σ, μ) ein Maßraum und $p, q \in [1, \infty]$ mit $1/p + 1/q = 1$, wobei $1/\infty = 0$. Dann gilt für alle messbaren reellen oder komplexwertigen Funktionen f und g auf S,*

$$\|fg\|_1 \leq \|f\|_p \|g\|_q. \tag{3.2}$$

Mit Anwendung der Hölder-Ungleichung ergeben sich folgende Abschätzungen:

$$|\langle \mathbf{d}, \boldsymbol{\ell}_h \rangle| \;\leq\; \begin{cases} \|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty, & \text{for } p = 1, q = \infty \\ \|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2, & \text{for } p = 2, q = 2 \\ \|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1, & \text{for } p = \infty, q = 1 \end{cases} \tag{3.3}$$

Sei $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ eine beliebige Ungleichung aus 3.3 mit konjungierten Hölder-Exponenten p und q. Um eine Hypothese im Sinne von Definition 2.5.1 zu zertifizieren, muss mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ gelten, dass für $\epsilon, \delta > 0$,

$$|L_{\mathcal{T}}(h) - L_{\mathcal{S}}(h)| \;\leq\; \|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q \;\leq\; \epsilon \quad \forall\, \mathbf{p}_{\mathcal{T}} \in \mathcal{P}. \tag{3.4}$$

Das Zerlegen der *inter-domain-gap* in $\|\mathbf{d}\|_p$ und $\|\boldsymbol{\ell}_h\|_q$, ähnelt der Faktorisierung $\Delta p \cdot \Delta \ell_X$ für binäre Klassifikationsprobleme. Gleichfalls ist nur der empirische *loss* $\hat{\ell}_h$ bekannt, sodass die alleinige Anwendung der Hölder-Ungleichung keine statistisch gerechtfertigten Aussagen liefert, die für die Zertifizierung notwendig wären. Folglich muss eine kleinste obere Schranke $\|\boldsymbol{\ell}_h\|_q^*$ für $\|\boldsymbol{\ell}_h\|_q$ gefunden werden, die mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ gilt. Für die Beschreibung der betreffenden Optimierungsaufgaben für die kleinsten oberen Schranken, siehe 3.7, 3.8 und 3.9.

Mit dem Erhalt der kleinsten oberen Schranken, kann für ein gegebenes $\epsilon > 0$ die Gleichung 3.4 nach

$$\|\mathbf{d}\|_p \;\leq\; \frac{\epsilon}{\|\boldsymbol{\ell}_h\|_q^*} \quad \forall\, \mathbf{p}_{\mathcal{T}} \in \mathcal{P}. \tag{3.5}$$

umgestellt werden. Dementsprechend wird die Menge $\mathcal{P}$ definiert durch:

$$\mathcal{P} = \left\{ \mathbf{p} \in [0,1]^N \,:\, \|\mathbf{p} - \mathbf{ps}\|_p \leq \frac{\epsilon}{\|\ell_h\|_q^*} \text{ und } \sum_{i=1}^{N} [\mathbf{p}]_i = 1 \right\} \tag{3.6}$$

Jedes Element aus $\mathcal{P}$ erfüllt die Gleichung 3.4 und stellt somit ein *sicheres* Klassenverhältnis dar. Im Gegensatz zum *Signed Certificate* wird $\mathcal{P}$ beim *Norm Certificate* nicht durch eine Menge an Datenbeispielen beschrieben, sondern ist durch einen Norm induzierten Abstand charakterisierbar.

Berechnen der kleinsten oberen Schranken $\|\ell_h\|_q^*$

Für die jeweils gestellten Optimierungsaufgaben ist ein Wahrscheinlichkeitsbudget von δ gegeben, welches durch das Minimieren von $\epsilon_1, \ldots, \epsilon_N$ aufgebraucht wird. Für ein ausreichend kleines $\tau > 0$, werden im Kontext der Optimierung strikte Ungleichung durch nicht-strikte Ungleichungen ersetzt. Mit $(\epsilon_1^*, \ldots, \epsilon_N^*)$ wird der optimierte Fehler repräsentiert, der die jeweilige Optimierungsaufgabe minimiert.

Summennorm $\|\boldsymbol{\ell}_h\|_1^*$

Die Summennorm $\|\cdot\|_1$ (auch bekannt als 1-Norm) repräsentiert für einen Vektor $\mathbf{x} \in \mathbb{R}^N$ mit $\|x\|_1 = \sum_{y=1}^{N} |x_y|$ die betragsmäßige Summe aller Vektorkomponenten. Da für jeden klassenweisen *loss* $\hat{\ell}(h,y) \geq 0$ gilt, stellt die 1-Norm durch $\|\hat{\boldsymbol{\ell}}_h\|_1 = \sum_{y=1}^{N} \hat{\ell}(h,y)$ die Summe des klassenabhängigen Verlustes dar. Jedes $\hat{\ell}(h,y)$ ist mit einem positiv korrespondierenden Fehler ϵ_y mit $\delta_y = e^{-2m_y\epsilon_y^2}$ assoziiert. Damit die kleinste obere Schranke für $\|\hat{\boldsymbol{\ell}}_h\|_1$ gefunden werden kann, muss $\sum_{y=1}^{N} \epsilon_y$ unter Einhaltung von δ minimiert werden.

$$\min_{\epsilon_1,\ldots\epsilon_N \in \mathbb{R}} \sum_{y=1}^{N} \epsilon_y, \quad \text{u. d. N.} \begin{cases} \epsilon_1, \ldots \epsilon_N & \geq \tau \\ \delta - \sum_{y=1}^{N} \delta_y & \geq 0 \end{cases} \tag{3.7}$$

Für $(\epsilon_1^*, \ldots, \epsilon_N^*)$ aus 3.7, wird die kleinste obere Schranke durch $\|\ell_h^*\| = \sum_y^N \hat{\ell}(h,y) + \epsilon_y^*$ gebildet.

Euklidische Norm $\|\ell_h\|_2^*$

Die euklidische Norm (2-Norm) wird über den empirischen Verlustvektor mit $\|\hat{\boldsymbol{\ell}}_h\|_2 = \sqrt{\sum_y^N \hat{\ell}(h,y)^2}$ gebildet. Aufgrund der Monotonieeigenschaft der Quadratwurzel, ist es ausreichend $(\epsilon_1, \ldots, \epsilon_N)$ zu finden, die $(\hat{\ell}(h,y) + \epsilon_y)^2$ minimieren. Die kleinste obere Schranke ist gegeben durch $\|\ell_h^*\|_2 = \sqrt{\sum_y^N (\hat{\ell}(h,y) + \epsilon_y^*)^2}$, wobei $(\epsilon_1^*, \ldots, \epsilon_N^*)$ aus 3.8 resultiert.

$$\min_{\epsilon_1,\ldots\epsilon_N \in \mathbb{R}} \sum_{y=1}^{N} (\hat{\ell}(h,y) + \epsilon_y)^2, \quad \text{u. d. N.} \begin{cases} \epsilon_1, \ldots \epsilon_N & \geq \tau \\ \delta - \sum_{y=1}^{N} \delta_y & \geq 0 \end{cases} \tag{3.8}$$

Supremumsnorm $\|\ell_h\|_\infty^*$

Die Supremumsnorm reduziert $\hat{\ell}_h$ auf das maximale klassenweise *loss*. Im Folgendem wird angenommen, dass $\hat{\ell}(h,y)$ mit $\hat{\ell}(h,y) = \text{maximum}(\{\hat{\ell}(h,1),\ldots,\hat{\ell}(h,N)\})$ der größte klassenbezogender Verlust repräsentiert wird. Folglich muss der Fehler ϵ_y minimiert werden. Es muss jedoch sichergestellt werden, dass $\hat{\ell}(h,y)+\epsilon_y$ weiterhin das Maximum von $\{\hat{\ell}(h,1)+\epsilon_1,\ldots,\hat{\ell}(h,N)+\epsilon_N\}$ verkörpert. Das heißt, es muss zusätzlich die Bedingung $\hat{\ell}(h,y)+\epsilon_y \geq \hat{\ell}(h,i)+\epsilon_i$ für alle $i \in \{1,\ldots,N\}$ gelten.

$$\min_{\epsilon_1,\ldots\epsilon_N\in\mathbb{R}} \sum_{y=1}^{N}(\hat{\ell}(h,y)+\epsilon_y)^2, \quad \text{u. d. N.} \begin{cases} \epsilon_1,\ldots\epsilon_N & \geq \tau \\ (\epsilon_y-\epsilon_i) \geq (\hat{\ell}(h,i)-(\hat{\ell(h,y)})) & \forall_{i\in\{1,\ldots,N\}} \\ \delta - \sum_{y=1}^{N}\delta_y & \geq 0 \end{cases} \tag{3.9}$$

Die kleinste obere Schranke für die Supremumsnorm ergibt also $\|\boldsymbol{\ell}_h\|_\infty^* = \hat{\ell}(h,y)+\epsilon_y$.

Methodenvariante $\mathbf{d}_+$

Wie gezeigt wurde, kann durch Anwendung der Hölderschen Ungleichung, die Domänenlücke mit $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ nach oben abgeschätzt werden. Beim Abbilden des Verschiebungsvektors auf eine p-Norm durch $\|\mathbf{d}\|_p = (\sum_{y=1}^{N}|[\mathbf{d}]_y|^p)^{1/p}$, wird der jeweilige Betrag $|[\mathbf{d}]_y|$ einer Verschiebungskomponente betrachtet. Dies hat zur Konsequenz, dass negative Fehleranteile $[\mathbf{d}]_y \cdot \ell(h,y) < 0$ fälschlicherweise positiv bewertet werden und die Abschätzung entsprechend gröber ausfällt. Für die resultierende Schätzung bedeutet dies, dass je größer der negative Fehleranteil durch die Dekomposition der *inter-domain-gap* ist, desto größer ist das Potenzial den tatsächlichen *domain*-induzierte Fehler zu überschätzten.

Von dieser Beobachtung motiviert, kann durch eine Modifikation von $\mathbf{d}$ eine Variante der Methode abgeleitet werden, die genauere Schranken ermöglicht, falls negative Fehleranteile existieren. Hierfür werden negative Komponenten des Verschiebungsvektors auf null gesetzt, wodurch ein Vektor $\mathbf{d}_+$ entsteht für den $max(0,[\mathbf{d}]_y)$ für alle $y \in \{1,\ldots,N\}$ gilt. Aus $\|\mathbf{d}_+\|_p \leq \|\mathbf{d}\|_p$ folgt, dass für die Abschätzung $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_{\boldsymbol{h}}\|_q \leq \|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_{\boldsymbol{h}}\|_q$ gilt. Für eine gegebene kleinste obere Schranke $\|\boldsymbol{\ell}_h\|_q^*$ ist es trivial zu sehen, dass die Methodenvariante $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_h\|_q^*$ ebenfalls die Gleichung 3.4 erfüllt.

Insgesamt kann die Abschätzung $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_{\boldsymbol{h}}\|_q$, in Abhängigkeit zur Größe des negativen Fehleranteils, eine genauere Abschätzungen der Domänenlücke ermöglichen. Aufgrund der Modifikation des Verschiebungsvektors können allerdings die validen Klassenverhältnisse nicht durch eine simple Ungleichung wie 3.4 beschrieben werden, wodurch die Charakterisierung der Menge $\mathcal{P}$ erschwert wird.

Kapitel 4

Vergleichende Analyse

Im Folgendem werden die vorgestellten Methoden zur Zertifizierung von Mehrklassen-Klassifikationsprobleme ausgiebig analysiert und bewertet. Dies geschieht in Form einer Evaluation, die einen systematischen Vergleich zwischen den Zertifizierungsmethoden ermöglicht. Hierfür wird die Analyse anhand von Qualitätsmerkmalen strukturiert, die das Leistungsverhalten der jeweiligen Zertifizierung charakterisieren. Abschließend werden die gesammelten Ergebnisse diskutiert und Anwendungsempfehlungen ausgesprochen.

4.1 Zielsetzung

Das präsentierte Repertoire an Zertifizierungen divergiert teilweise stark im jeweiligen Leistungsverhalten. Damit eine Methode strukturiert analysiert werden kann, muss dessen Verhalten durch relevante Eigenschaften charakterisiert werden. In Anlehnung zur Evaluation der binärer Zertifizierung in [6] wurden folgende Qualitätsmerkmale identifiziert:

- **Korrektheit:** Gemäß eines durch den Anwender spezifizierten δ, darf eine Zertifizierung höchstens mit einer Wahrscheinlichkeit von δ fehlschlagen. Dies hat zur Konsequenz, dass die Abschätzung der *inter-domain-gap* mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ korrekt sein muss.

- **Genauigkeit:** Falls die Abschätzung beliebig grob ausfällt, ist es trivial die Korrektheit einer oberen Schranke zu gewährleisten. Eine korrekte aber ungenaue Schätzung der Domänenlücke hat zur Folge, dass der prognostizierte Fehler ϵ, der durch die jeweilige Domainverschiebung entsteht, überschätzt wird. Dementsprechend wird eine korrekte obere Schranke gesucht, welche möglichst klein ist. Folglich stellt die Schätzung der Domänenlücke im Idealfall eine kleinste obere Schranke dar, die mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ gilt.

- **Interpretation:** Ein trainiertes Klassifikationsmodell wird für eine Menge $\mathcal{P}$ zertifiziert, die gemäß der (ϵ, δ)-Zertifizierung, sichere Klassenverhältnisse beinhalten.

Für den Anwender ist es relevant die Menge $\mathcal{P}$ beschreiben und interpretieren zu können. Unabhängig von der Anzahl der betrachteten Klassen, sollte ein Zertifikat den Nutzern die validen Klassenverhältnisse intuitiv und simpel zugänglich machen können.

- **Effizienz:** Das Zertifizieren von Klassifikationsproblemen sollte für beliebig viele Klassen effizient berechenbar sein.

Die Analyse hat das Ziel, die vorgestellten Zertifizierungsmethoden anhand der dargestellten Eigenschaften einzuordnen und zu bewerten.

4.2 Methodischer Aufbau der Untersuchung

Damit das Verhalten der Zertifizierungen eingeschätzt werden kann, ist ein experimenteller Aufbau notwendig, der belastbare Ergebnisse liefert. Aus diesem Grund muss die Evaluation so gestaltet sein, dass eine Verzerrung der Ergebnisse durch Zufallseinflüsse nicht gegeben beziehungsweise sehr unwahrscheinlich ist. Insbesondere wird dies durch eine ausreichend hohe Anzahl an Experimenten, die auf unterschiedlichen Datensätzen operieren, gewährleistet.

4.2.1 Datensätze

Insgesamt werden die Zertifikate auf sechs Datensätzen evaluiert, die Zwecks Visualisierungen, auf drei Klassen beschränkt wurden. Bei Datensätzen die ursprünglich mehr als drei Label-Klassen aufwiesen, wurden die Klassen zusammengelegt, sodass sich das Lernproblem auf drei Klassen konzentriert. Bei der Auswahl der Datensätze wurde darauf geachtet,

Datensatz	#Features	m_y	#Instanzen	$\mathbb{P}(Y = y)$
optdigits	64	$[3915, 1133, 572]$	5620	$[0,70; 0,20; 0,10]$
satimage	36	$[3746, 1981, 703]$	6430	$[0,58; 0,31; 0,11]$
pendigits	16	$[7740, 2197, 1055]$	10992	$[0,70; 0,20; 0,10]$
eye movements	27	$[3804, 2870, 4262]$	10936	$[0,35; 0,26; 0,39]$
bng	40	$[330606, 330992, 338402]$	1000000	$[0,33; 0,33; 0,33]$
shuttle	9	$[8953, 45586, 3461]$	58000	$[0,15; 0,79; 0,06]$

Tabelle 4.1: Die Datensätze stammen von der Plattform *openML* und umfassen sowohl Datenmengen mit gleichverteilten Klassenhäufigkeiten, wie auch Datensätze mit unbalancierten Klassenverteilungen. Die Anzahl der Klasseninstanzen sind in der Spalte m_y aufgetragen.

dass diese eine breite Spanne an unterschiedlichen Ausprägungen abdecken. Insbesondere variieren die Datenmengen in der Anzahl der Klasseninstanzen m_y, die aufgrund der PAC-Schranken einen direkten Einfluss auf die durchgeführten Zertifizierungen besitzen.

4.2.2 Design der Experimente

In jedem Experiment wird eine Zertifizierungsmethode auf einem Datensatz D getestet (siehe Tabelle 4.1). Der Datensatz wird in drei disjunkte Teilmengen D_{trn}, D_{val} und $D_{\text{remaining}}$ aufgeteilt. Die Datenmengen D_{trn} und D_{val} werden stratifiziert aus D gezogen, und repräsentieren Dateninstanzen aus der *domain* $\mathcal{S}$. Folglich entsprechen die Klassenverhältnisse aus D_{trn} und D_{val} näherungsweise der Klassenverteilung $\mathbb{P}(Y = y)$ von D.

Ein Klassifikator h wird durch D_{trn} trainiert und prognostiziert Klassenlabel für die Datenbeispiele aus D_{val}. Auf Grundlage der Prognosen können die Zertifizierungsmethoden angewendet werden, sowie $\hat{L}_{\mathcal{S}}(h)$ geschätzt werden.

Anschließend wird die ausgestellte Zertifizierung durch eine Vielzahl an gleichverteilten Punkten $\mathbf{p}_{\mathcal{T}}$ aus dem Wahrscheinlichkeitsraum $[0, 1]^3$ geprüft, wobei $\mathbf{p}_{\mathcal{T}}$ Klassenverhältnisse aus dem Deployment symbolisieren. Damit eine ausreichende Abdeckung des Wahrscheinlichkeitsraums gewährleistet ist, werden pro Experiment 1000 Verschiebungspunkte geprüft. Es entsteht eine Testmenge D_{tst} mit dem Klassenverhältnis $\mathbf{p}_{\mathcal{T}}$, indem Dateninstanzen nach $\mathbf{p}_{\mathcal{T}}$ aus $D_{\text{remaining}}$ gezogen wird. Durch D_{tst} kann der empirische Verlust $\hat{L}_{\mathcal{T}}(h)$ von h im Deployment $\mathbf{p}_{\mathcal{T}}$ bestimmt werden. Wiederum wird die (ϵ, δ)-Zertifizierung in Abhängigkeit zur Verschiebung gesetzt, sodass der zertifizierte *domain*-induzierte Fehler ϵ prognostiziert wird.

Die Experimente wurden neunmal unter verschiedenen Datenaufteilungen wiederholt, sodass robuste Ergebnisse gewährleistet sind. Folglich wurde pro Datensatz jede Zertifizierungsmethode an 9000 Punkten getestet.

In den durchgeführten Experimenten wurde als Klassifikator die Logistische Regression mit einer Standardkonfiguration verwendet, die unter dem *zero-one-loss* ausgewertet wird. Dabei wurde die Logistische Regression aus keinen spezifischen Grund gewählt. Stattdessen ist das Funktionsprinzip der Zertifizierungen unabhängig von der Wahl des Klassifikators zu betrachten. Ebenso können die Zertifizierungen mit beliebigen dekomponierbaren Verlustfunktionen betrieben werden. Im Anhang sind exemplarisch Zertifizierungsergebnisse aufgeführt, die unter verschiedenen Klassifikatoren und *loss*-Funktionen produziert wurden.

4.3 Durchführung der Evaluation

4.3.1 Korrektheit

Die Korrektheit einer Zertifizierung ist äquivalent mit der Fragestellung, ob der durch die Zertifizierung prognostizierte Fehler ϵ eine obere Schranke für $\hat{L}_{\mathcal{T}}$ charakterisiert. Dass heißt, es muss $\hat{L}_S(h)+\epsilon \geq \hat{L}_{\mathcal{T}}(h)$ gelten. Nach Definition 2.5.1 darf eine (ϵ, δ)-Zertifizierung höchstens mit einer Wahrscheinlichkeit von δ den wahren *domain*-induzierten Fehler nicht durch ϵ beschränken.

In Tabelle 4.2 sind die Fehlschläge für $\delta = 0,05$ für die entsprechenden Zertifizierungsmethoden aufgelistet. Die Ergebnisse zeigen, dass jede der präsentierten Zertifizierungen sehr wenige Fehlschläge produziert und deutlich unter dem vorgegebenen Wahrscheinlichkeitsbudget von δ liegt. Aufgrund der gröberen Abschätzung durch die Hölderschen Ungleichung weisen die *Normed Certificates* erwartungsgemäß ein noch geringeres Fehlerniveau als das *Signed Certificate* auf. Insgesamt zeigen die Ergebnisse für alle vorgestellten Zertifizierungen ein sehr hohes Maß an Korrektheit.

Datensatz	Anzahl Fehlschläge	Anteil Fehlschläge
Signed Certificate	143	0,0026
$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	0	$0,0$
$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	1	$0,0$
$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	0	$0,0$
$\|\mathbf{d}_+\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	3	$0,0001$
$\|\mathbf{d}_+\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	3	$0,0001$
$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	1	$0,0$

Tabelle 4.2: Korrektheit der Zertifikate unter einem Budget von $\delta = 0.05$. Insgesamt wurden pro Methode 54000 Zertifizierungen ausgestellt.

Zur besseren Einschätzung des Einflusses der PAC-Schranken auf die Korrektheit der Zertifizierung, wurden die Experimente ohne PAC-Schranken wiederholt. Dass heißt, es wurde der jeweilige empirische Verlust verwendet und nicht dessen obere Schranke. Wie die Ergebnisse in Tabelle 4.3 zeigen, verursachten die empirischen Zertifikate minimal mehr Fehler, wie die PAC-betriebenen Methoden. Somit bleibt auch für die rein empirischen Abschätzungen das Fehlerniveau sehr niedrig. Jedoch verliert die Zertifizierung ihre statistische Aussagekraft, falls nicht zusätzlich PAC-Schranken verwendet werden.

Datensatz	Anzahl Fehlschläge	Anteil Fehlschläge
Signed Certificate	176	$0,0033$
$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	2	$0,0$
$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	2	$0,0$
$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	2	$0,0$
$\|\mathbf{d}_+\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	154	$0,0029$
$\|\mathbf{d}_+\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	201	0.0037
$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	1	$0,0$

Tabelle 4.3: Korrektheit der Zertifikate beim Verzicht auf PAC-Schranken bei 54000 durchgeführten Zertifizierungen pro Methode.

4.3.2 Genauigkeit

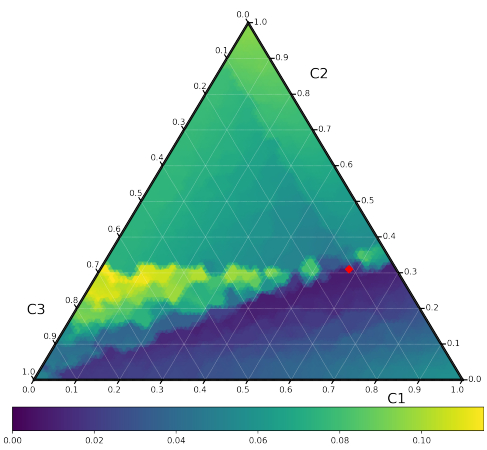
Wie zuvor gesehen, weisen die Mehrklassen-Zertifikate einen hohen Grad an Korrektheit auf und sind somit mit Abschätzungen assoziiert, die mit einer hohen Wahrscheinlichkeit obere Schranken der *inter-domain-gap* darstellen. Allerdings sollten die oberen Schranken möglichst klein sein, damit diese die *inter-domain-gap* akkurat charakterisieren können. Ungenaue obere Schranken haben die Konsequenz, dass der durch die Zertifizierung prognostizierte *domain*-induzierte Fehler ϵ systematisch überschätzt wird. Eine durchgeführte (ϵ, δ)-Zertifizierung auf Grundlage eines solchen ungenauen ϵ-Schätzers, hat zur Folge, dass die Menge $\mathcal{P}$ nicht alle validen Klassenverhältnisse umfassen kann. Somit wird der Umfang von $\mathcal{P}$ unterschätzt.

Die Genauigkeit der Vorhersage von ϵ durch die Zertifizierungsmethode ist daher ein wichtiges Untersuchungsobjekt bei der Einschätzung der Leistungsfähigkeit. Hierbei wird die Genauigkeit als Distanz zwischen $\hat{L}_S(h) + \epsilon$ und $\hat{L}_{\mathcal{T}}(h) + \epsilon_{\text{tst}}$ aufgefasst. Der Schätzfehler ϵ_{tst} resultiert aus der Berechnung der PAC-Schranke in 2.1.1 und reflektiert den Umstand, dass $\hat{L}_{\mathcal{T}}(h)$ im Experiment ebenfalls nur aus endlichen Datenmengen geschätzt wurde. Je kleiner die absoluten Abweichungen $|\hat{L}_S(h) + \epsilon - \hat{L}_{\mathcal{T}} - \epsilon_{\text{tst}}|$ sind, desto akkurater wurde ϵ durch die Zertifizierung geschätzt. Die jeweils ermittelten Distanzen wurden in Abhängigkeit zur Zertifizierungmethode gruppiert und anschließend nach dem arithmetischen Mittel aggregiert. Somit wurde für jede getestete Label-Verschiebung die mittlere absolute Abweichung (engl: *mean absolute deviation*, **MAD**) zur Distanz null berechnet.

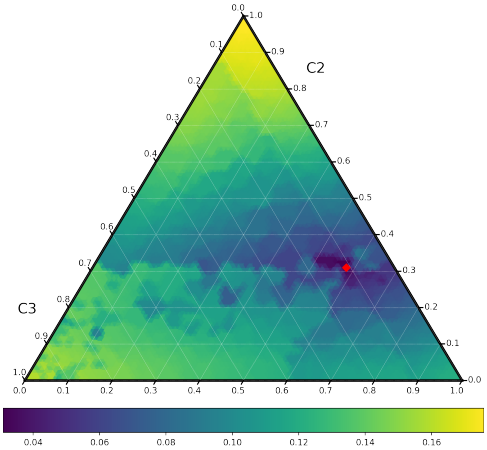
Die gesammelten Ergebnisse in Tabelle 4.4 zeigen, dass ein erheblicher Unterschied in den Schätzfähigkeiten zwischen dem *Signed Certificate* und der Klasse der *Normed Certificate* besteht.

***Signed Certificate*:** Mit einer mittleren Abweichung von nur $0,0648$, weißt das *Signed Certificate* exzellente Schätzfähigkeiten auf, die eine akkurate Vorhersage von ϵ ermöglichen. Zusätzlich bietet das *Signed Certificate* die Bildung von Konfidenzintervallen, die den wahren Fehler ϵ mit einer Wahrscheinlichkeit von mindestens $(1 - \delta)^2$ beinhalten. Das optional gebildete Konfidenzintervall kann verwendet werden, um die Unsicherheit des Schätzers einzuschätzen, indem die Breite des Intervalls betrachtet wird. Sehr breite Intervalle weisen auf eine hohe Unsicherheit des Schätzers hin, was auf eine unzureichende Datenbasis zurückzuführen ist. Zudem kann anhand der Schätzintervalle direkt abgelesen werden, ob eine Klassenverschiebung *signifikante* Auswirkungen auf die Modellperformanz besitzt. Somit liefern Konfidenzintervalle dem Anwender wertvolle Informationen, die erhaltenen Prognosen zu ϵ besser einschätzten zu können.

***Normed Certificate*:** Die Zertifizierungen, die auf der Hölderschen Ungleichung beruhen, weisen im Vergleich zum *Signed Certificate* deutlich schlechtere Schätzfähigkeiten auf. Gemessen am **MAD** schneidet aus der Gruppe $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ die Zertifizierung mit



(a) Distanz von $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$: In den meisten Regionen wurde der Fehler ϵ mit geringen Abweichung vorhergesagt.

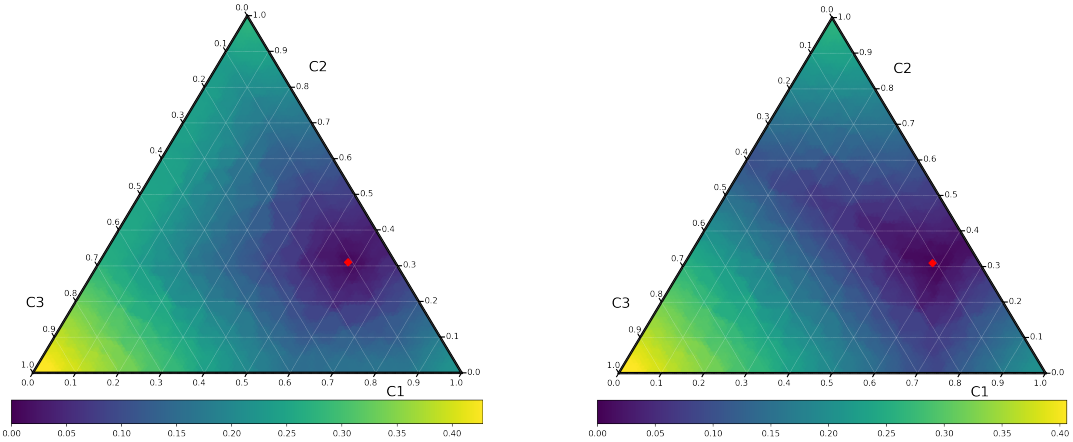


(b) Breite des Konfidenzintervalls: Hellere Areale symbolisieren Klassenverschiebungen in denen der ϵ-Schätzer eine größere Unsicherheit aufwies.

Abbildung 4.1: Das *Signed Certificate* unter Anwendung des *zero-one loss* auf dem Datensatz *optdigits*. Sofern nicht anders gekennzeichnet, symbolisiert die einzeichnete rote Markierung die *Source* Klassenverhältnisse $\mathbf{p}_S$.

$p = q = 2$ am Besten ab, während $\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$ und $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ auf einen ähnlichen Schätzniveau liegen. Durch das Modifizieren des Verschiebungsvektors $\mathbf{d}$ zu $\mathbf{d}_+$ konnte die Genauigkeit der Prognosen für ϵ verbessert werden. Bei der Verwendung von $\mathbf{d}_+$ profitiert die Methode mit $p = 1$ und $q = \infty$ am stärksten. Dort konnte die Genauigkeit, gemessen an der **MAD**, mehr als verdoppelt werden. Ebenso weist $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ eine um 30% geringere Abweichung im direkten Vergleich zur Ursprungsmethode auf. Im Fall von ($p = \infty$,$q = 1$) verbessert die Modellvariante die Genauigkeit im Durchschnitt lediglich um 10%. Das geringere Verbesserungspotenzial ist durch die Supremumsnorm auf $\mathbf{d}_+$ zu erklären, wodurch $\mathbf{d}_+$ auf die maximale absolute Komponente reduziert wird. Infolgedessen ist die Modifikation des Vektors nur in wenigen Verschiebungskonstellationen bemerkbar. Siehe dazu Abbildung 4.2.

Alle untersuchten *Normed Certificates* zeigten mit steigender Label-Verschiebung kontinuierlich größer werdende Ungenauigkeiten. Wurden nur Fehler für kleinere Label-Verschiebungen vorhergesagt, lieferten alle Methoden akkurate Ergebnisse. Mit größer werdenden Verschiebungen, die durch $\mathbf{d}$ beziehungsweise $\mathbf{d}_+$ repräsentiert werden, wurden die *Normed Certificates* zunehmend ungenauer. Wenn extreme Verschiebungen analysiert wurden, neigten die Zertifikate zum starken Überschätzen des *domain*-induzierten Fehler.



(a) Genauigkeit von $\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$ **(b)** Genauigkeit von $\|\mathbf{d}_+\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$

Abbildung 4.2: Die Genauigkeit von $\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$ im direkten Vergleich mit dessen $\mathbf{d}_+$ Variante. Die jeweiligen Schätzungen fanden unter den gleichen Bedingungen wie in Abbildung 4.1 statt.

Zertifikat	**MAD**	Q_1	Q_2	Q_3
Signed Certificate	$0,0648$	$0,0217$	0.0488	$0,0843$
$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,2709$	$0,1271$	$0,2159$	$0,3682$
$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,2236$	$0,1059$	$0,1704$	$0,29$
$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,2754$	$0,1244$	$0,2101$	$0,3419$
$\|\mathbf{d}_+\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,1246$	$0,054$	$0,0922$	$0,169$
$\|\mathbf{d}_+\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,157$	$0,0632$	$0,1087$	$0,2051$
$\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,2492$	$0,1005$	$0,1759$	$0,3047$

Tabelle 4.4: Die ermittelten Distanzen zwischen $\hat{L}_S(h) + \epsilon$ und $\hat{L}_\mathcal{T}(h) + \epsilon_{\text{tst}}$ zusammengefasst durch die *Mittlere Absolute Abweichung* zur Distanz 0 (**MAD**), sowie nach dem unteren *Quartil* (Q_1), dem *Median* (Q_2) und dem oberen *Quartil* (Q_3).

4.3.3 Interpretation

In den vorausgegangenen Analysen wurde das Leistungsverhalten der Zertifizierungen anhand der verwendeten Abschätzungen der *inter-domain-gap* bewertet. Neben akkuraten Prognosen des *domain*-induzierten Fehlers, erwarten Anwender bei einer (ϵ, δ)-Zertifizierung eines trainierten Klassifikationsmodells, dass die zertifizierte Menge $\mathcal{P}$ die sicheren Klassenverschiebungen umfassend und für den Anwender verständlich dargestellt werden kann.

Normed Certificate Wie in Kapitel 3.4 beschrieben, können valide Klassenverhältnisse, die unter $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ zertifiziert wurden, durch die Ungleichung $\|\mathbf{d}\|_p \leq \frac{\epsilon}{\|\boldsymbol{\ell}_h\|_q^*} = \Delta p^*$ beschrieben werden. Folglich setzt sich die Menge $\mathcal{P}$ aus Punkten $\mathbf{p}_\mathcal{T} \in [0,1]^N$ zusammen,

für die 3.4 gilt. Für die *Normed Certificate* kann daher die maximale Verschiebung als Wert einer Norm ausgedrückt werden, was eine einfache Charakterisierung von $\mathcal{P}$ erlaubt.

Für das Zertifikat $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ ist es möglich solche Grenzen zu definieren. Durch das Berechnen der Supremumsnorm auf $\mathbf{d}$ reduziert sich $\|\mathbf{d}\|_\infty$ auf die Klasse i, die die größte absolute Klassenverschiebung mit $|[\mathbf{p}_\mathcal{T}]_i - [\mathbf{p}_\mathcal{S}]_i| = \Delta p$ aufweist. In analoger Weise zur binären Zertifizierung, kann der zulässige Bereich der sicheren Klassenverhältnisse durch $[\ [\mathbf{p}_S]_i - \Delta p^*\ ,\ [\mathbf{p}_\mathcal{S}]_i + \Delta p^*] = [p^{\min}_{\mathcal{T},i}, p^{\max}_{\mathcal{T},i}]$ beschrieben werden. Hierbei ist $\Delta p^* = \frac{\epsilon}{\|\boldsymbol{\ell}_h^*\|_1}$ für alle Klassen konstant und stellt die größte absolute Verschiebung dar, die eine Klasse haben darf, um die Zertifizierung gemäß Definition 2.5.1 zu erfüllen. Daher ist die Menge $\mathcal{P}$ für $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ beschreibbar durch:

$$\mathcal{P} = \left\{ \mathbf{p} \in [0,1]^N \ :\ [\mathbf{p}]_i \in [p^{\min}_{\mathcal{T},i}, p^{\max}_{\mathcal{T},i}]\ \ \forall\, i \in \{1,\dots,N\} \text{ und } \sum_{i=1}^{N} [\mathbf{p}]_i = 1 \right\} \tag{4.1}$$

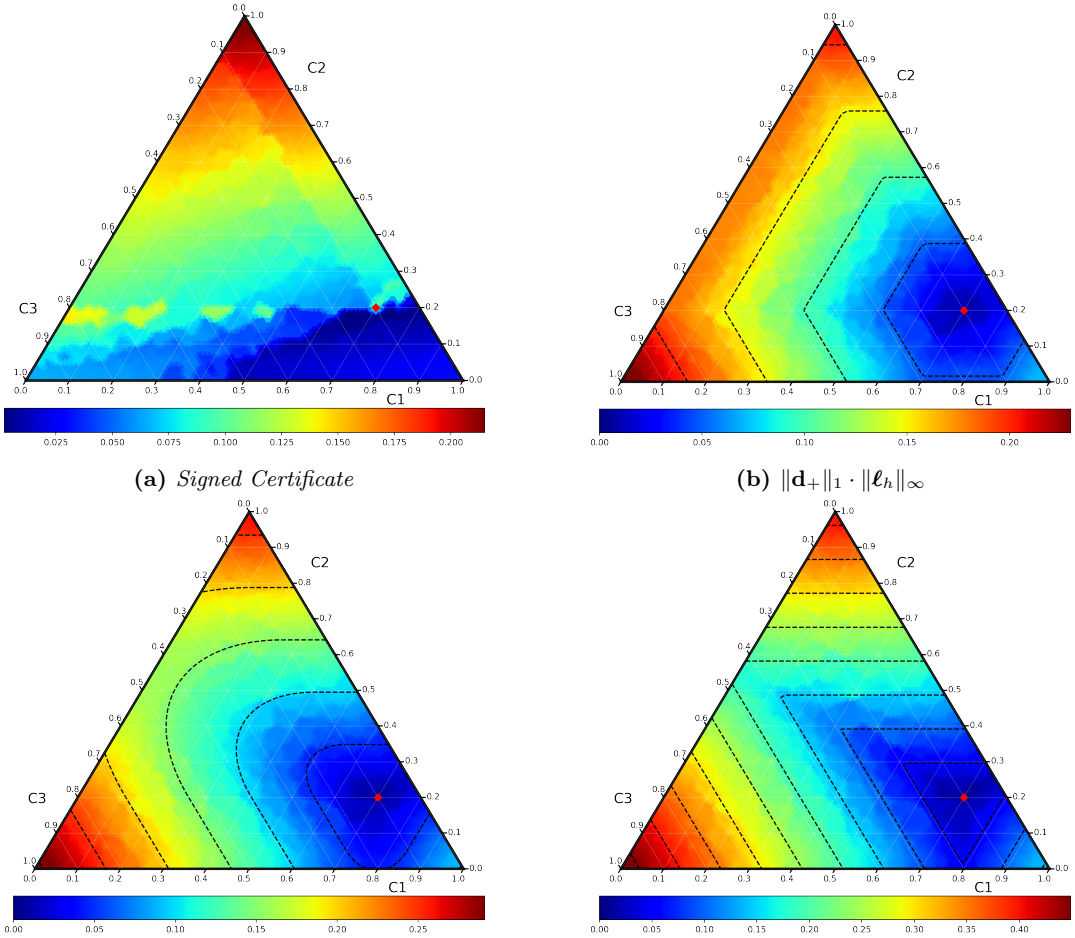
In Tabelle 4.5 sind für $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ die jeweiligen maximalen Δp^* eingetragen, die durch die Zertifizierung ermittelt wurden. Die simple Charakterisierung der Grenzen für maximale Label-Verschiebungen, ermöglichen für $(p = \infty, q = 1)$ die Menge $\mathcal{P}$ direkt zu beschreiben. Beispielsweise gilt nach Tabelle 4.5 für den *optdigits* Datensatz, dass jedes Klassenverhältnis gemäß $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ als sicher gilt, wenn ausgehend von $\mathbf{p}_S$ die klassenweise absolute Abweichung kleiner als $0,2$ beträgt.

Approximative Charakterisierung von $\mathcal{P}$

Für die Methodenvariante $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ kann $\mathcal{P}$ nicht durch die Ungleichung 3.5 beschrieben werden. Gleichfalls ist das *Signed Certificate* nur im Stande einzelne Klassenverschiebungen zu bewerten. Damit eine Menge $\mathcal{P}$ beschrieben werden kann, muss der *domain*-induzierte Fehler an ausreichend vielen Punkten im Wahrscheinlichkeitsraum $[0,1]^N$ evaluiert werden. Bei ausreichender Abdeckung des Raums, ergibt sich eine näherungsweise Kartografierung von ϵ für beliebige Verteilungen $\mathbf{p}_\mathcal{T} - \mathbf{p}_S$, wobei dies durch Interpolation auf Grundlage der evaluierten Punkte ermöglicht wird. Dieses Vorgehen entspricht der hier durchgeführten experimentellen Anordnung.

Für die Zertifikate $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ können die Grenzen der sicheren Klassenverhältnisse visuell inspiziert werden. Dies wird durch die Konstruktion der *Normed Certificates* gewährleistet. So wird ausgehend von $\mathbf{p}_S$ ein Fehlerareal durch $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ induziert, wobei die verwendete Norm $\|\cdot\|_p$ auf $\mathbf{d}_+$ die Form der Grenzen bestimmt. Siehe dazu Abbildung 4.3.

Wiederum weißt die durch das *Signed Certificate* prognostizierte Fehlerlandschaft keine allgemeinen Muster auf. Das Identifizieren und beschreiben von sicheren Klassenverhältnissen stellt somit eine größere Schwierigkeit dar. Falls Klassifikationsprobleme mit mehr als drei Klassen betrachtet werden, ist das Visualisieren von maximal gültigen Klassenverteilungen nicht mehr möglich.



(a) *Signed Certificate* **(b)** $\|\mathbf{d}_+\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$

(c) $\|\mathbf{d}_+\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$ **(d)** $\|\mathbf{d}_+\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$

Abbildung 4.3: Die Modellvariante $\mathbf{d}_+$ sowie das *Signed Certificate* können $\mathcal{P}$ nur approximativ bestimmen. Beim $\|\mathbf{d}_+\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ Zertifikat sind die Grenzen der maximal zulässigen Label-Verschiebung durch Konturlinien identifizierbar. Die Fehlerlandschaft des *Signed Certificate* weißt im Allgemeinen keine strukturellen Muster auf, die ein einfaches Charakterisieren von Grenzen ermöglicht.

Datensatz	Zertifikat	$L_S(h)$	$\mathbf{p}_S^\top$	Δp^*
optdigits	$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,0906919$	$[0.7, 0.2, 0.1]$	$0,388121$
optdigits	$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,0906919$	$[0.7, 0.2, 0.1]$	$0,313647$
optdigits	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,0906919$	$[0.7, 0.2, 0.1]$	0.196824
satimage	$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,103752$	$[0.58, 0.31, 0.11]$	0.52352
satimage	$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,103752$	$[0.58, 0.31, 0.11]$	$0,364574$
satimage	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,103752$	$[0.58, 0.31, 0.11]$	0.213732
pendigits	$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,159151$	$[0.7, 0.2, 0.1]$	$0,191792$
pendigits	$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,159151$	$[0.7, 0.2, 0.1]$	$0,187042$
pendigits	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,159151$	$[0.7, 0.2, 0.1]$	$0,139398$
eye movements	$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,512418$	$[0.350.26, 0.39]$	$0,175881$
eye movements	$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,512418$	$[0.350.26, 0.39]$	$0,106797$
eye movements	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,512418$	$[0.350.26, 0.39]$	$0,0619069$
bng	$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,113992$	$[0.33, 0.33, 0.33]$	$0,642102$
bng	$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,113992$	$[0.33, 0.33, 0.33]$	$0,480726$
bng	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,113992$	$[0.33, 0.33, 0.33]$	$0,285714$
shuttle	$\|\mathbf{d}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,0580599$	$[0.15, 0.79, 0.06]$	$0,33393$
shuttle	$\|\mathbf{d}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$0,0580599$	$[0.15, 0.79, 0.06]$	$0,3194$
shuttle	$\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$0,0580599$	$[0.15, 0.79, 0.06]$	0.247768

Tabelle 4.5: Zulässige Klassenklassenverhältnisse Δp^*, gemäß $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ mit $\epsilon = 0,1$ und $\delta = 0,05$.

Insbesondere für das *Signed Certificate* besteht die Herausforderung, aus der evaluierten Punktmenge Teilmengen verständlich zu beschreiben, für die der prognostizierte Fehler kleiner als ϵ ist. Ein allgemeiner und naheliegender Ansatz ist das Berechnen von Konvexen Hüllen um solche *sicheren* Punktwolken, wodurch die Grenzen von sicheren Klassenverhältnissen beschrieben werden können. Hierfür müsste im Vorfeld durch Clusteringverfahren die zusammenhängenden *sicheren* Punktwolken identifiziert werden. Da in dieser Arbeit nur Probleme mit genau drei Klassen behandelt werden, kann die Durchführbarkeit des skizzierten Ansatzes bezüglich Fehleranfälligkeit und Praktikabilität nicht abschließend bewertet werden.

4.3.4 Effizienz

Wie zuvor diskutiert, kann $\mathcal{P}$ von einigen Methoden nur approximativ bestimmt werden, indem der Fehler ϵ für eine Vielzahl an Verschiebungen prognostiziert wird. Ein effizientes Auswerten von einer Menge an Verschiebungspunkten ist somit von Bedeutung. Dementsprechend sind Laufzeitverhalten und die Ressourcenanforderungen eng mit der Interpretierbarkeit der Zertifizierung verknüpft.

Zertifikat	pro Punkt	10 Punkte	100 Punkte	1000 Punkte
$\|\mathbf{d}_{(+)}\|_1 \cdot \|\boldsymbol{\ell}_h\|_\infty$	$0,81$ $(\pm 0,2)$	$0,79$ $(\pm 0,22)$	$0,86$ $(\pm 0,28)$	$0,79$ $(\pm 0,19)$
$\|\mathbf{d}_{(+)}\|_2 \cdot \|\boldsymbol{\ell}_h\|_2$	$2,46$ $(\pm 2,3)$	$4,46$ $(\pm 1,6)$	$2,7$ $(\pm 3,1)$	$2,58$ $(\pm 2,0)$
$\|\mathbf{d}_{(+)}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$	$3,39$ $(\pm 2,39)$	$4,19$ $(\pm 1,25)$	$4,5$ $(\pm 2,3)$	$3,59$ $(\pm 4,3)$
Normed Certificate	$17,7$ $(\pm 9,91)$	$163,2$ $(\pm 43,1)$	1820 (± 342)	15300 (± 2082)

Tabelle 4.6: Laufzeitmessungen (in Sekunden) für die Zertifizierungsmethoden. Getestet wurde auf dem Datensatz *optdigits*. Die Messungen wurden insgesamt fünfmal wiederholt.

Normed Certificate Bei den Methoden aus der Klasse $\|\mathbf{d}_{(+)}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ muss die kleinste obere Schranke $\|\boldsymbol{\ell}_h\|_q^*$ optimiert werden, was für alle Methoden mit geringen Aufwand realisierbar war. Soll für eine gegebene Verschiebung $\mathbf{d} = \mathbf{p}_\mathcal{T} - \mathbf{p}_S$ der *domain*-induzierte Fehler geschätzt werden, so gilt $\epsilon = \|\mathbf{d}_{(+)}\| \cdot \|\boldsymbol{\ell}_h\|_q^*$. Vorhersagen für eine Menge an Punkten $\mathbf{P}$ mit $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}_M\}$ lassen sich durch eine Matrizenmultiplikation mit

$$\begin{bmatrix} \|\mathbf{p}_1 - \mathbf{p}_S\|_p \\ \vdots \\ \|\mathbf{p}_i - \mathbf{p}_S\|_p \\ \vdots \\ \|\mathbf{p}_M - \mathbf{p}_S\|_p \end{bmatrix} \cdot \begin{pmatrix} \|\boldsymbol{\ell}_h\|_q^* & & & & \\ & \ddots & & & \\ & & \|\boldsymbol{\ell}_h\|_q^* & & \\ & & & \ddots & \\ & & & & \|\boldsymbol{\ell}_h\|_q^* \end{pmatrix} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_i \\ \vdots \\ \epsilon_M \end{bmatrix}$$

berechnen. Da $\|\boldsymbol{\ell}_h\|_q^*$ unabhängig von der Label-Verschiebung ist, muss $\|\boldsymbol{\ell}_h\|_q^*$ nur einmal bestimmt werden.

Signed Certificate Beim *Signed Certificate* wird der Abstand optimiert, der zwischen dem positiven und negativen Anteil des Verlustes unter einer Verschiebung $\mathbf{d}$ entsteht. Da die vorzeichenbehafteten Verlustanteile direkt von $\mathbf{d}$ abhängen, kann die Optimierung nicht unabhängig von einer Klassenverschiebung durchgeführt werden. Daher muss für jeden Verschiebungspunkt die Schranken gemäß 2.17 beziehungsweise 2.18 berechnet werden. Wie aus der Tabelle 4.6 entnehmbar ist, führt dies mit steigender Anzahl an Punkten zu enormen Laufzeiten. Für das *Signed Certificate* besteht die Möglichkeit die Laufzeit zu verkürzen, in dem die Evaluation der Punkte parallelisiert werden. Allerdings würde eine parallele Berechnung einen höheren Ressourcenaufwand erfordern.

Insgesamt zeigen die Laufzeitmessungen, dass die vorzeichenbehaftete Zertifizierung sehr schlecht mit der Anzahl an zu prognostizierten Punkten skaliert. Da mit steigender Klassenanzahl ungleich mehr Punkte für eine ausreichende Abdeckung des Raumes benötigt werden, ist eine vollständige Abbildung des *domain*-induzierten Fehlers nur unter massiven Ressourceneinsatz zu erwarten. Jedoch ist hier anzumerken, dass der Rechenaufwand für eine einzelne Punktschätzung unabhängig von der Anzahl der Klassen ist.

4.4 Diskussion

Die durchgeführte Evaluation zeigte, dass keine der vorgestellten Zertifizierungsmethoden die in 4.1 definierten Anforderungen voll-umfänglich erfüllen kann. Stattdessen zeigte sich, dass das Repertoire an Zertifizierungen unterschiedliche Leistungsverhalten aufweisen, die ein Spektrum an Anwendungsprofilen induzieren. Insbesondere wurde offensichtlich, dass genaue Abschätzungen der Domänenlücke die Beschreibung der Menge $\mathcal{P}$ erschweren. Eine simple Charakterisierung von $\mathcal{P}$, die in analoger Weise zur binären Zertifizierung funktioniert, konnte nur auf Grundlage einer robusteren Abschätzung der *inter*-domain-gap erfolgen.

Die Pole zwischen sehr genauen Schätzfähigkeiten einerseits und leichte Interpretierbarkeit andererseits werden durch das *Signed Certificate* und $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ verkörpert.

Das *Signed Certificate* weißt exzellente Schätzfähigkeiten des *domain*-induzierten Fehlers auf. So kann nur das *Signed Certificate* den Fehler ϵ vorzeichenbehaftet prognostizieren. Zudem bietet das Zertifikat die Option ϵ durch Konfidenzintervalle zu schätzen. Dies erlaubt dem Anwender wertvolle Information über die Unsicherheit des Schätzers zu erhalten.

Allerdings kann das *Signed Certificate* die Menge $\mathcal{P}$ nur approximativ durch ein Menge von Datenbeispielen beschreiben. Zudem wird die Interpretation erschwert, da das Zertifikat für jede Klassenverschiebung berechnet werden muss. Folglich ist die Beschreibung von $\mathcal{P}$ mit hohen Aufwand verbunden, wobei dies insbesondere für hochdimensionale Klassifikationsprobleme gilt. Insgesamt kann für das *Signed Certificate* eine starke Empfehlung ausgesprochen werden, falls nur der Fehler von einigen wenigen Verschiebungen abgeschätzt werden muss.

Im Vergleich zum *Signed Certificate*, eignet sich das *Normed Certificate* $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ für den konträren Anwendungsfall. Diese Zertifizierung korrespondiert mit einer deutlich gröberen Abschätzung der Domänenlücke, die es jedoch ermöglicht, $\mathcal{P}$ intuitiv zu beschreiben. Das Alleinstellungsmerkmal der Methode ist in diesem Fall, dass die Interpretierbarkeit von $\mathcal{P}$ nicht von der Anzahl der Klassen abhängt. So ist es möglich, die validen Klassenverhältnisse für beliebig hochdimensionale Klassifikationsprobleme simpel zu charakterisieren. Dementsprechend eignet sich die Zertifizierung nach $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$, falls eine einfache und direkte Charakterisierung von beliebig hochdimensionalen Klassifikationsproblemen in den Anforderungen eine hohe Gewichtung erfährt.

Die übrigen *Normed Certificate* weisen ähnliche Eigenschaften auf, wie die $\|\mathbf{d}\|_\infty \cdot \|\boldsymbol{\ell}_h\|_1$ Zertifizierung. Allerdings ist das Beschreiben von gültigen Klassenverhältnissen durch Abweichungen unter der $\|\cdot\|_1$ und $\|\cdot\|_2$ Norm weniger intuitiv, als für die Supremumsnorm.

Durch die Modifikation des Vektors $\mathbf{d}$ zu $\mathbf{d}_+$ konnte eine verbesserte Genauigkeit bei den *Normed Certificates* erreicht werden. Jedoch war es aufgrund des modifizierten Verschiebungsvektors nicht mehr möglich die Menge $\mathcal{P}$ durch einen festen Abstand zu bestimmen. Stattdessen konnte $\mathcal{P}$ nur durch eine Menge von Dateninstanzen charakterisiert werden.

Die Modellvariante $\mathbf{d}_+$ bestätigt somit das Muster, dass eine einfache Charakterisierung von $\mathcal{P}$ eine gröbere Schätzung der Domänenlücke impliziert.

Kapitel 5

Fazit und Ausblick

In dieser Arbeit wurde erfolgreich die Anwendbarkeit der Zertifikate auf Mehrklassen-Klassifikationsprobleme erweitert. Hierfür wurden mehrere Zertifizierungsstrategien vorgestellt, die jeweils unterschiedliche Anwendungsprofile besitzen. Somit deckt die Gesamtheit der vorgestellten Zertifizierungen eine breite Palette von Anwendungen ab.

Ebenfalls wurde deutlich, dass keine Zertifizierungsmethode allen Anforderungen gerecht werden kann. Stattdessen muss der Anwender eine adäquate Methode anhand des Anforderungsprofils auswählen.

Weitere Arbeiten können sich auf die Weiterentwicklung der Zertifizierungsmethoden beziehen. So konnte mit der Methodenvariante $\mathbf{d}_+$ gezeigt werden, dass die Genauigkeit der Abschätzung der Domänenlücke verbessert werden kann, wenn der negative Fehleranteil ignoriert wird. Darauf aufbauend würde die Genauigkeit der Zertifizierung weitergehend gesteigert werden, wenn der negative Fehlerbestandteil durch eine größte untere Schranke abgeschätzt werden würde. Dies hätte zur Folge, dass sowohl der positive- wie auch der negative Fehlerbestandteil durch optimierte Schranken abgeschätzt werden würde und für die Schätzung der Domänenlücke verwendet werden würden, was die Schätzung der Domänenlücke und somit das Zertifikat weiter verbessern würde.

Teil II

ACS-Akquisestrategien

Kapitel 6

Heuristische ACS-Strategien

Wie bereits erläutert, versuchen ACS-Strategien das Akquiseverhalten von generierten Daten zu optimieren, sodass ein Modell mit möglichst wenigen Daten performant trainiert werden kann. Lomasky et al. [13] empfehlen einen schrittweisen Aufbau der Trainingsmenge auf Grundlage von Akquise-Empfehlungen durch ACS-Strategien.

Den Startpunkt bildet ein anfänglicher Trainingsdatensatz D_1, deren klassenweise Anzahl von Dateninstanzen durch $\mathbf{m}[1] = (m_1, \ldots, m_N)$ gegeben ist. Die initiale Zusammensetzung von D_1 wird dabei durch den Anwender beziehungsweise Domänenexperten spezifiziert. Die Akquisephasen werden in Runden strukturiert, in denen ein bestimmtes Budget an Daten durch den Generator erzeugt wird. Für eine Runde i können insgesamt $b_{[i]}$ Dateninstanzen generiert werden, wobei die Annahme gilt, dass die klassenweisen Akquisekosten identisch sind. ACS-Strategien versuchen in jeder Runde vorteilhafte Klassenverteilungen $\mathbf{p}_{[i]} \in [0, 1]^N$ zu finden, nach denen $b_{[i]}$ Daten produziert werden. Dies bedeutet, dass sich die Trainingsmenge für Runde i durch $D_i = D_{i-1} + D_{\text{new}}$ zusammensetzt, wobei D_{new} aus den $b_{[i]}$ Dateninstanzen besteht, die nach $\mathbf{p}_{b_{[i]}}$ generiert wurden. Auf diese Weise wird ein Trainingsdatensatz sukzessive aufgebaut, bis eine zufriedenstellende Leistung des Modells erreicht ist oder ein anderes Abbruchkriterium erfüllt ist.

Die meisten von Lomasky et al. [13] vorgeschlagenen Strategien treffen ihre Verteilungsempfehlung $\mathbf{p}_{b_{[i]}}$ auf Grundlage der ermittelten Modell-Performanz auf den bereits generierten ACS-Datensatz D_{i-1}. In [13] wird die Modellgüte durch *accuracy* (*acc*) bemessen. Hierbei wurde im Zuge der Modellevaluierung eine Kreuzvalidierung durchgeführt, sodass die Modell-Performanz durch gemittelte *acc*-Werte dargestellt wird.

Zunächst werden mit *uniform* und *proportional* ACS-Strategien aus [13] vorgestellt, die nicht auf Betrachtungen der Modellgüte gründen. Währenddessen werden für *inverse*, *improvement* und *redistriction* die Klassenverhältnisse aktiv nach dem bisherigen Leistungsverhalten des Modells gewählt, wobei die Strategien jeweils unterschiedlichen *heuristischen* Annahmen unterliegen, die zu stark divergierenden Akquirierungen führen können.

6.1 uniform

Die *uniform* Strategie spricht in jeder Runde die Empfehlung aus, gleichverteilt von allen Klassen Dateninstanzen zu produzieren.

$$[\mathbf{p}_{[i]}]_y = \frac{1}{N} \cdot b_{[i]} \tag{6.1}$$

Die Strategie unterliegt der Annahme, dass die Fehlerraten der Klassen sehr ähnlich sind. Dass heißt, jede Klasse ähnlich schwer zu trainieren ist.

6.2 proportional

In *proportional* werden Datenbeispielen gemäß der initialen Klassenverteilung in D_1 produziert.

$$[\mathbf{p}_{[i]}]_y = [\mathbf{p}_{[i-1]}]_y \cdot b_{[i]} \tag{6.2}$$

Es wird angenommen, dass durch die Spezifikation von D_1 Domänenwissen reflektiert wurde, dass sich vorteilhaft auf das Training des Modells auswirkt. Die in D_1 festgelegten Klassenverhältnisse können sowohl die Schwierigkeit des Lernens einzelner Klassen als auch die Klassenverteilung im Deployment widerspiegeln, wobei letzteres häufig nur unter Unsicherheit bekannt ist.

6.3 inverse

Bei der Bewertung der Modellleistung zeigen Klassifikatoren oft unterschiedliche klassenbezogene Vorhersagefähigkeiten, was ein Hinweis sein kann, dass die Klassen unterschiedlich schwer zu lernen sind. Die Strategie *inverse* folgt der Idee, dass die Klassenverhältnisse des Trainingssatzes den Schweregrad der Klasse abbilden sollten. Die Grundannahme dabei ist, dass mehr Dateninstanzen die Lernfähigkeit der jeweiligen Klassen erhöhen, was die Modellleistung steigert. Zu diesem Zweck, werden in *inverse* umgekehrt proportional zum ermittelten klassenbedingten *acc*-Wert Empfehlungen ausgesprochen.

$$[\mathbf{p}_{[i]}]_y = \frac{\frac{1}{acc[y]}}{\sum_j^N \frac{1}{acc[j]}} \cdot b_{[i]} \tag{6.3}$$

6.4 improvement

In *improvement* bildet ebenfalls die bisherige Modell-Performanz die Entscheidungsgrundlage. Allerdings verfolgt die Strategie die Idee, dass Datenbeispiele aus Klassen produziert werden sollten, die das größte Verbesserungspotenzial bezüglich der Prognosefähigkeiten

besitzen. Dies basiert auf der Erwartung, dass für Klassen, für die in den vorherigen Runden nur geringe Veränderungen in der Modell-Perfomanz zu erkennen waren, weitere Klasseninstanzen nicht zu einer relevanten Verbesserung der Leistung führen. Sei $acc_{[i-1]}$ die ermittelten *acc*-Werte der Runde $i-1$, dann gilt:

$$[\mathbf{p}_{[i]}]_y = max\left(0, \frac{acc[y] - acc_{[i-1]}[y]}{\sum_{j=1}^{N} acc[j] - acc_{[i-1]}[j]}\right) \cdot b_{[i]} \tag{6.4}$$

6.5 redistricting

Die Idee hinter der Strategie *redistricting* lautet, dass vorzugsweise Dateninstanzen aus *instabilen* Klassen, die in der Nähe von volatilen Entscheidungsgrenzen liegen, generiert werden sollten. Für Instanzen aus instabilen Klassen ist es charakteristisch, dass betreffende Label-Vorhersagen eine hohe Varianz aufweisen. Dem entsprechend werden während der Akquise-Phasen instabile Klassen durch einen frequentierten Wechsel der Label-Vorhersagen bezüglich der Klasseninstanzen identifiziert. Sei $\text{redistricted}_i[y]$ die Anzahl der divergierenden Prognosen für die Instanzen der Klasse y im Vergleich zur vorherigen Runde, dann gilt:

$$[\mathbf{p}_{[i]}]_y = \frac{\frac{\text{redistricted}_i[y]}{p_y}}{\sum_{j=1}^{N} \frac{\text{redistricted}_i[j]}{p_y}} \tag{6.5}$$

wobei p_y das Klassenverhältnis der Trainingsdaten aus der Runde $i-1$ beschreibt.

Kapitel 7

Strategien unter unsicheren Deploymentbedingungen

Im vorherigen Teil der Arbeit wurde bereits analysiert, dass unter einem ACS-Setting eine Verschiebung der Klassenverteilung im Deployment zu einer Verschlechterung der Modell-Performanz führen kann. Der durch die Domänenverschiebung verursachte Fehler wurde durch Zertifizierungsmethoden abgeschätzt. Ebenfalls zeigte die Quantifizierung der *inter-domain-gap*, dass die aktive Wahl von $\mathbf{p}_{\mathcal{S}}$ die Domänenlücke konstituiert [6]. Dies hat zur Konsequenz, dass die Datenakquise nach ACS-Strategien einen unmittelbaren Effekt auf den *domain*-induzierten Fehler besitzt.

Die ACS-Strategien nach Lomasky et. al. [13] sind sich der Domänenlücke nicht bewusst. Stattdessen wird bei den meisten der vorgestellten heuristischen Strategien die Akquiseentscheidung auf Grundlage der bereits generierten ACS-Daten getroffen. Ausnahme stellt die Strategie *proportional* dar. In dieser Strategie besteht die Möglichkeit die wahre Deployment-Klassenverteilung durch den Anwender zu reflektieren, wodurch der *domain-gap* mit den damit verbundenen Folgeproblemen nicht mehr vorhanden wäre. In der Tat wurde bereits in den qualitativen Untersuchungen zur Domänenlücke festgestellt, dass *proportional* für $m \to \infty$ eine optimale Akquisitionsstrategie ist [7]. Dies gilt allerdings nur unter der Voraussetzung, dass die tatsächlichen Bedingungen im Deployment bekannt sind. In der Regel können Domänenexperten nur die Klassenverteilung im Deployment schätzen. Die Schätzungen sind naturgemäß mit einem Grad an Unsicherheit verbunden.

Motiviert von diesen Beobachtungen wurde bereits für binäre Klassifikationsprobleme eine ACS-Strategie vorgestellt, die sich der Domänenlücke bewusst ist und diese schrittweise durch Akquiseempfehlungen minimiert [5]. Dabei geschieht die Minimierung des *domain*-induzierten Fehlers auf Basis des in [6] eingeführten binären Zertifikats und unter Verwendung eines Prior, der die geschätzte Klassenverteilung von $\mathbf{p}_{\mathcal{T}}$ abbildet. Der Prior kann dabei von einem Domänenexperten mit beliebigen Grad an Unsicherheit gesetzt werden.

Analog zur Erweiterung der binären Zertifizierung auf Mehrklassen-Klassifikationsprobleme, besteht das Ziel Akquisestrategien abzuleiten, die in ähnlicher Weise die Domänenlücke bei Mehrklassen-Klassifikationsproblemen minimiert. Aufgrund der strukturellen Ähnlichkeit zwischen $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ und der binären Zertifizierung $\Delta p \cdot \Delta \ell_X$ bietet es sich für das Vorhaben an, Strategien auf der Grundlage der *Norm Certificates* abzuleiten.

7.1 ACS-Strategien auf Basis der *Norm Certificates*

Das wesentliche Vorgehen zur Herleitung der binären Zertifizierung [5] ist auf Mehrklassen-Klassifikationsprobleme übertragbar. Die Strategie hat das Ziel die Abschätzung der *inter*-domain-gap $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ zu minimieren. Dazu muss eine Prior-Verteilung $\hat{\mathbb{P}}$ gegeben sein, welche die Klassenverteilung im Deployment abbildet. Unter der Verwendung von $\hat{\mathbb{P}}$ kann der *domain*-induzierte Fehler durch

$$\epsilon^* = \int_V \hat{\mathbb{P}}(\mathbf{x} = \mathbf{p}_{\mathcal{T}}) \cdot \|\mathbf{p}_{\mathcal{S}} - \mathbf{x}\|_p \cdot \|\boldsymbol{\ell}_h\|_q^* \, d\mathbf{x} \tag{7.1}$$

über den Prior marginalisiert werden, wobei V einen N-dimensionaler Wahrscheinlichkeitsraum darstellt. Aufgrund der optimierten kleinsten oberen Schranke $\|\boldsymbol{\ell}_h\|_q^*$, ist 7.1 mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ eine obere Schranke des tatsächlichen marginalisierten Fehlers.

Der Fehler ϵ^* ist abhängig vom Akquisevektor $\mathbf{m} = (m_1, \ldots, m_N)$. Dies wird durch inspizieren von

$$\epsilon^*(\mathbf{m}) = \int_V \hat{\mathbb{P}}(\mathbf{x} = \mathbf{p}_{\mathcal{T}}) \cdot \|\mathbf{p}_{\mathcal{S}}(\mathbf{m}) - \mathbf{x}\|_p \cdot \|\boldsymbol{\ell}_h(\mathbf{m})\|_q^* \, d\mathbf{x} \tag{7.2}$$

ersichtlich. Hier ergibt sich die Klassenverteilung der ACS-generierten Daten $\mathbf{p}_{\mathcal{S}}$ mit $\mathbf{p}_{\mathcal{S}}(\mathbf{m}) = \frac{m}{\sum_1^N m_i}$ trivialerweise direkt aus $\mathbf{m}$. Für die kleinste obere Schranke wurden die jeweiligen Wahrscheinlichkeit δ_j bereits unter dem Budget δ optimiert, sodass $\|\boldsymbol{\ell}_h\|_q^*$ nur von $\mathbf{m}$ abhängt. Es gilt, dass je mehr Daten akquiriert werden, umso genauer kann die kleinste obere Schranke geschätzt werden. Folglich kann $\|\boldsymbol{\ell}_h\|_q^*$ durch $\mathbf{m}$ minimiert werden. Dies ist konträr zum Vorgehen beim Zertifizieren, in der $\mathbf{m}$ gegeben ist und δ_j optimiert werden muss.

Der Fehler ϵ^* soll in jeder Runde i schrittweise minimiert werden, indem $b_{[i]}$ Daten nach einer Verteilung $\mathbf{p}_{[i]}$ generiert werden. Die optimale Verteilung $\mathbf{p}_{[i]}$ bildet sich aus dem negativen Gradienten $-\Delta_{\mathbf{m}}\epsilon^* = -\left(\frac{\partial \epsilon^*}{\partial m_1}, \ldots, \frac{\partial \epsilon^*}{\partial m_N}\right)$ am aktuellen Vektor $\mathbf{m}$. Für $\mathbf{p}_{[i]}$ gilt:

$$[\mathbf{p}_{[i]}]_y = \frac{max(0, -[\Delta_{\mathbf{m}}\epsilon^*]_y)}{\sum_{i=1}^N max(0, -[\Delta_{\mathbf{m}}\epsilon^*]_i)} \tag{7.3}$$

Folglich wird die sukzessive Minimierung von ϵ^* durch ein Gradientenabstiegsverfahren mit einer Schrittweite von $b_{[i]}$ realisiert.

7.2 Gradientenberechnung

Der tiefste Gradientenabstieg am Punkt $\mathbf{m}$ kann durch die Produktregel berechnet werden:

$$\nabla_{\mathbf{m}}\epsilon^* = \nabla_{\mathbf{m}} f_p \cdot \|\boldsymbol{\ell}_h\|_q^* + f_p \cdot \nabla_{\mathbf{m}}\|\boldsymbol{\ell}_h\|_q^* \tag{7.4}$$

wobei

$$f_p(\mathbf{m}) = \int_V \hat{\mathbb{P}}(\mathbf{x} = \mathbf{p}_{\mathcal{T}}) \cdot \|\mathbf{p}_{\mathcal{S}} - \mathbf{x}\|_p \; d\mathbf{x} \tag{7.5}$$

Damit $\nabla_{\mathbf{m}}\epsilon^*$ nach der Produktregel gebildet werden kann, müssen Gradienten für f_p und $\|\boldsymbol{\ell}_h\|_q^*$ bestimmt werden.

Bestimmung von $\nabla_{\mathbf{m}}\|\boldsymbol{\ell}_h\|_q^*$

Für die kleinste oberen Schranke $\|\boldsymbol{\ell}_h\|_q^*$ konnten für $q = \{1, 2, \infty\}$ geschlossene partielle Ableitungen $\frac{\partial\|\boldsymbol{\ell}_h\|_q^*}{\partial m_y}$ gefunden werden, die die Bildung des Gradienten ermöglichen. Die Gradienten können wie folgt gebildet werden:

$q = 1$

$$\|\boldsymbol{\ell}_h(\mathbf{m})\|_1^* = \hat{\ell}(h,1) + \sqrt{\frac{-ln(\delta_1)}{2\; m_1}} + \cdots + \hat{\ell}(h,N) + \sqrt{\frac{-ln(\delta_1)}{2\; m_N}} \tag{7.6}$$

$$[\nabla_{\mathbf{m}}\|\boldsymbol{\ell}_h\|_1^*]_y = \frac{-\sqrt{ln(1/\sqrt{\delta_y})}}{2\; m_y^{2/3}} \tag{7.7}$$

$q = 2$

$$\|\boldsymbol{\ell}_h(\mathbf{m})\|_2^* = \sqrt{\left(\hat{\ell}(h,1) + \sqrt{\frac{-ln(\delta_1)}{2\; m_1}}\right)^2 + \cdots + \left(\hat{\ell}(h,N) + \sqrt{\frac{-ln(\delta_1)}{2\; m_N}}\right)^2} \tag{7.8}$$

$$[\nabla_{\mathbf{m}}\|\boldsymbol{\ell}_h\|_2^*]_y = \frac{\sqrt{2}\; ln(\delta_y) - 2\; \hat{\ell}(h,y)\sqrt{-m_y\; ln(\delta_y)}}{4\; m_y^{3/2}\sqrt{2\; \hat{\ell}(h,y)\left(\sqrt{2}\; \sqrt{-m_y\; ln(\delta_y)} + m_y\; \hat{\ell}(h,y)\right) - ln(\delta_y)}} \tag{7.9}$$

$q = \infty$

$$\|\boldsymbol{\ell}_h(\mathbf{m})\|_\infty^* = max\left(\left\{\hat{\ell}(h,1) + \sqrt{\frac{-ln(\delta_1)}{2\; m_1}}, \ldots, \hat{\ell}(h,N) + \sqrt{\frac{-ln(\delta_1)}{2\; m_N}}\right\}\right) \tag{7.10}$$

$$[\nabla_{\mathbf{m}}\|\boldsymbol{\ell}_h\|_\infty^*]_y = \begin{cases} \frac{-\sqrt{ln(1/\sqrt{\delta_y})}}{2\; m_y^{2/3}} & \text{falls } \hat{\ell}(h,y) + \epsilon_y = \|\boldsymbol{\ell}_h(\mathbf{m})\|_\infty^* \\ 0 & \text{sonst} \end{cases} \tag{7.11}$$

7.2.1 Berechnen von f_p und $\nabla_{\mathbf{m}} f_p$

Damit die Funktion $f_p(\mathbf{m})$ unter $\mathbf{m}$ evaluiert werden kann, muss zunächst das Integral $\int_V \hat{\mathbb{P}}(\mathbf{x} = \mathbf{p}_\mathcal{T}) \cdot \|\mathbf{p}_\mathcal{S} - \mathbf{x}\|_p \, d\mathbf{x}$ gelöst werden. Für eine analytische Auswertung des Integrals sollte deshalb eine Stammfunktion für f_p bekannt sein. Im Allgemeinen ist dies für f_p nicht möglich da insbesondere die Prior-Verteilungsfamilie von $\hat{\mathbb{P}}$ nicht vorgeschrieben ist. Stattdessen wird das Integral näherungsweise durch Verfahren der numerischen Integration gelöst. wodurch $f_p(\mathbf{m})$ für $p = \{1, 2, \infty\}$ berechnet werden kann. Wiederum muss der Gradient $\nabla_{\mathbf{m}} f_p$ approximativ bestimmt werden.

Zwar erfordert das numerische Lösen von f_p beziehungsweise $\nabla_{\mathbf{m}} f_p$ einen höheren Ressourcenaufwand und liefert nur näherungsweise Lösungen, allerdings erlaubt es Anwendern die Prior-Klassenverteilung flexibel zu wählen.

Numerische Integration

Das Lösen von $\int_V \hat{\mathbb{P}}(\mathbf{x} = \mathbf{p}_\mathcal{T}) \cdot \|\mathbf{p}_\mathcal{S} - \mathbf{x}\|_p \, d\mathbf{x} = \int_V g(\mathbf{x}) \, d\mathbf{x}$ adressiert das Problem der mehrdimensionalen Integration über einem Simplex-Raum V. Das Integral kann durch Verfahren aus der numerischen Quadratur näherungsweise bestimmt werden. Hierbei wird im Allgemeinen das Integral durch eine gewichtete Summe von Funktionswerten approximiert, die an $x_1, \ldots, x_n$ Stützstellen ausgewertet werden:

$$f_p = \int_V g(\mathbf{x}) \, d\mathbf{x} \approx \sum_{i=1}^{n} w_i g(x_i) \tag{7.12}$$

In der Regel unterscheiden sich die Verfahren aus der numerischen Quadratur hauptsächlich durch die Wahl der Gewichtung $w_1, \ldots, w_n$ und den Stützstellen.

Einige Methoden, wie die Gauß-Quadratur, approximieren g zunächst durch einen Interpolationspolynom $\mathbf{p}_k$ vom Grad k. Im Falle der Gauß-Quadratur hat dies den Vorteil, dass $\mathbf{p}_k$ exakt integriert werden kann, so dass gilt:

$$f_p = \int_V g(\mathbf{x}) \, d\mathbf{x} \approx \int_V \mathbf{p}_k(\mathbf{x}) \cdot w(\mathbf{x}) \, d\mathbf{x} = \sum_{i=1}^{n} w_i \mathbf{p}_k(x_i) \tag{7.13}$$

Alternativ bietet sich die Monte-Carlo-Integration für die Lösung von f_p an. In seiner einfachsten Form werden im Monte-Carlo Verfahren die Stützstellen $x_1, \ldots, x_n$ durch gleichverteiltes Ziehen von n Vektoren bestimmt. Das Integral kann mit $Q = \int_V \, d\mathbf{x}$ durch

$$Q \frac{1}{n} \sum_{i=1}^{n} g(x_i) \tag{7.14}$$

approximiert werden. Die Genauigkeit der Approximation hängt von der Anzahl n der verwendeten Stützstellen ab. So gilt nach dem Gesetz der Großen Zahlen:

$$f_p(\mathbf{m}) = \lim_{n \to \infty} Q \frac{1}{n} \sum_{i=1}^{n} g(x_i) \tag{7.15}$$

Die Monte-Carlo-Integration wird besonders für hochdimensionale Integrale empfohlen [14], da diese geringer unter dem Fluch der Dimensionalität leiden.

Approximative Gradientenberechnung

Damit $\nabla_{\mathbf{m}}\epsilon^*$ bestimmt werden kann, muss für ein Vektor $\mathbf{m}$ ein Gradient $\nabla_{\mathbf{m}}f_p$ ermittelt werden. Dies geschieht auf Grundlage der näherungsweise berechneten Funktion f_p. Für die Approximation des Gradienten können Finite-Differenzen Verfahren (engl. *finite difference*) verwendet werden. Bei diesem Vorgehen wird der Differenzquotient

$$\lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \tag{7.16}$$

nummerisch approximiert. Bezogen auf das Berechnen von $\nabla_{\mathbf{m}}f_p$ können die partiellen Ableitungen $\frac{\partial f_p}{\partial m_y}$ durch

$$\frac{\partial f_p}{\partial m_y} = f_p(m_1,\ldots,m_y+h,\ldots,m_n) - f_p(\mathbf{m}) \tag{7.17}$$

mittels Vorwärtsdifferenzen geschätzt werden.

Kapitel 8

Evaluation von ACS-Strategien

In der abschließenden Evaluation wird das Leistungsvermögen der vorgestellten ACS-Strategien unter dem Einfluss von Label-Verschiebungen verglichen. Zunächst wird der Versuchsaufbau der Experimente vorgestellt.

8.1 Design der Experimente

8.1.1 Experimenteller Aufbau

Der experimentelle Aufbau ist analog zur Evaluation der Strategie in [5], die für binäre Klassifikationsprobleme die *domain-gap* minimiert. Die Leistungsfähigkeit der hier diskutierten Strategien werden anhand eines gegebenen ACS-Settings evaluiert, die die Zusammensetzung der initialen Trainingsmenge, sowie die Akquisephasen bezüglich der Anzahl und Größe der Batches festlegt.

Jedes Experiment durchläuft die definierten Akquisephasen und bestimmt in jeder Runde die Modellleistung des Klassifikators, der auf den generierten Daten trainiert wurde. Der Datensatz D wird in den Datenmengen D_{pool} und $D_{\text{remaining}}$ aufgeteilt. Die Menge D_{pool} umfasst die potenziellen Dateninstanzen, die *generiert* werden können. Folglich werden die Trainingsinstanzen nicht durch einen tatsächlichen Generator produziert, stattdessen wird das Generieren der Instanzen durch die Ziehung aus D_{pool} simuliert.

Da die Evaluation insbesondere die Leistungsfähigkeit der Strategien in Abhängigkeit zu den Klassenverhältnissen im Deployment analysiert, wird eine Testmenge D_{tst} mit einem Klassenverhältnis von $\mathbf{p}_{\mathcal{T}}$ aus $D_{\text{remaining}}$ gezogen. Demnach repräsentiert D_{tst} die Deployment-Daten, auf denen das ACS-trainierte Modell getestet wird.

Als Klassifikator wurde für die Evaluation die Logistische Regression mit einer Standardkonfiguration verwendet, wobei die Modell-Performanz durch die *accuracy* approximiert wurde. Insgesamt gab es zehn Runden, in denen eine Batch von 50 Dateninstanzen nach den Empfehlungen der untersuchten Strategie aus D_{pool} gezogen wurden. Die Ausnahme ist der initiale Trainingssatz, bei dem jede Klasse durch 20 Datenbeispiele repräsentiert

wurde. Damit robuste Fehlereinschätzungen gewährleistet sind, wurden die Experimente 15 mal unter jeweils verschiedenen Datenaufteilungen wiederholt.

8.1.2 Auswertung

Die Evaluation hat das Ziel, das Leistungsvermögen der diskutierten ACS-Strategien zu vergleichen und in einer Rangfolge einzuordnen. Dies geschieht auf Grundlage der rundenweise ermittelten Modell-Performanz. Dabei ist es von großer Bedeutung, dass signifikante Unterschiede in der Modellleistung identifiziert werden können. Andernfalls könnten ausgewiesene Leistungsunterschiede zwischen den Methoden zufällig entstanden sein. Ebenso zielt die Bewertung darauf ab, robuste Schlussfolgerungen über das Verhalten von ACS-Strategien auf der Grundlage einer großen Anzahl von ausgewerteten Datensätzen zu gewinnen.

Eine komfortable Möglichkeit signifikante Unterschiede der Performanz zwischen den Methoden darzustellen, bieten die *critical difference diagrams* [2] [9], die ebenfalls für die Auswertungen in [5] verwendet wurden. Das *critical difference diagram* ermöglicht eine kompakte Visualisierung von signifikanten Leistungsunterschieden von ACS-Strategien über mehrere Datensätze. Hierbei können für jede Batch-Runde die ermittelten Ränge der zu vergleichenden Strategien visualisiert werden und auf signifikante Unterschiede inspiziert werden. Die Signifikanz zwischen dem Leistungsverhalten der ACS-Strategien wird anhand des *Wilcoxon-Vorzeichen-Rang-Test* paarweise ermittelt.

8.1.3 Datensätze

Damit signifikante Resultate möglich sind, müssen die ACS-Strategien auf vielen Datensätzen getestet werden. Folglich wird die Datenauswahl, die für die Evaluierung der Zertizierungsmethoden verwendet wurde, um die Datensätze aus Tabelle 8.1 erweitert. Dem entsprechend werden die ACS-Strategien auf insgesamt 17 Datensätzen evaluiert.

8.1.4 Wahl des Priors

Für die durchgeführten Experimente wurde eine mehrdimensionale Normalverteilung als Prior $\hat{\mathbb{P}}$ gewählt. Die Dichtefunktion der Normalverteilung ist für einen Erwartungswertvektor μ und einer Kovarianzmatrix $\mathbf{\Sigma}$ durch

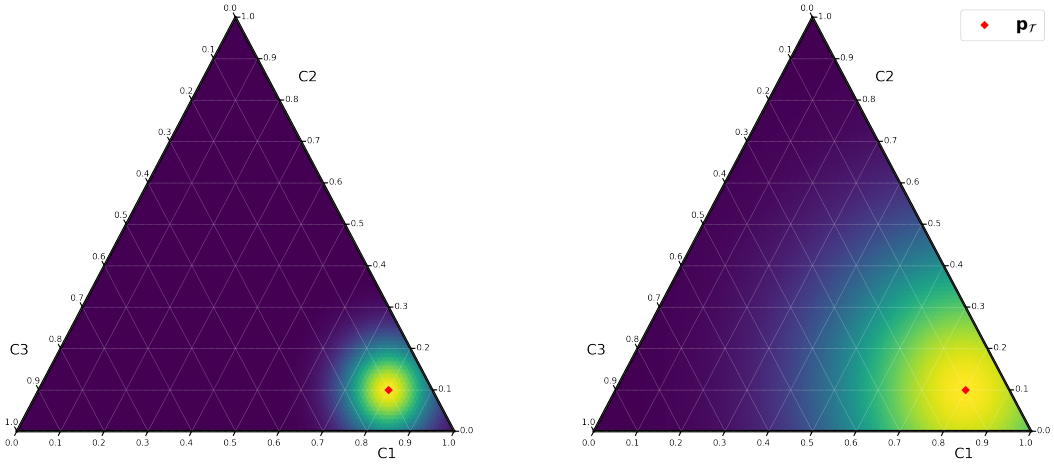
$$f_{\text{normal}}(\mathbf{x}) = \frac{1}{\sqrt{(2\ \pi)^N\ det(\mathbf{\Sigma})}}\ exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^{T\ \mathbf{\Sigma}^{-1}\ (\mathbf{x}-\mu)}\right) \tag{8.1}$$

gegeben. Der Anwender beziehungsweise Domänenexperte kann die Deployment-Klassenverteilung durch Spezifikation von μ schätzen, wobei die Unsicherheit durch $\mathbf{\Sigma}$ kontrolliert wird. Dadurch kann die Prior-Verteilung intuitiv für die Schätzung der Deployment-Klassenverteilung verwendet werden.

Datensatz	#Features	m_y	#Instanzen	$\mathbb{P}(Y = y)$
connect 4	42	$[44473, 6449, 16635]$	67557	$[0,65; 0,10; 0,25]$
waveform 5000	40	$[1655, 1692, 1653]$	5000	$[0,33; 0,34; 0,33]$
cmc	9	$[629, 333, 511]$	1473	$[0,42; 0,23; 0,35]$
dna	180	$[1654, 767, 765]$	3186	$[0,52; 0,24; 0,24]$
wall robot navigation	24	$[826, 2097, 2533]$	5456	$[0,15; 0,38; 0,46]$
bng vehicle	18	$[508281, 256101, 235618]$	1000000	$[0,51; 0,26; 0,24]$
jannis	54	$[38522, 28790, 16421]$	83733	$[0,46; 0,34; 0,2]$
nursery	8	$[4266, 4320, 4372]$	12958	$[0,33; 0,33; 0,34]$
har	561	$[3917, 4034, 2049]$	10299	$[0,39; 0,4; 0,2]$
first order theorem proving	51	$[3643, 1365, 1110]$	6118	$[0,6; 0,22; 0,18]$
wine quality white	11	$[2198, 1457, 1243]$	4898	$[0,6; 0,22; 0,18]$

Tabelle 8.1: Die Datensätze stammen von der Plattform *openML* und erweitern die Gesamtmenge an Datensätzen. Die Datensätze wurden nach keinen bestimmten Kriterien ausgewählt, außer dass sie eine ausreichende Anzahl von Dateninstanzen enthalten.

Alternativ können auch andere multivariate Verteilungen eingesetzt werden, wie beispielsweise die Dirichletverteilung.



Schätzung mit geringer Streuung

Schätzung unter hoher Unsicherheit

Abbildung 8.1: In Form einer mehrdimensionalen Normalverteilung kann der Prior die Deployment-Klassenverteilung schätzen, indem er den Erwartungswertvektor μ unter einer beliebigen Streuung (Unsicherheit) festlegt.

8.2 Durchführung

8.2.1 Heuristische Strategien unter wechselnden Deploymentbedingungen

Zunächst wird das Leistungsverhalten von ACS-Strategien unter verschiedenen Deployment-Klassenverteilungen betrachtet. Wenn die Anzahl der Klasseninstanzen im Deployment gleichmäßig verteilt ist, dann entsprechen sie der Klassenzusammensetzung des anfänglichen Trainingssatzes.

Das dargestellte *critical difference diagram* in Abbildung 8.2 fasst die Ergebnisse zur Auswertung der heuristischen ACS-Strategien für $\mathbf{p}_{\mathcal{T}} = (0.33, 0.33, 0.33)$ zusammen. Danach weisen die meisten der untersuchten ACS-Strategien keine signifikanten Leistungsunterschiede für gleichverteilte Klassen auf. Daraus folgt, dass keine der heuristischen Strategien bei der Gesamtzahl der ausgewerteten Datensätze überlegen war. Die Ausnahme ist die Strategie *improvement*, die im Vergleich zu den restlichen ACS-Strategien signifikant schlechter abschnitt.

Die durch *improvement* getroffenen Akquise-Entscheidungen führten ebenfalls zu großen Abweichungen bezüglich den gleichverteilten Klassenproportionen der Anwendung. Die übrigen ACS-Strategien führten durchschnittlich zu Trainingsmengen, die näherungsweise gleichverteilt waren.

Im nächsten Schritt wird die Deployment-Verteilung auf $\mathbf{p}_{\mathcal{T}} = (0.7, 0.2, 0.1)$ gesetzt, wodurch eine Label-Verschiebung zur gleichverteilten initialen Trainingsmenge verursacht wird. Beim Inspizieren der Ergebnisse in Abbildung 8.4 wird ersichtlich, dass die Strategie *proportional* die dominierende Strategie ist. Somit erzielt die Strategie, die die tatsächlichen Deployment-Klassenverhältnisse widerspiegelt, im Vergleich zu den übrigen ACS-Strategien deutlich bessere Resultate. Nicht immer sind jedoch die wahren Klassenverhältnisse exakt bekannt, wodurch die Strategie *proportional* im Allgemeinen nicht anwendbar ist.

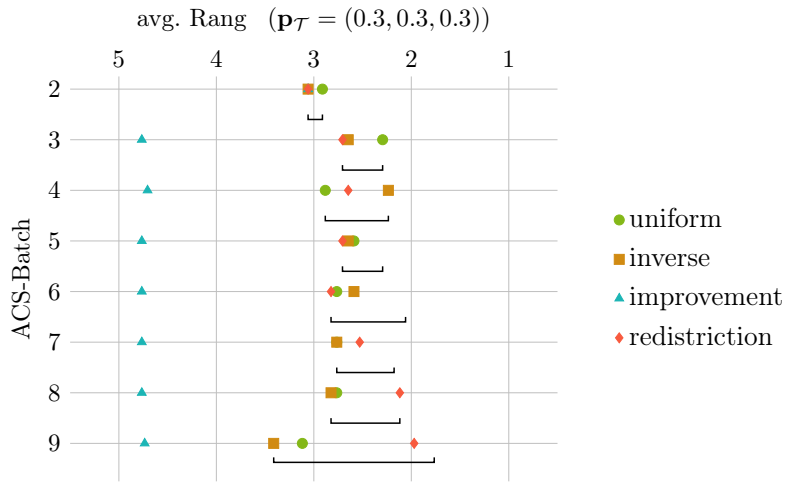


Abbildung 8.2: Die Repräsentation der Ergebnisse für eine gleichverteilte Klassenverteilung durch *critical difference diagrams*. Jede vertikale Achse korrespondiert mit einem *critical difference diagram* in Abhängigkeit zur aktuellen Runde (Batch). Die horizontalen Positionen korrespondieren mit den durchschnittlichen Rängen der ACS-Strategien die durch mehrere Datensätze ermittelt wurden sind. Strategien, die nach dem Wilcoxon Signed-Rank-Test nicht signifikant sind, werden durch horizontale Verbindungen symbolisiert.

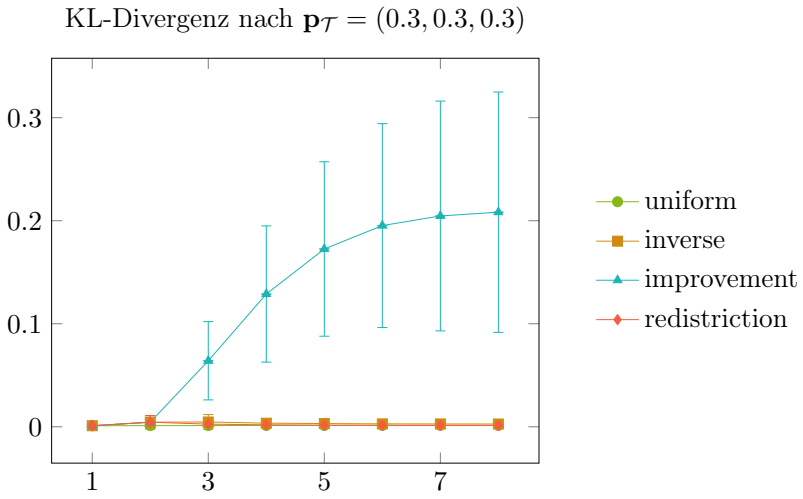


Abbildung 8.3: In Abhängigkeit zur Akquisephase (horizontale Achse) zeigt der Graph die durchschnittliche Divergenz der Klassenverhältnisse zwischen ACS-generierten Daten und der Deployment-Verteilung für Abbildung 8.2. Bewertet werden die Abweichungen durch die *Kullback-Leibler-Divergenz* (vertikale Achse).

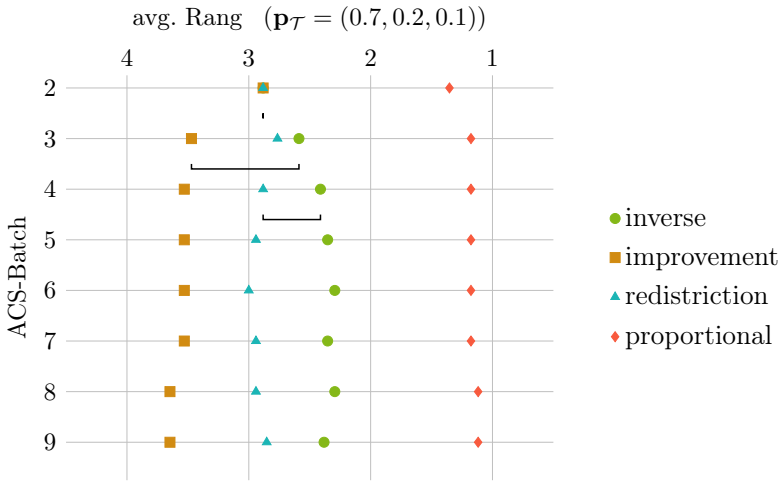


Abbildung 8.4: Die Strategie *proportional* ist bei Label-Verschiebungen im Deployment den übrigen Strategien, die sich der Domänenlücke nicht bewusst sind, klar überlegen.

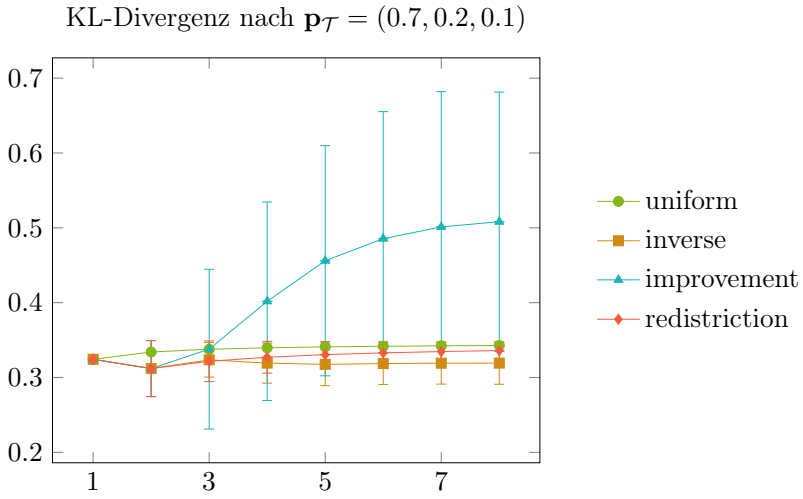


Abbildung 8.5: Die heuristischen Strategien sind sich der Domänenverschiebung in Abbildung 8.4 nicht bewusst. Dementsprechend verbleibt die entstandene *domain-gap* über die Akquisephasen. Die Strategie *proportional* hätte in dieser Darstellung eine konstante KL-Divergenz von 0.

8.2.2 Strategien zur Minimierung von Domänenlücken

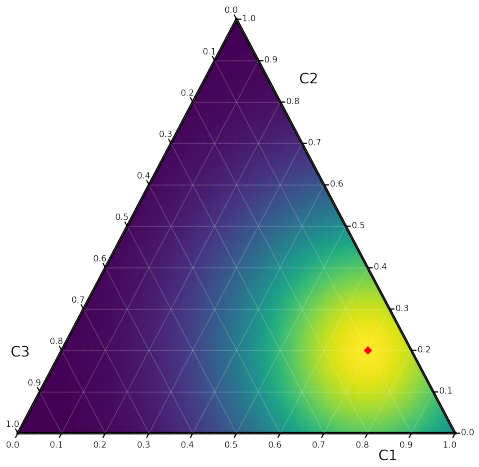
Die vorangegangene Reihe von Experimenten zeigt, dass bei einer gegebenen Label-Verschiebung die Strategie *proportional*, die Daten gemäß der Deployment-Verteilung akquiriert, den anderen Strategien überlegen war. Im Folgendem wird das Leistungsvermögen der Strategien $domaingap(p, q)$ analysiert, die die Domänenlücke auf Grundlage der Abschätzung $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q^*$ nachdem in Kapitel 7 beschriebenen Verfahren sukzessive minimieren. Hierfür wurde die kleinste obere Schranke $\|\boldsymbol{\ell}_h\|_q^*$ unter einem Wahrscheinlichkeitsbudget von $\delta = 0.05$ optimiert.

Im Gegensatz zu *proportional* ist die tatsächliche Klassenzusammensetzung in der Anwendung den $domaingap(p, q)$ Strategien nicht bekannt, sondern müssen durch die Spezifikation eines Priors geschätzt werden. Folglich ist zu vermuten, dass die Genauigkeit der Schätzung von $\mathbf{p}_\mathcal{T}$ einen direkten Effekt auf das Leistungsvermögen der $domaingap(p, q)$ Strategien besitzt. Es werden daher insgesamt drei verschiedene Prior ($\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$) getestet, die jeweils unter Unsicherheit die Deployment-Klassenverteilung mit unterschiedlicher Genauigkeit schätzen. Siehe dazu Abbildung 8.6.

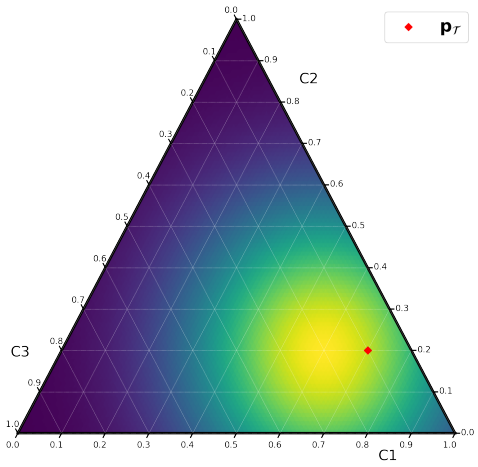
Die Ergebnisse für die $domaingap(p, q)$ Strategie unter den Schätzungen $\mathbf{A}$,$\mathbf{B}$ und $\mathbf{C}$ sind in den Abbildungen 8.7 bis 8.12 einsehbar. Die Resultate zeigen, dass die Strategie *proportion* den durchschnittlich führenden Rang einnimmt. Im Gegensatz zu den erreichten Ergebnissen der heuristischen Strategien, fallen die Leistungsunterschiede zwischen den Strategien $domaingap(p, q)$ und *proportion* gering aus. In einigen Akquisephasen sind die erreichten Performanz-Werte so marginal, dass sie nach dem Wilcoxon-Vorzeichen-Rang-Test nicht mehr signifikant sind.

Erwartungsgemäß beeinflusste die Qualität der Schätzung des Priors die erzielte Leistung der Strategien. So wurden unter dem exakten Schätzer $\mathbf{A}$ ($\mu = \mathbf{p}_\mathcal{T}$) die besten Ergebnisse für alle Methoden erzielt. Dicht gefolgt von den Strategien $domaingap(p, q)_B$, die unter einem leicht verzerrten Prior betrieben wurden. Die stärker abweichende Schätzung ($\mathbf{C}$) beeinträchtigte das Leistungsvermögen der Strategien erheblich.

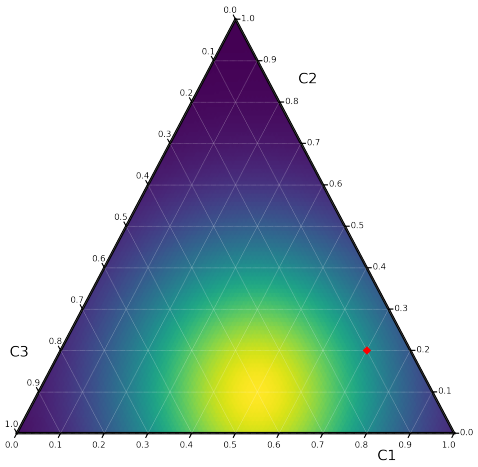
Alle Strategien zeigten ein gemeinsames Konvergenzverhalten bezüglich $\mathbf{p}_\mathcal{T}$. Ausgehend vom initialen Trainingsdatensatz strebt die erste Akquise-Empfehlung der Strategien stark zum gesetzten Prior. Aufgrund der relativ klein gewählten Anfangsmenge in Verbindung mit einer vergleichsweise großen Batch-Größe, erreicht die Strategie zumeist in einem Schritt ein Klassenverhältnis, welches nahe beim Erwartungswertvektors des gegebenen Priors ist. Im nächsten Schritt divergieren die Klassenverhältnisse der ACS-generierten Daten wieder von μ und nähern sich in den folgenden Runden in oszillierenden Bewegungen wieder dem Erwartungswertvektor an. Dieses Verhalten ist insbesondere auf die hoch gewählte Unsicherheit des Prior zurückzuführen.



A: $\mu = (0.7, 0.2, 0.1)$



B: $\mu = (0.6, 0.2, 0.2)$



C: $\mu = (0.5, 0.1, 0.4)$

Abbildung 8.6: In der Form einer mehrdimensionalen Normalverteilung wurde der eingesetzte Prior unter konstanter Streuung (Unsicherheit) durch die Setzung des Erwartungswertvektors μ spezifiziert. Die Kategorien **A**, **B** und **C** spiegeln unterschiedliche Schätzgenauigkeiten bezüglich $\mathbf{p}_\mathcal{T} = (0.7, 0.2, 0.1)$ wieder.

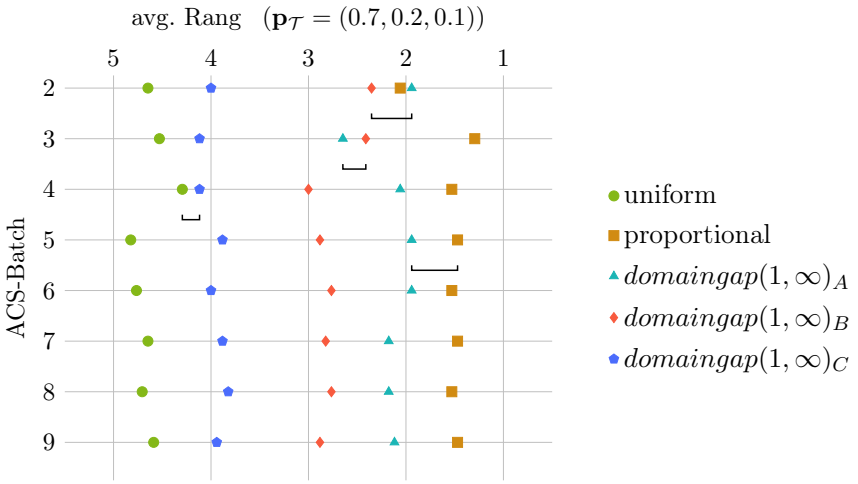


Abbildung 8.7: *Critical difference diagram* für $domaingap(1, \infty)$.

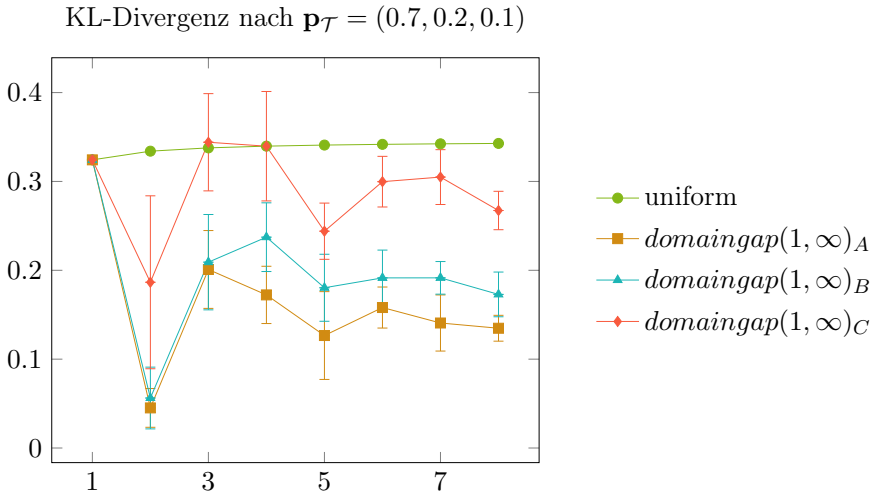


Abbildung 8.8: KL-Divergenz nach $\mathbf{p}_\mathcal{T}$ für $domaingap(1, \infty)$.

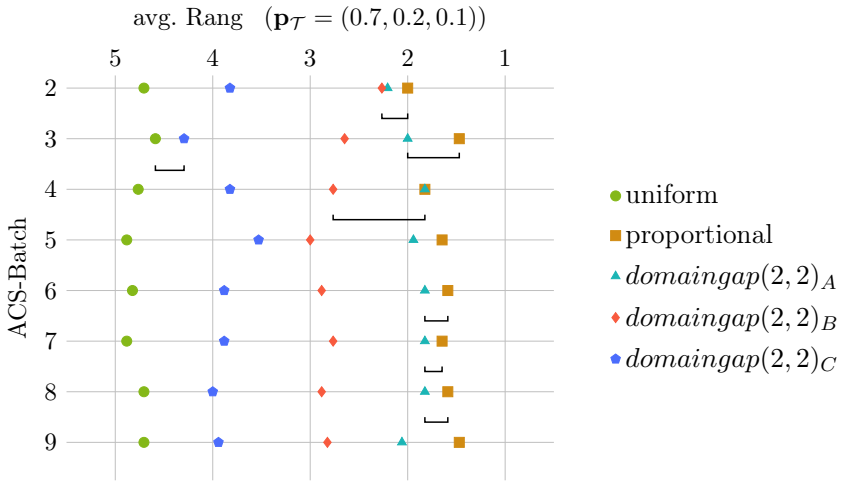


Abbildung 8.9: *Critical difference diagram* für $domaingap(2, 2)$.

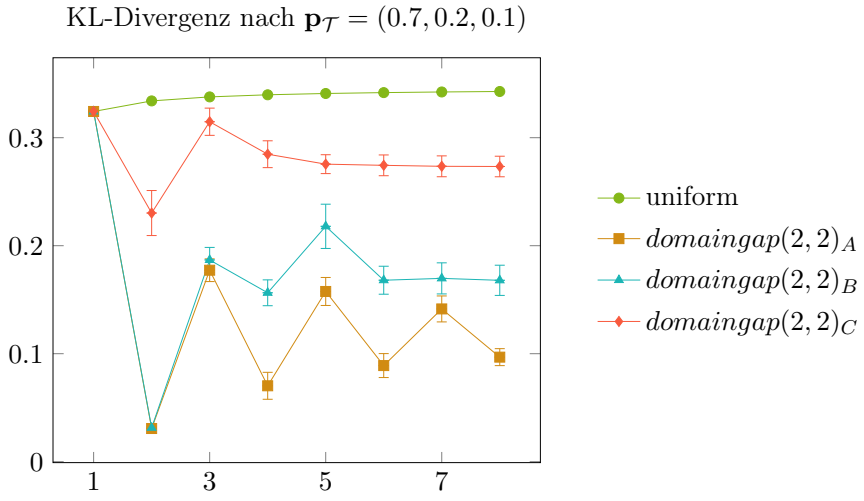


Abbildung 8.10: KL-Divergenz nach $\mathbf{p}_\mathcal{T}$ für $domaingap(2, 2)$.

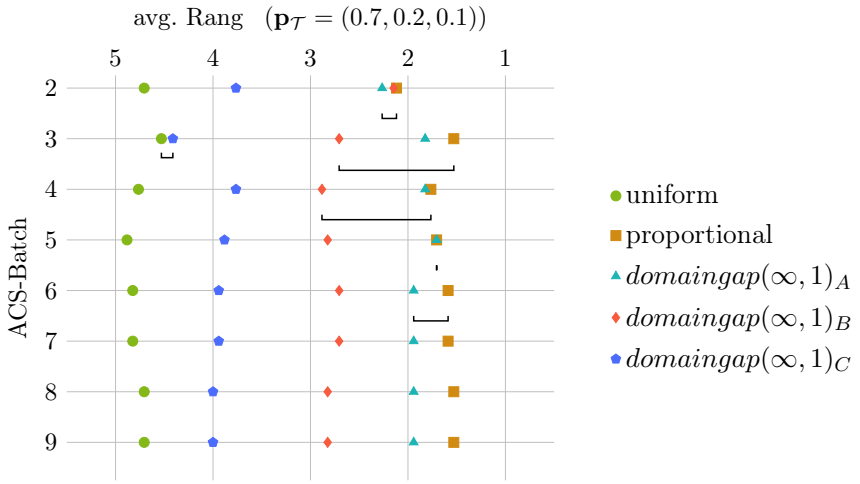


Abbildung 8.11: *Critical difference diagram* für $domaingap(\infty, 1)$.

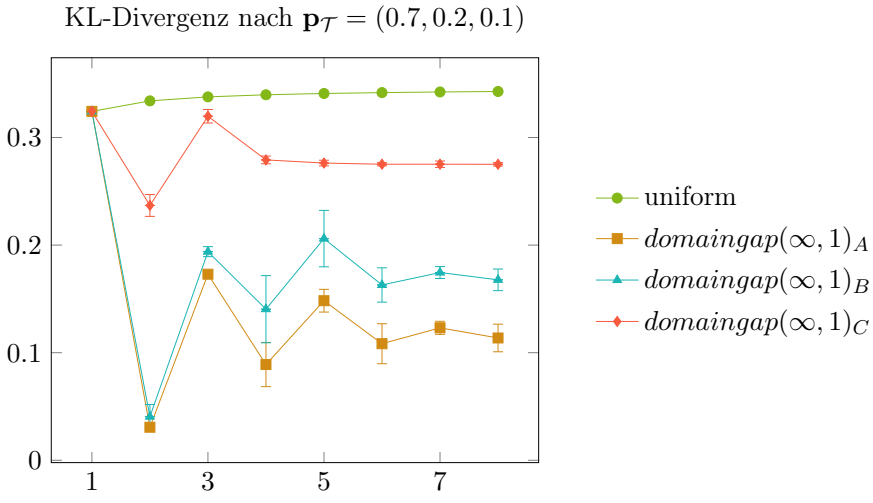


Abbildung 8.12: KL-Divergenz nach $\mathbf{p}_\mathcal{T}$ für $domaingap(\infty, 1)$.

8.2.3 Kombination von ACS-Strategien

Die gesammelten Ergebnisse zeigen, dass die Strategie $domaingap(p, q)$ unter einen näherungsweise akkuraten Prior den heuristischen Strategien überlegen ist. Die $domaingap(p, q)$ Strategien sind mit geringen Aufwand mit beliebig anderen ACS-Strategien kombinierbar.

Hierfür wird die Funktion in 7.5 definierte Funktion f_p um einen zusätzlichen Prior $\hat{\mathbb{P}}_{\text{ACS}}$ erweitert:

$$f_p(\mathbf{m}) = \int_V \hat{\mathbb{P}}_{\text{ACS}}(\mathbf{x} = \mathbf{p}_{\text{acs}}) \cdot \hat{\mathbb{P}}(\mathbf{x} = \mathbf{p}_{\mathcal{T}}) \cdot \|\mathbf{p}_{\mathcal{S}} - \mathbf{x}\|_p \, d\mathbf{x} \tag{8.2}$$

Da die Funktion f_p numerisch bestimmt wird, ist die dargestellte Erweiterung direkt realisierbar.

Der Prior $\hat{\mathbb{P}}_{\text{ACS}}$ reflektiert die Akquise-Entscheidung einer beliebig anderen ACS-Strategie. Dabei lässt sich $\hat{\mathbb{P}}_{\text{ACS}}$ als ein zusätzliches Gewicht interpretieren, das je nach kombinierter Strategie die geschätzte Schwierigkeit der zu lernenden Klassen mit einbezieht.

Im Fall, dass $\hat{\mathbb{P}}_{\text{ACS}}$ aus der Familie der multivariaten Normalverteilung stammt, wird der Erwartungswertvektor auf die Akquise-Empfehlung der kombinierten Strategie gesetzt. Für die Bestimmung der Varianz durch die Kovarianzmatrix Σ ergeben sich viele Freiheitsgrade.

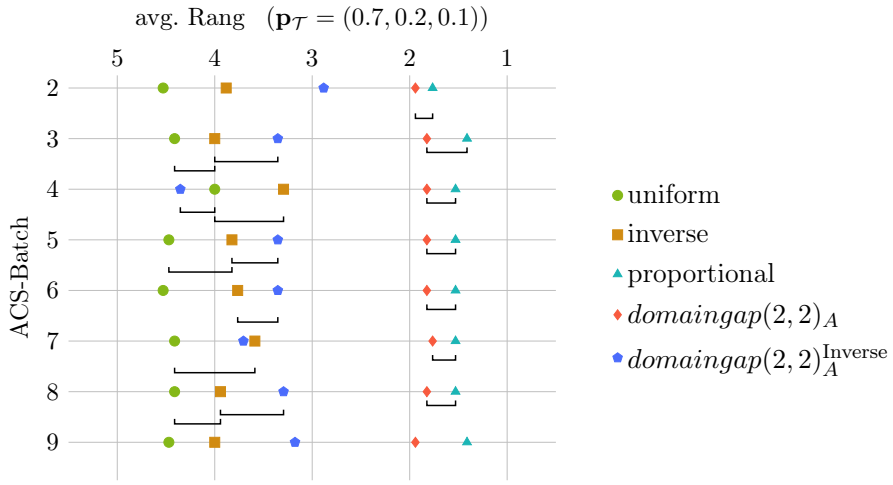


Abbildung 8.13: Beispielhafte Kombination der heuristischen Strategie *inverse* mit der Strategie $domaingap(2, 2)$. Die Kombination $domaingap(2, 2)_A^{\text{Inverse}}$ erzielte geringfügig bessere Resultate als die *inverse* Strategie. Allerdings waren die Verbesserungen in den meisten Akquise-Runden nicht signifikant.

8.2.4 Limitierungen der Evaluation

Die Strategien wurden nur auf Datensätzen getestet, die drei Klassen umfassten. Zum Leistungsverhalten bei beliebig vielen Klassen kann daher zu diesem Zeitpunkt keine Aussage

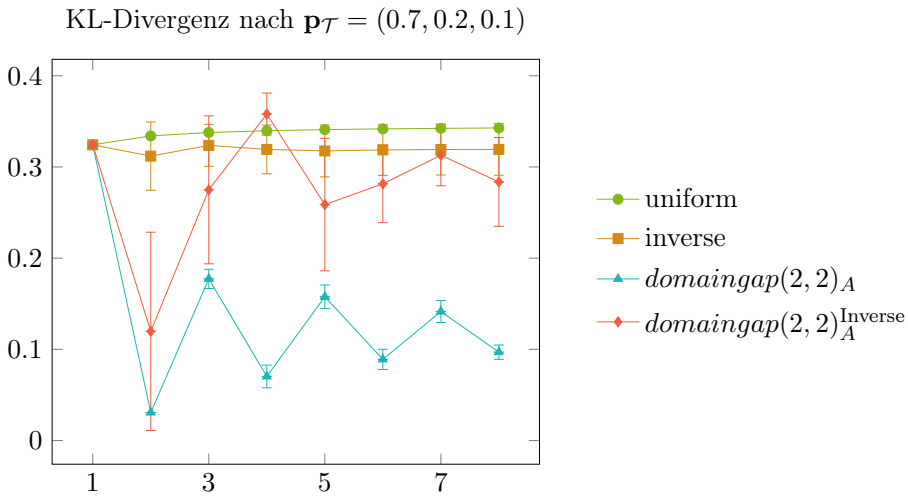


Abbildung 8.14: Das Konvergenzverhalten zu Abbildung 8.13. Die jeweiligen Gewichtungen von $domaingap(2,2)_A$ und *inverse* sind am Akquise-Verhalten von $domaingap(2,2)_A^{\text{Inverse}}$ erkennbar.

getroffen werden. Die Entscheidung, nur Datensätze mit drei Klassen zu evaluieren, begründet sich in dem Laufzeitverhalten mancher Methoden, welches mit steigender Klassenzahl lange Laufzeiten der Experimente bedeutet hätte. Ansonsten illustrieren die hier betrachteten Drei-Klassenprobleme die Funktionsweise der Mehrklassen-Erweiterungen ausreichend. Eine Evaluation mit mehr Klassen bleibt für zukünftige Arbeiten offen.

Bei den durchgeführten Experimenten wurde die Laufzeit für eine Akquise-Empfehlung durch $domaingap(p,q)$ von der Optimierung der kleinsten oberen Schranke $\|\boldsymbol{\ell}_h\|_q^*$ klar dominiert. Währenddessen konnten die numerisch gelösten f_p und ∇f_p in Sekundenbruchteilen durch eine Gauß-Quadratur in Verbindung mit *finite differences* berechnet werden. Bedingt durch die Restriktion der Klassenanzahl, kann allerdings keine Aussage zum Laufzeitverhalten und zu den Ressourcenanforderungen bei Klassifikationsproblemen mit einer beliebigen Anzahl von Klassen getroffen werden.

An dieser Stelle sei erneut darauf hingewiesen, dass es sich bei den Ergebnissen um eine Mittelung aus einer großen Anzahl von ausgewerteten Datensätzen handelt. Dadurch haben die Ergebnisse einen robusten Charakter, aus dem sich verallgemeinernde Erkenntnisse gewinnen lassen. Es ist jedoch wichtig zu beachten, dass die Leistung von ACS-Strategien je nach Datensatz stark variieren kann, so dass eine umfassende Analyse der Methoden für einzelne, bestimmte Anwendungen unerlässlich bleibt.

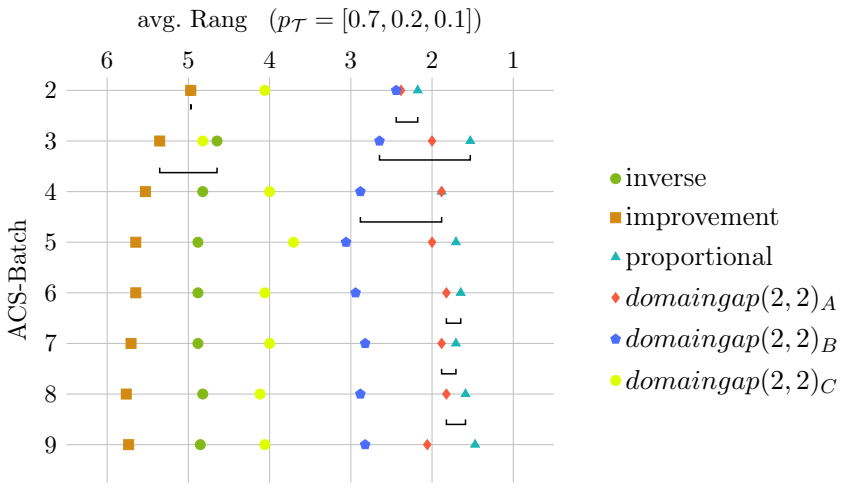


Abbildung 8.15: Bei einer Label-Verschiebung sind die heuristischen Strategien signifikant unterlegen.

Kapitel 9

Fazit und Ausblick

Die durchgeführte Evaluation verdeutlicht, dass die Deployment-Verteilung der Klassen die Leistungsfähigkeit von ACS-Strategien erheblich negativ beeinträchtigen kann. So zeigten Strategien, welche die resultierende Domänenlücke bei Akquise-Empfehlungen ignorierten ein unterlegendes Leistungsverhalten.

Mit der Einführung der Strategie $domaingap(p, q)$ ist es gelungen ein Verfahren zu entwickeln, dass auf Basis von $\|\mathbf{d}\|_p \cdot \|\boldsymbol{\ell}_h\|_q$ den prognostizierten *domain*-induzierten Fehler durch Akquise-Entscheidungen minimiert. Hierbei muss ein Prior spezifiziert werden, der die tatsächlichen Klassenverhältnisse im Deployment unter einem beliebigen Grad an Unsicherheit schätzt. Die Ergebnisse zeigen, dass die Schätzung von $\mathbf{p}_{\mathcal{T}}$ nicht exakt sein muss, damit $domaingap(p, q)$ zufriedenstellende Resultate erzielt.

Die gewonnenen Erkenntnisse aus der Evaluierung bestätigten die Notwendigkeit, dass ACS-Strategien Deploymentbedinungen in der Form von Label-Verschiebungen bei Akquise-Empfehlungen berücksichtigen sollten. Viele der populären ACS-Akquisestrategien gründen ihre Empfehlung einzig auf die approximierte Schwierigkeit eine Klasse für einen gegebenen Klassifikator zu lernen. Folglich könnte die Leistungsfähigkeit von bestehenden Strategien gesteigert werden, wenn auftretende Klassenverschiebungen reflektiert und kontrolliert werden.

Das beschriebene Verfahren zur Kombination von ACS-Strategien gibt einen Ausblick darauf, wie die Integration von geschätzten Deployment-Klassenverteilungen beispielhaft realisiert werden könnte. Darauf aufbauend sind viele Weiterentwicklungen möglich.

Anhang A

Anhang

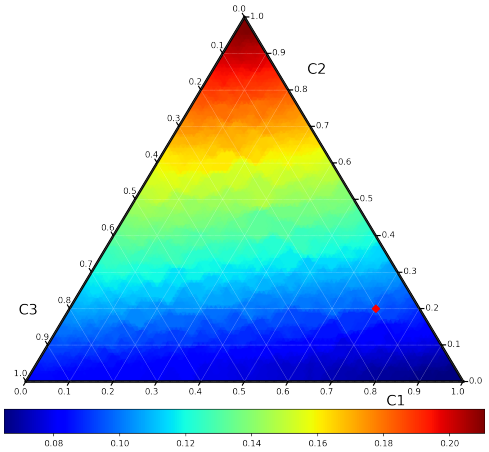
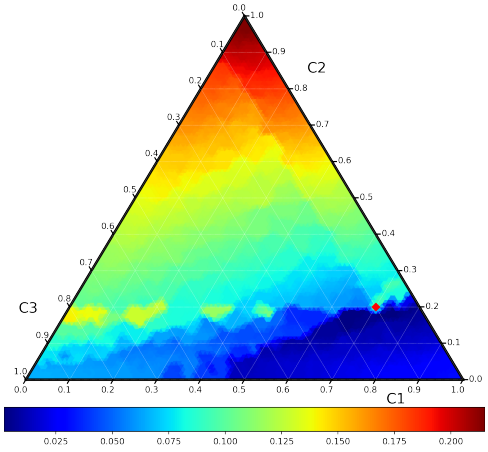
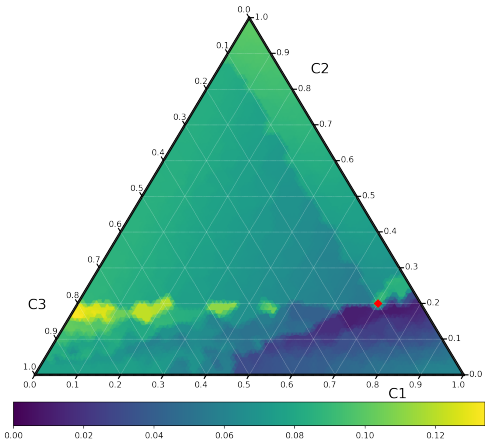


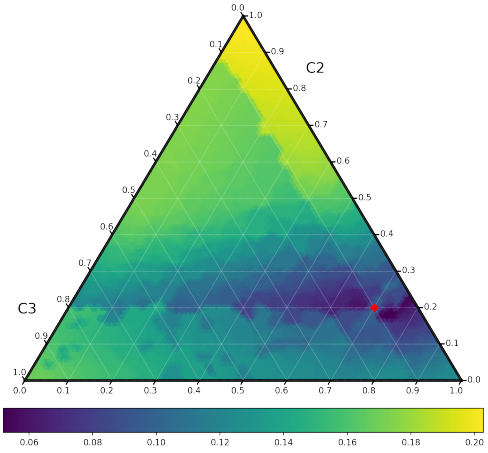
Abbildung A.1: optdigits: $L_{\mathcal{T}} + \epsilon_{\mathcal{T}}$, 01-Loss



Zertifizierter Fehler ϵ

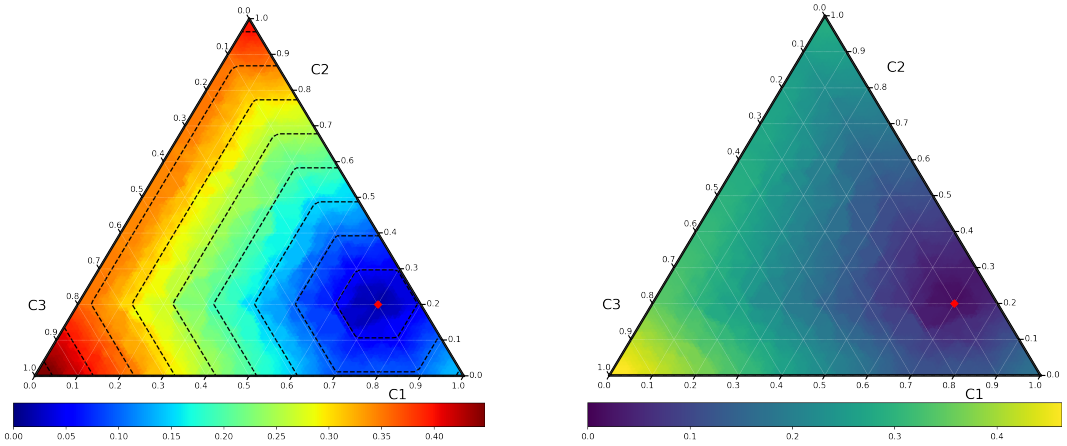


$|\hat{L}_S + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$



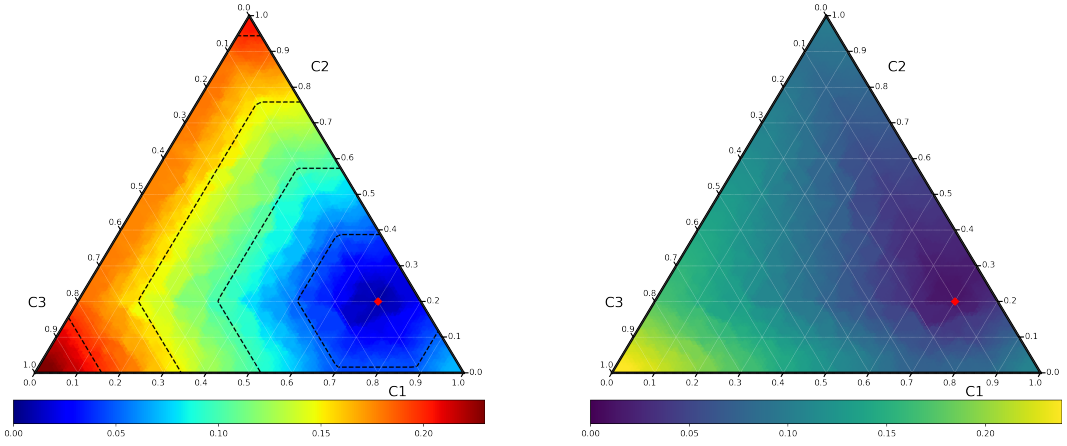
Breite des Konfidenzintervalls

Abbildung A.2: optdigits: Signed Certificate, Logistische Regression, 01-Loss



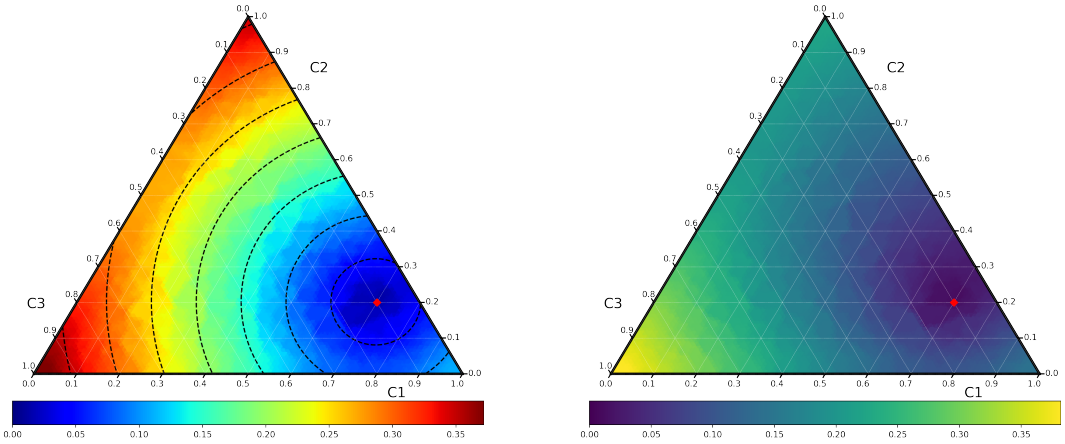
Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.3: optdigits: $||P||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



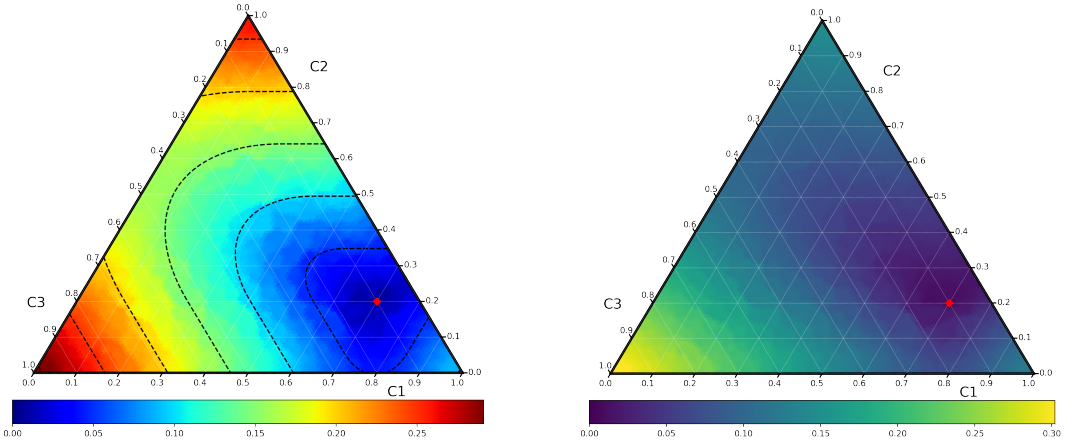
Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.4: optdigits: $||P_0||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.5: optdigits: $||P||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.6: optdigits: $||P_0||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss

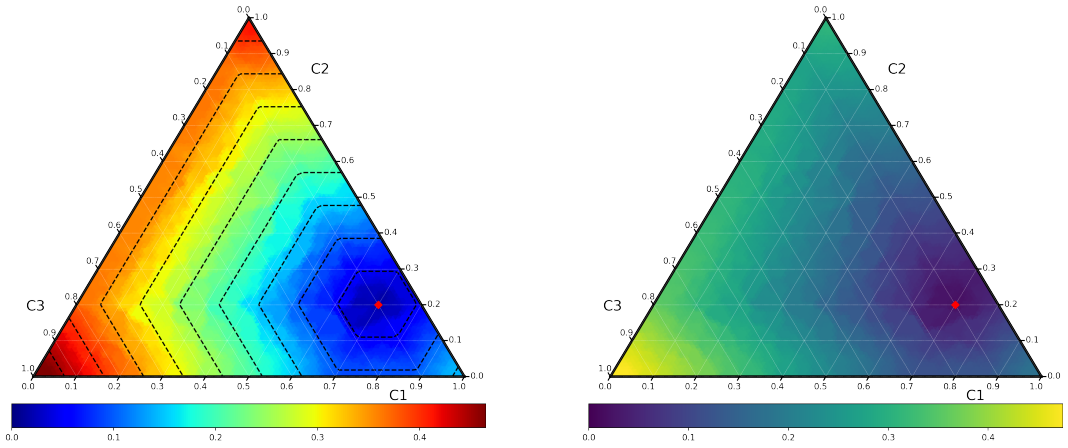


Abbildung A.7: optdigits: $||P||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

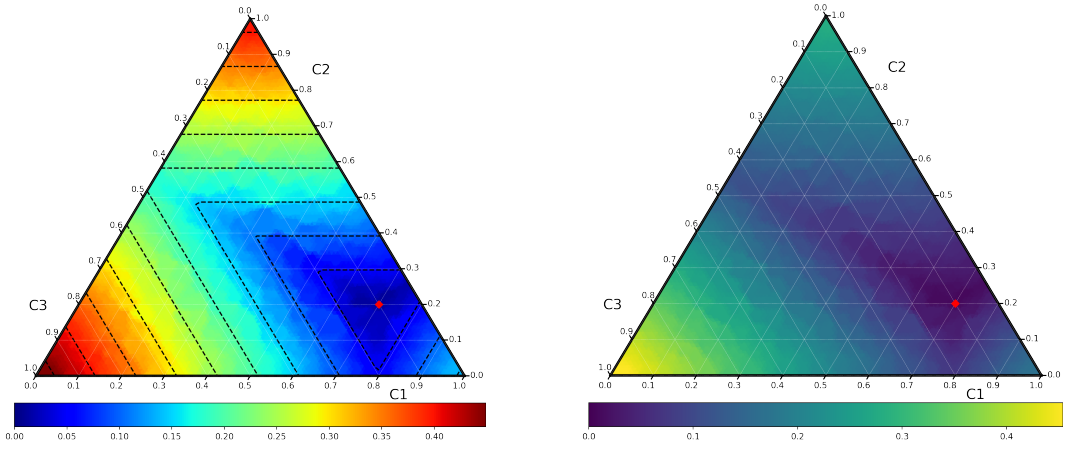


Abbildung A.8: optdigits: $||P_0||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

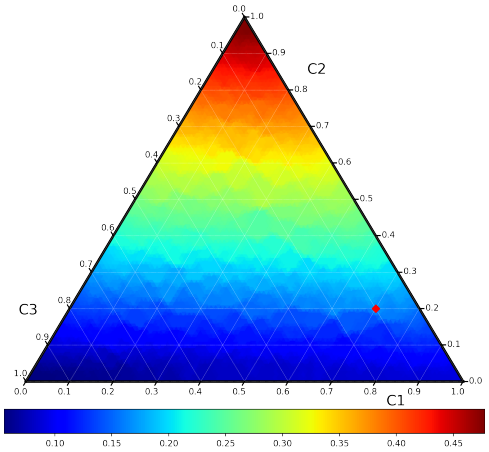
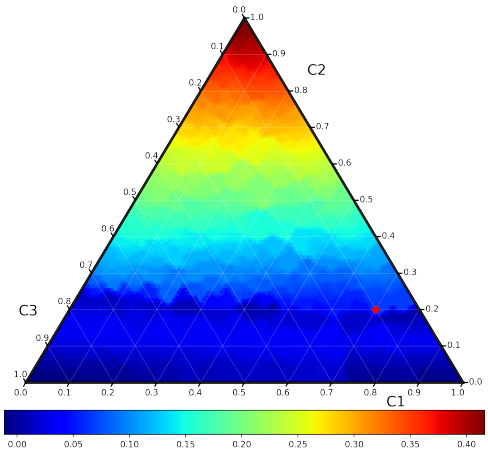
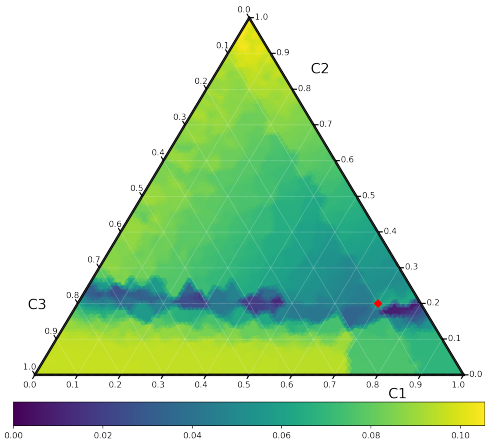


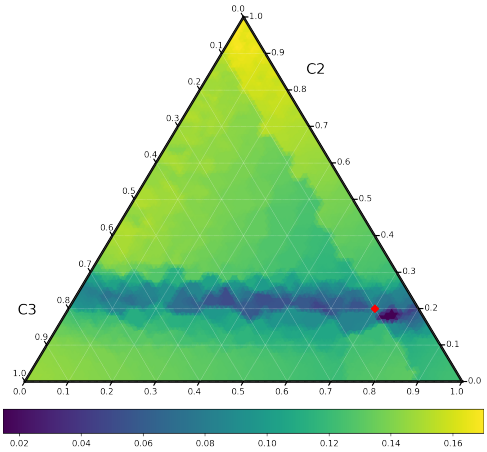
Abbildung A.9: pendigits: $L_{\mathcal{T}} + \epsilon_{\mathcal{T}}$, 01-Loss



Zertifizierter Fehler ϵ

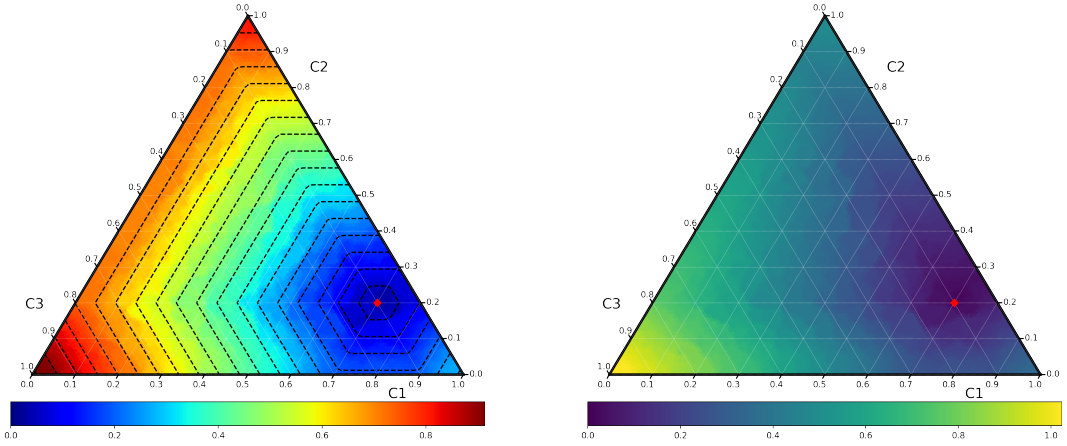


$|\hat{L}_{\mathcal{S}} + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$



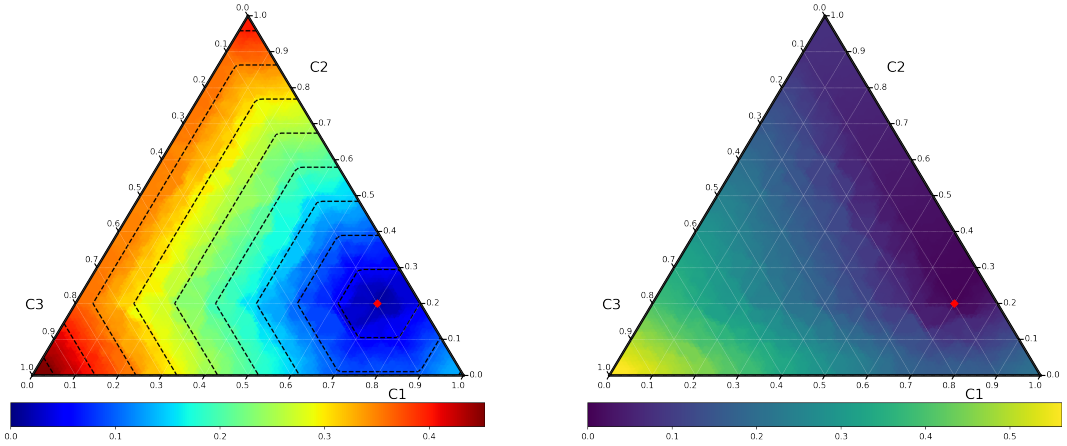
Breite des Konfidenzintervalls

Abbildung A.10: pendigits: Signed Certificate, Logistische Regression, 01-Loss



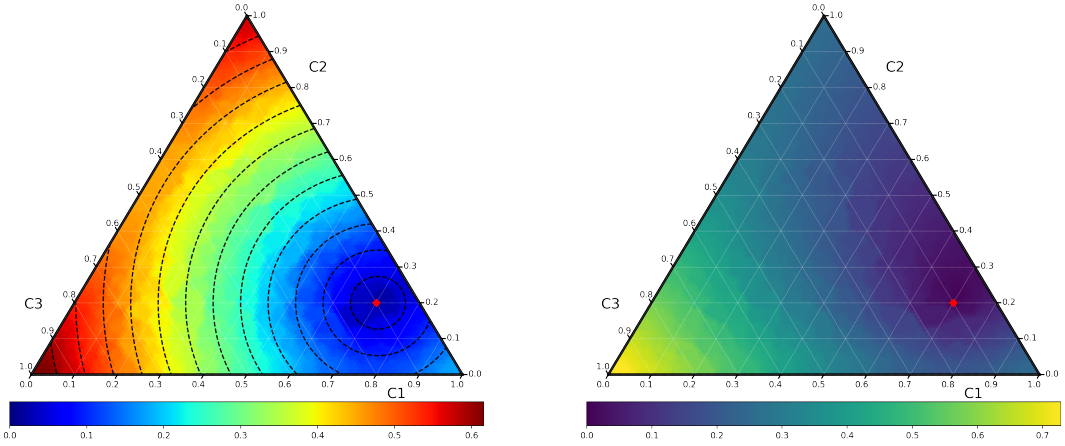
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.11: pendigits: $||P||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



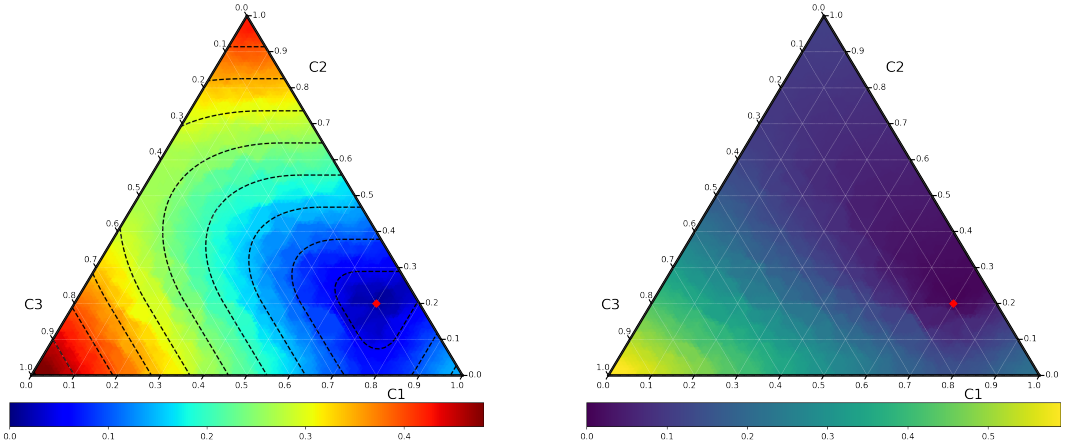
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.12: pendigits: $||P_0||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.13: pendigits: $||P||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.14: pendigits: $||P_0||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss

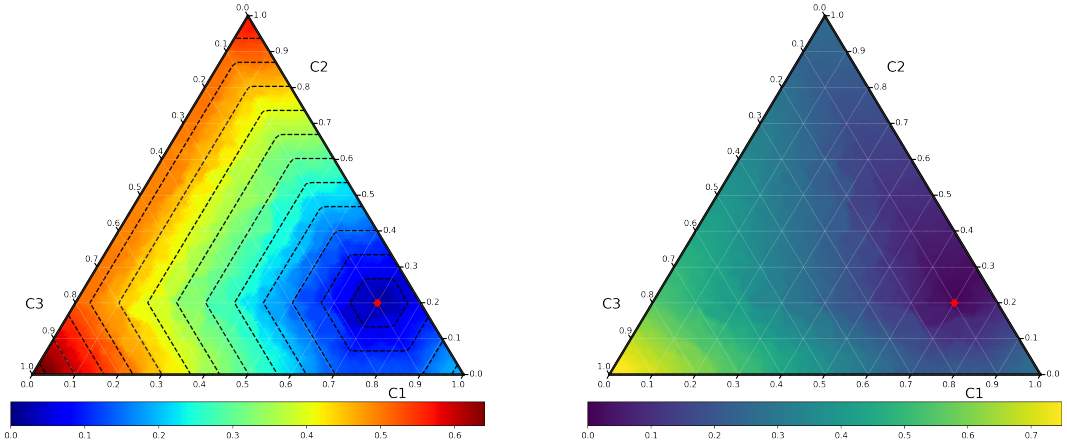


Abbildung A.15: pendigits: $||P||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

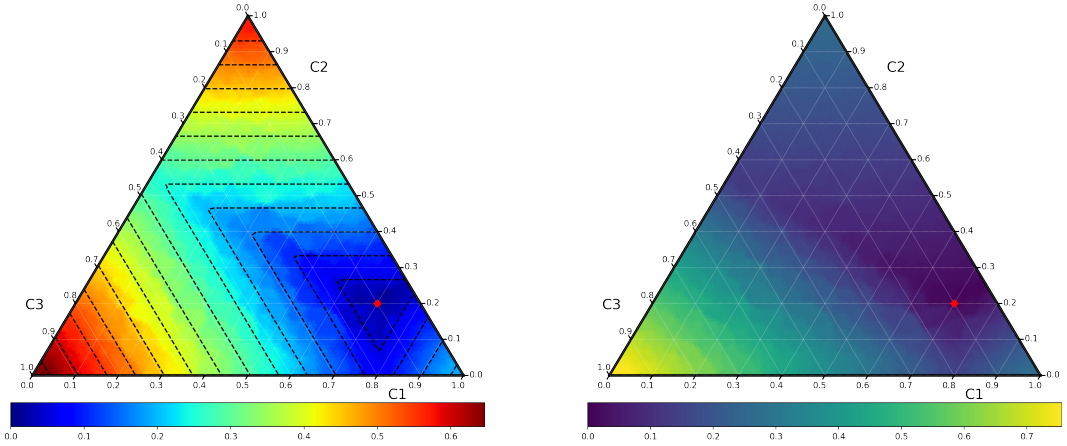


Abbildung A.16: pendigits: $||P_0||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

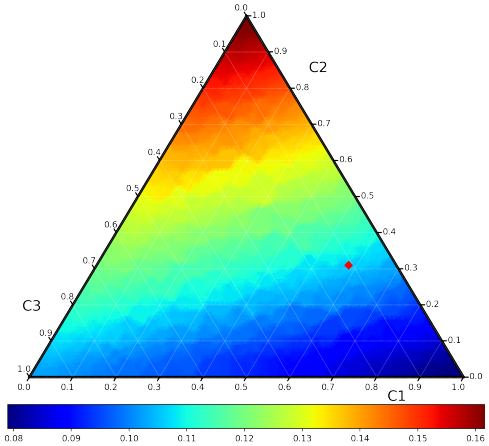
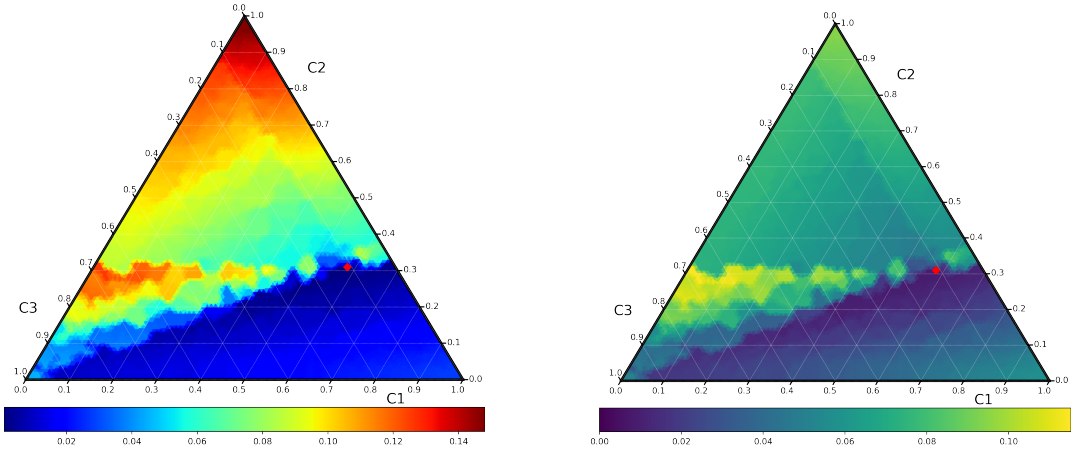
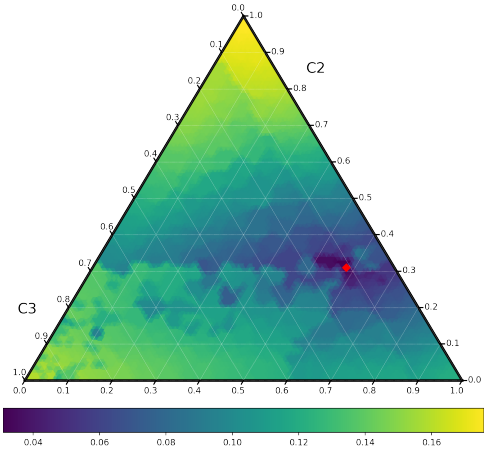


Abbildung A.17: satimage: $L_{\mathcal{T}} + \epsilon_{\mathcal{T}}$, 01-Loss

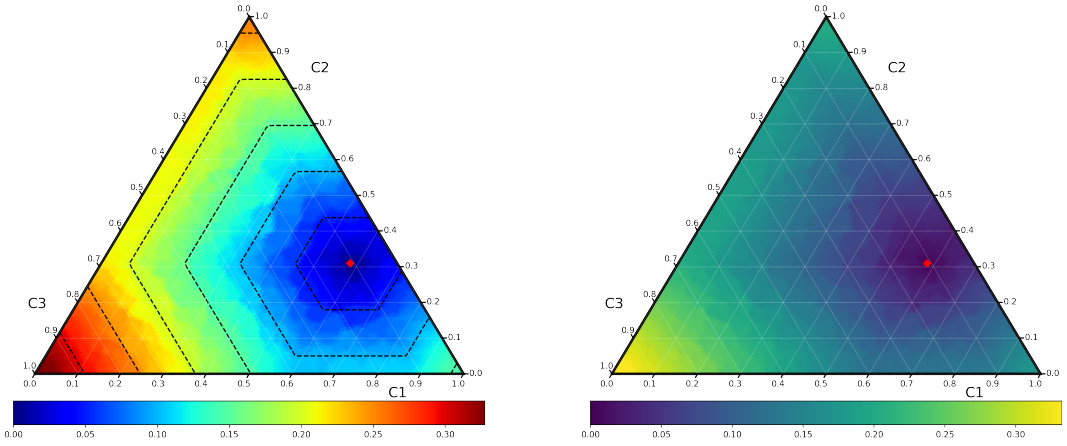


Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$



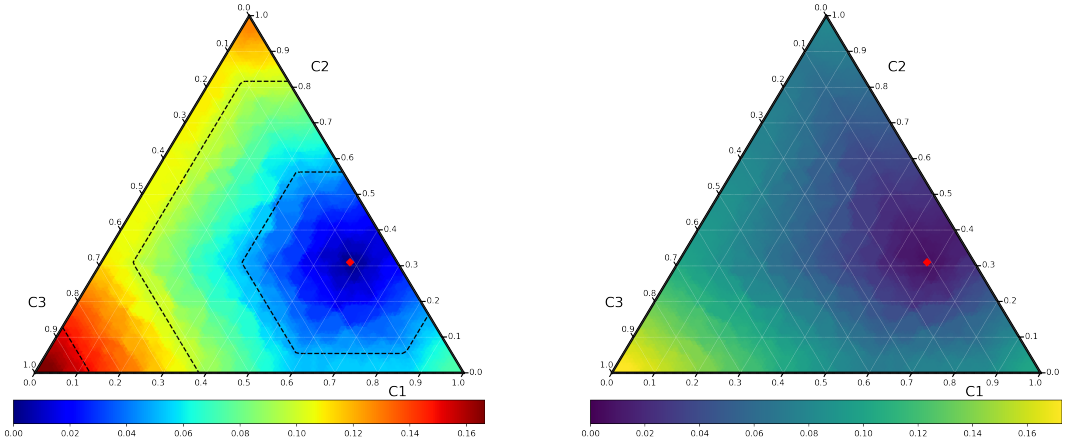
Breite des Konfidenzintervalls

Abbildung A.18: satimage: Signed Certificate, Logistische Regression, 01-Loss



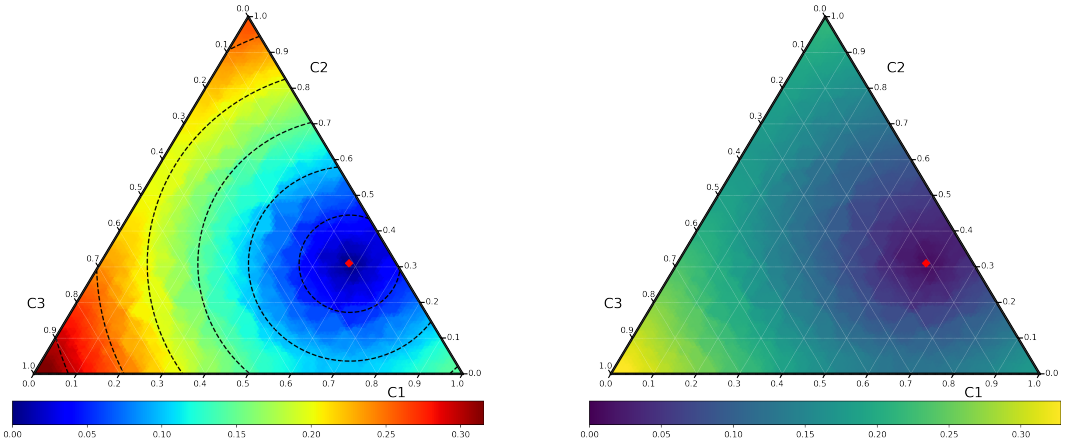
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.19: satimage: $||P||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



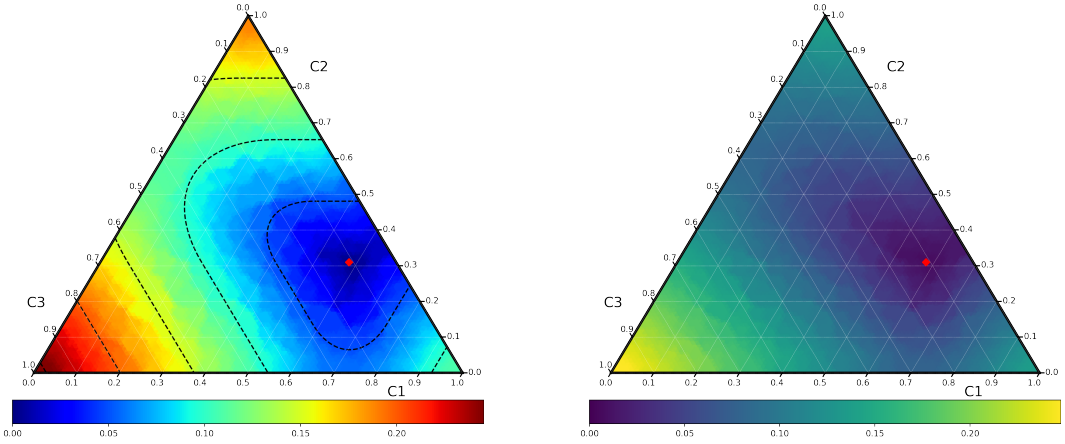
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.20: satimage: $||P_0||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



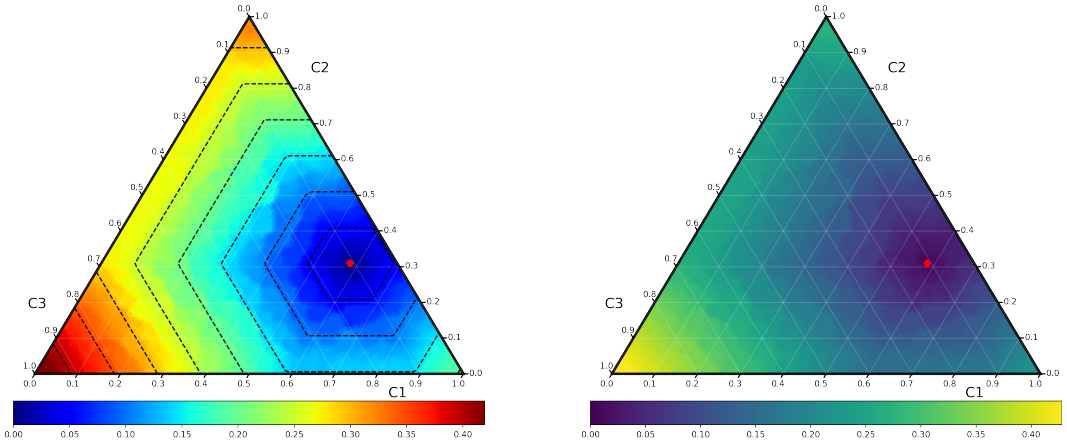
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$

Abbildung A.21: satimage: $||P||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



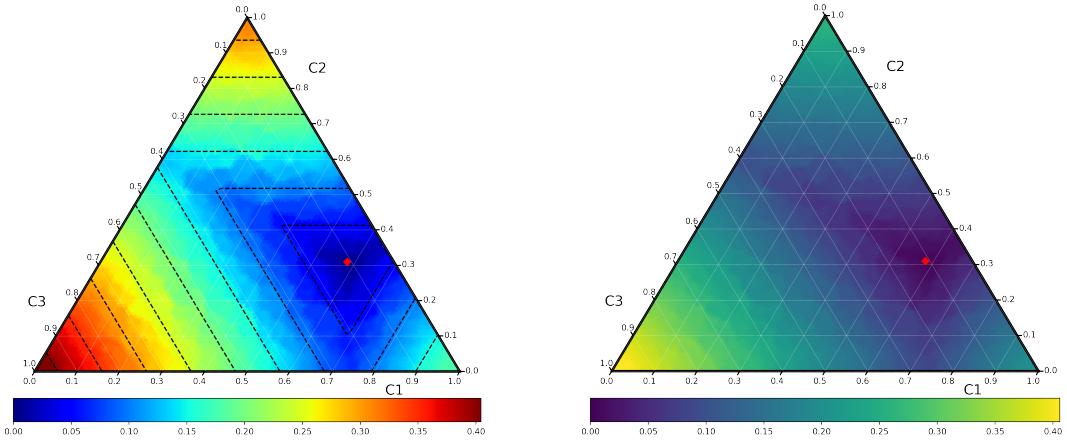
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$

Abbildung A.22: satimage: $||P_0||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.23: satimage: $||P||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.24: satimage: $||P_0||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

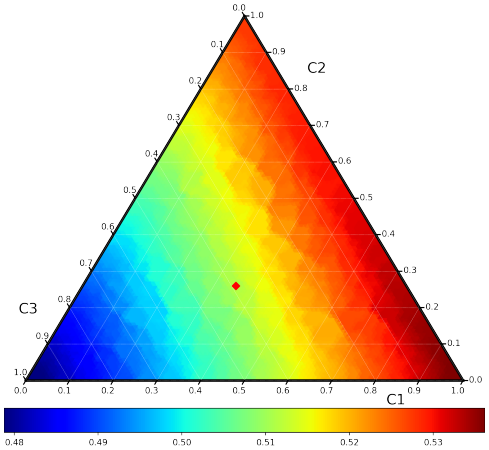
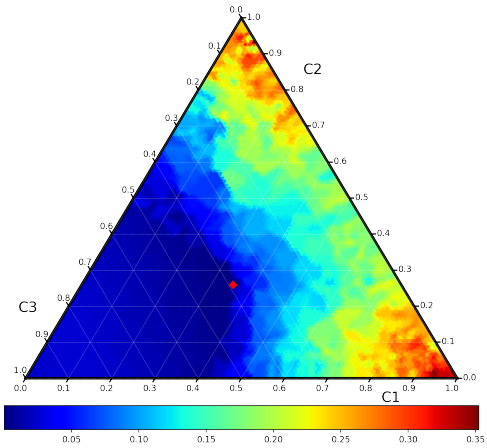
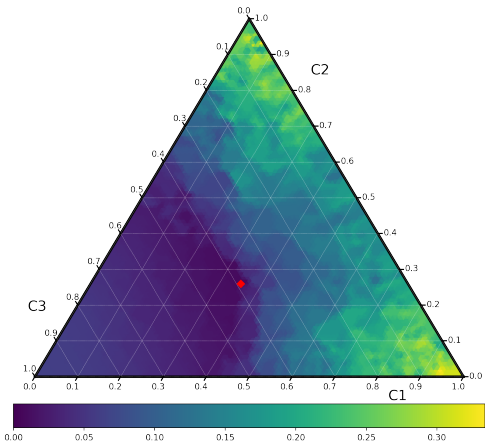


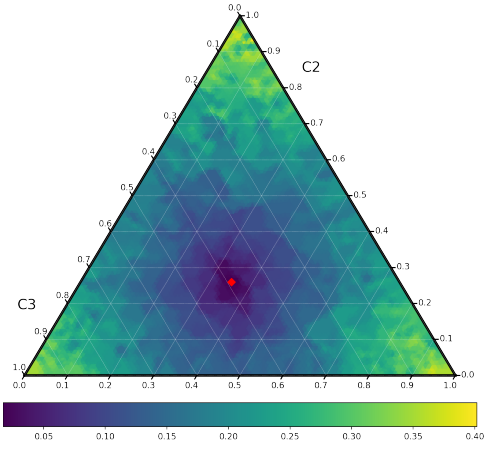
Abbildung A.25: eye movements: $L_{\mathcal{T}} + \epsilon_{\mathcal{T}}$, 01-Loss



Zertifizierter Fehler ϵ



$|\hat{L}_{\mathcal{S}} + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$



Breite des Konfidenzintervalls

Abbildung A.26: eye movements: Signed Certificate, Logistische Regression, 01-Loss

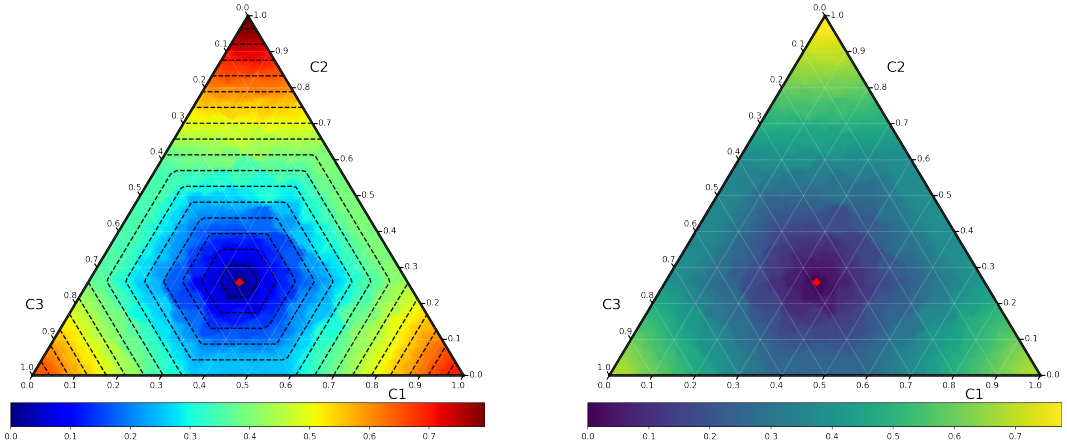


Abbildung A.27: eye movements: $||P||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss

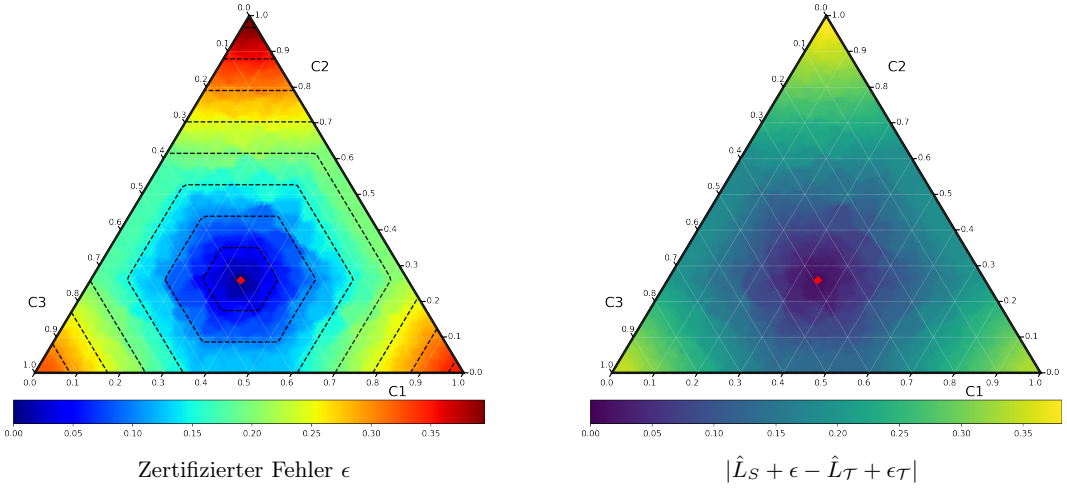
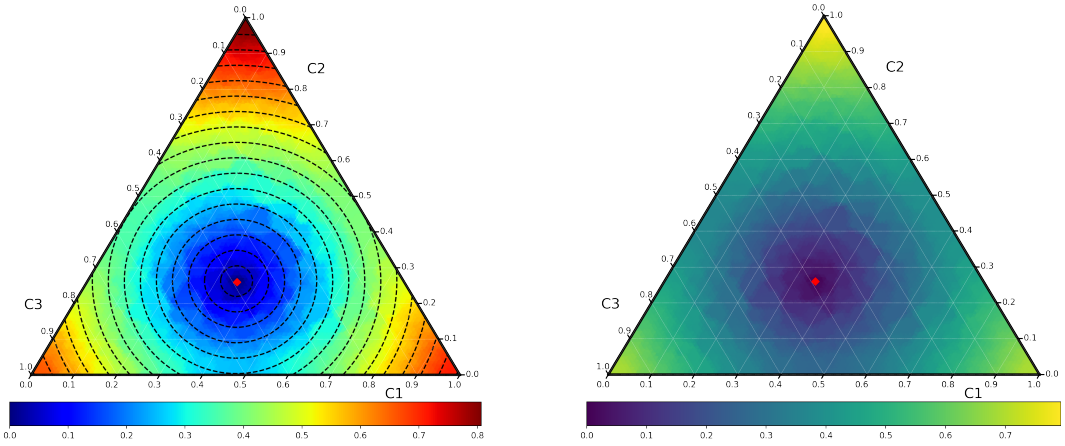
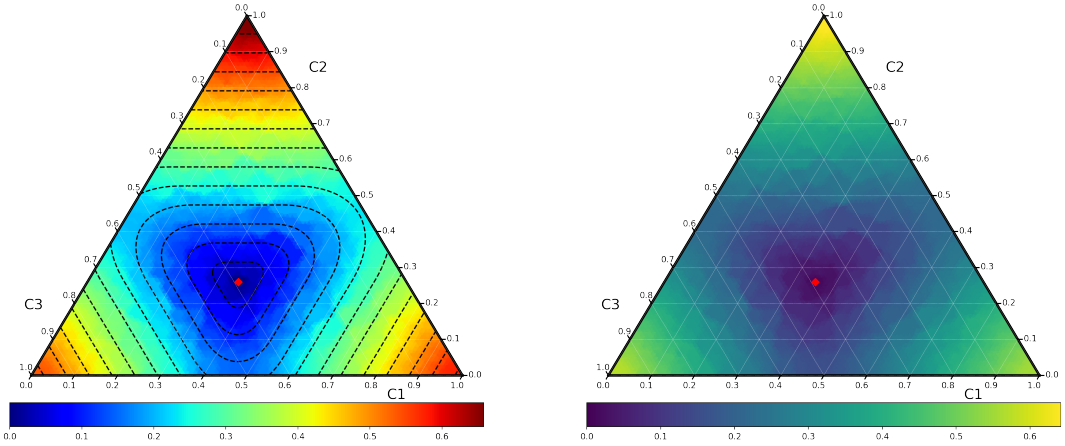


Abbildung A.28: eye movements: $||P_0||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.29: eye movements: $||P||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.30: eye movements: $||P_0||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss

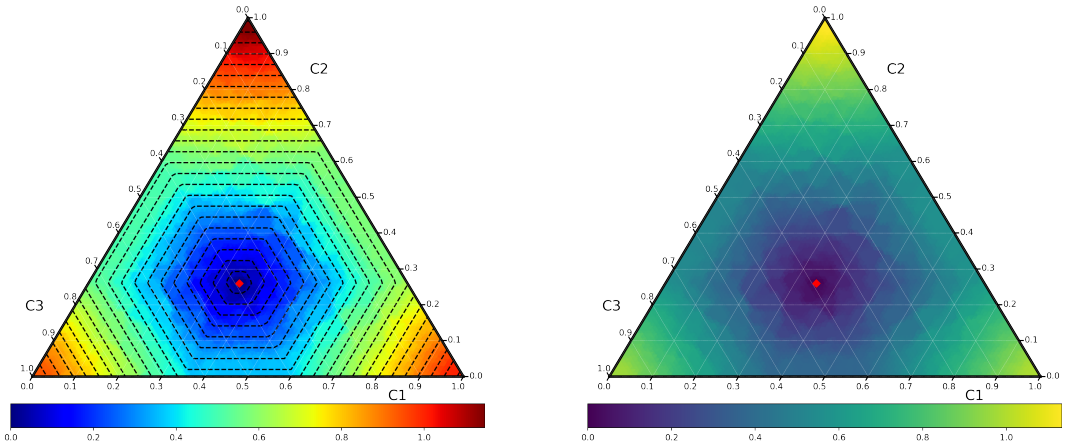


Abbildung A.31: eye movements: $||P||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

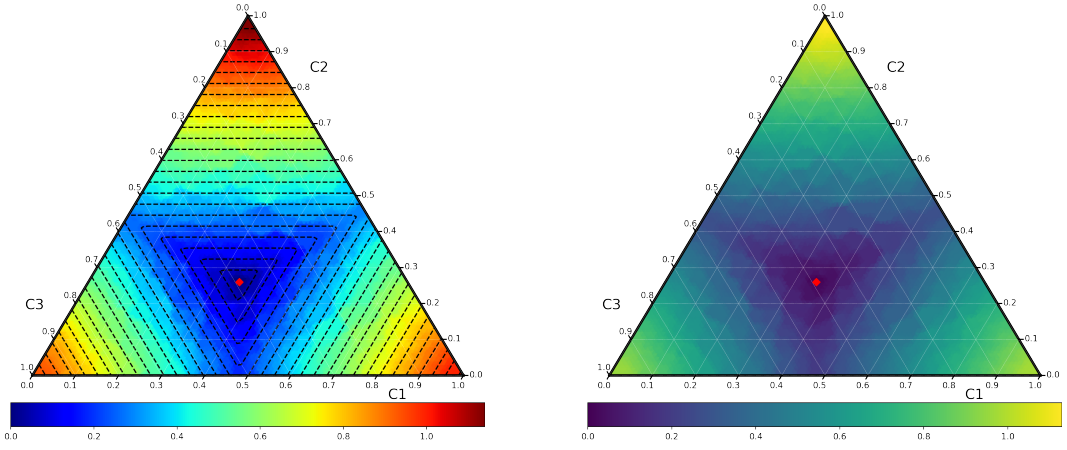


Abbildung A.32: eye movements: $||P_0||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss

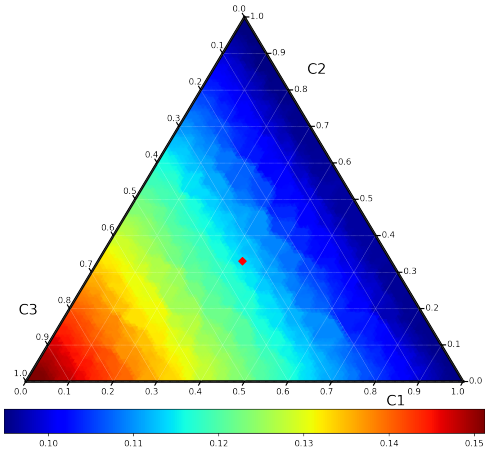
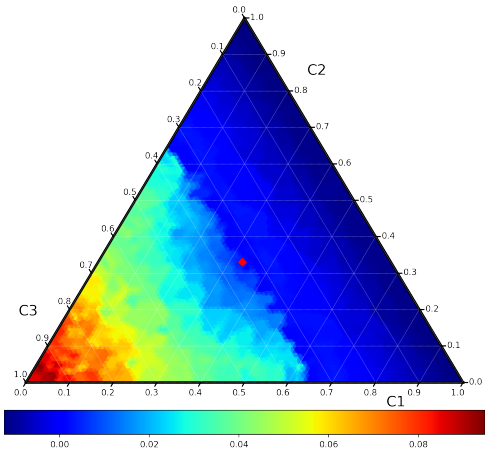
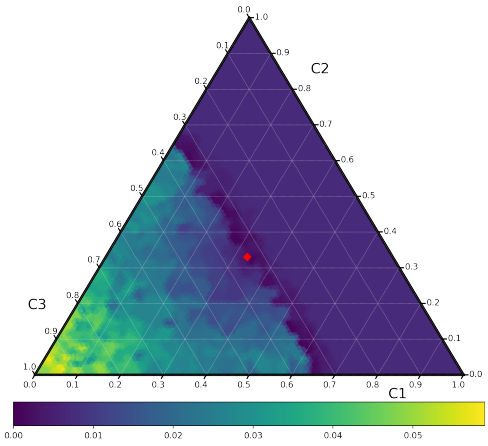


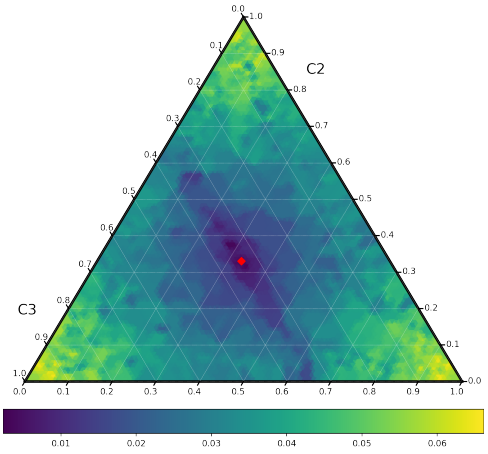
Abbildung A.33: bng: $L_{\mathcal{T}} + \epsilon_{\mathcal{T}}$, 01-Loss



Zertifizierter Fehler ϵ

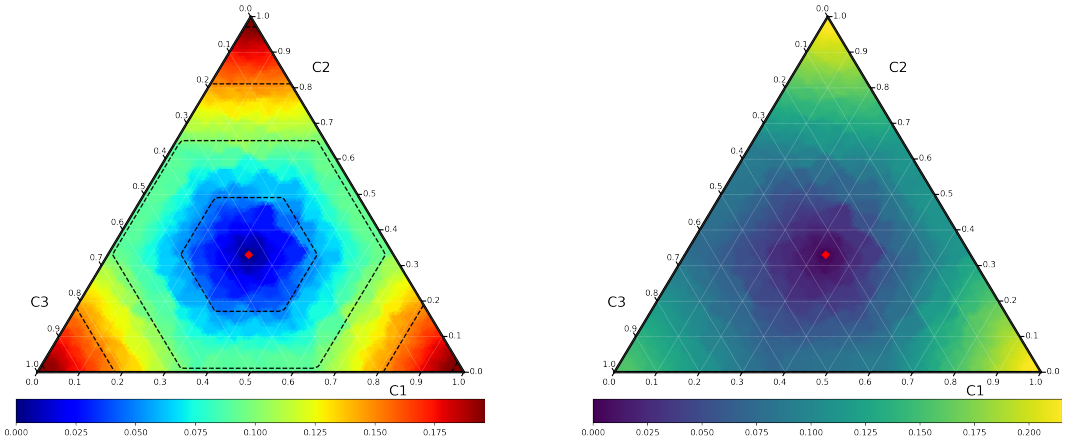


$|\hat{L}_{\mathcal{S}} + \epsilon - \hat{L}_{\mathcal{T}} + \epsilon_{\mathcal{T}}|$



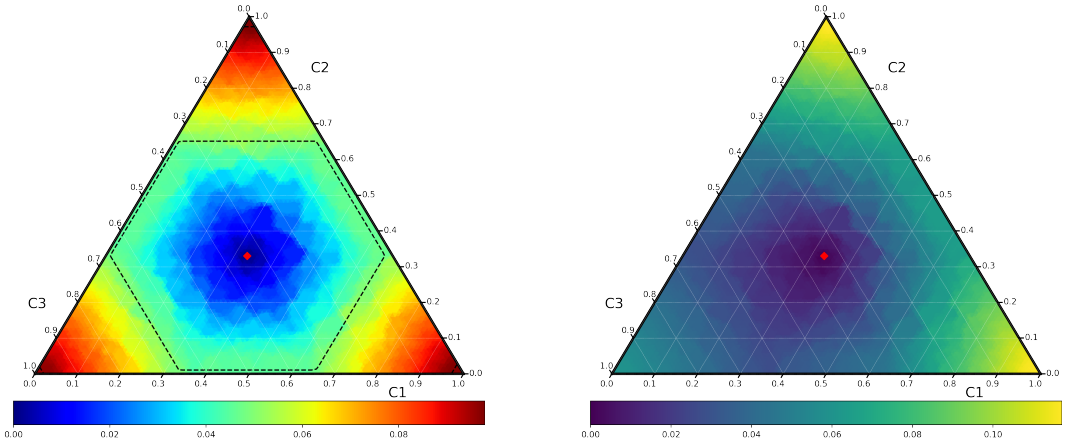
Breite des Konfidenzintervalls

Abbildung A.34: bng: Signed Certificate, Logistische Regression, 01-Loss



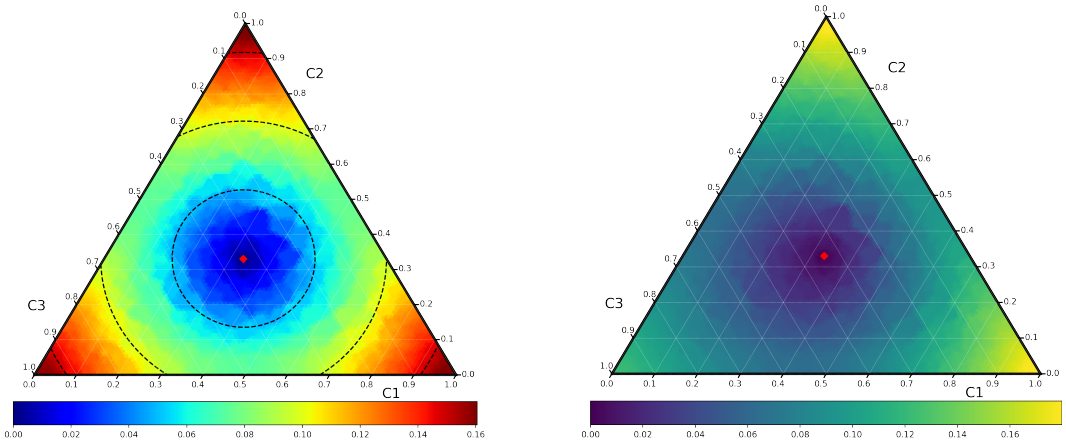
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.35: bng: $||P||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



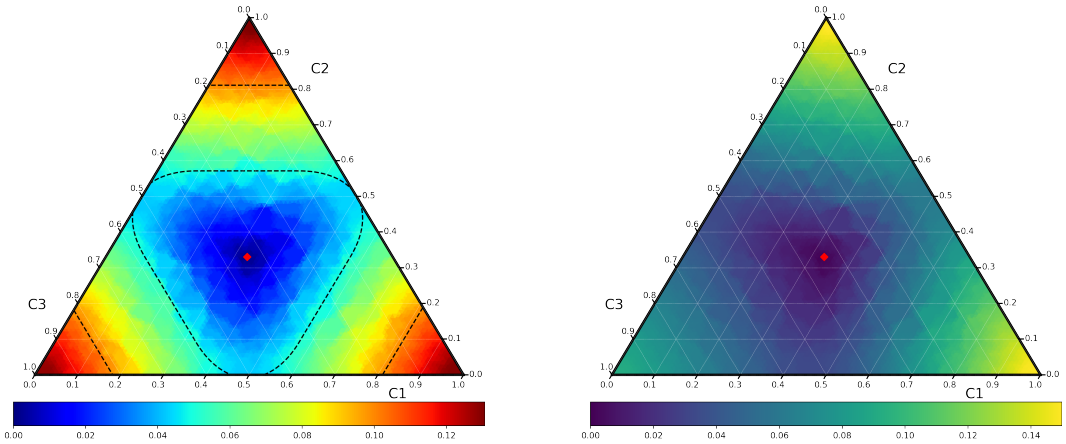
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.36: bng: $||P_0||_1 \cdot ||\ell_h||_\infty$, Logistische Regression, 01-Loss



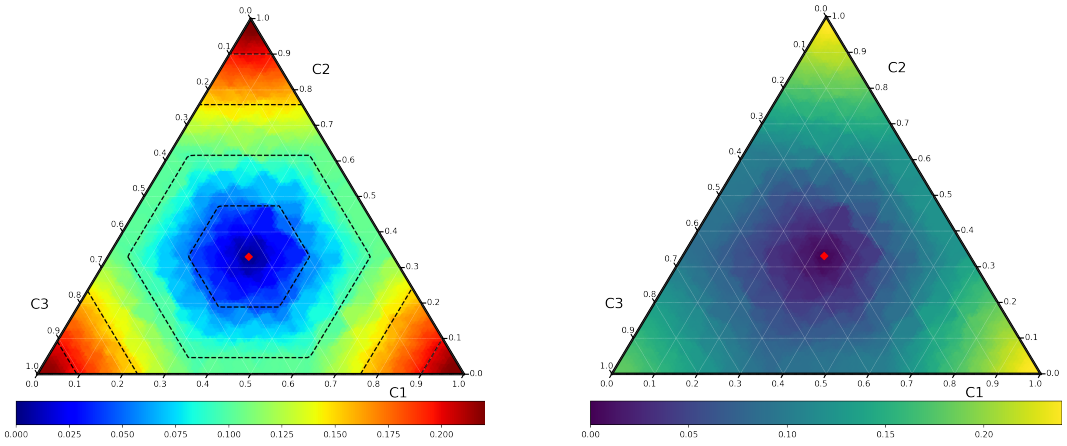
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.37: bng: $||P||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



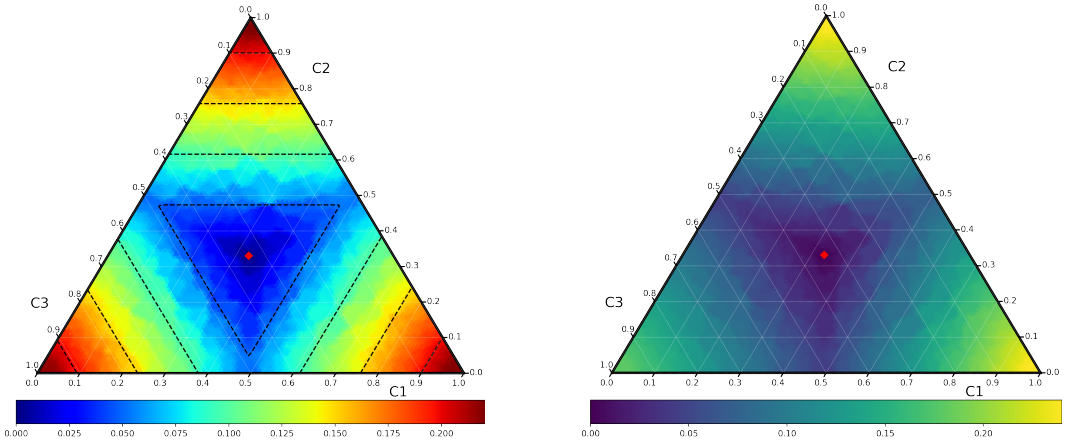
Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.38: bng: $||P_0||_2 \cdot ||\ell_h||_2$, Logistische Regression, 01-Loss



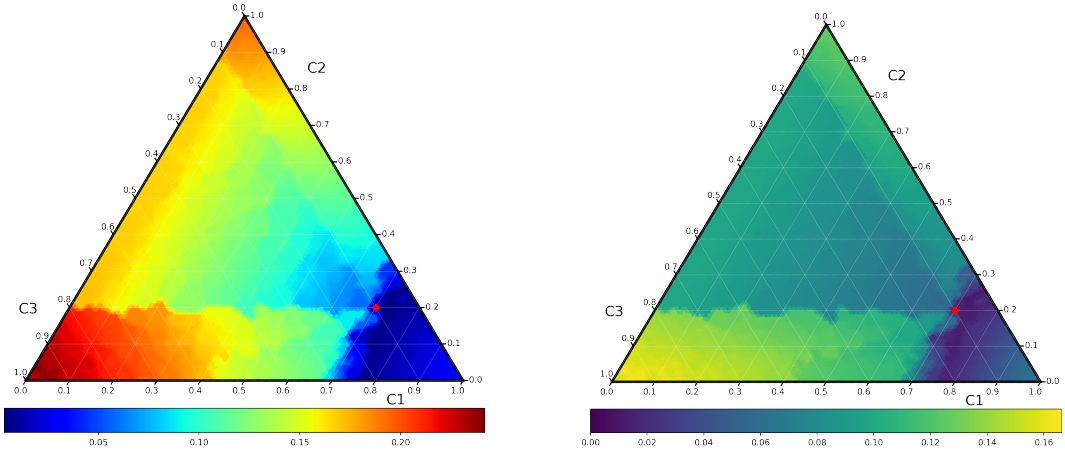
Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.39: bng: $||P||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss



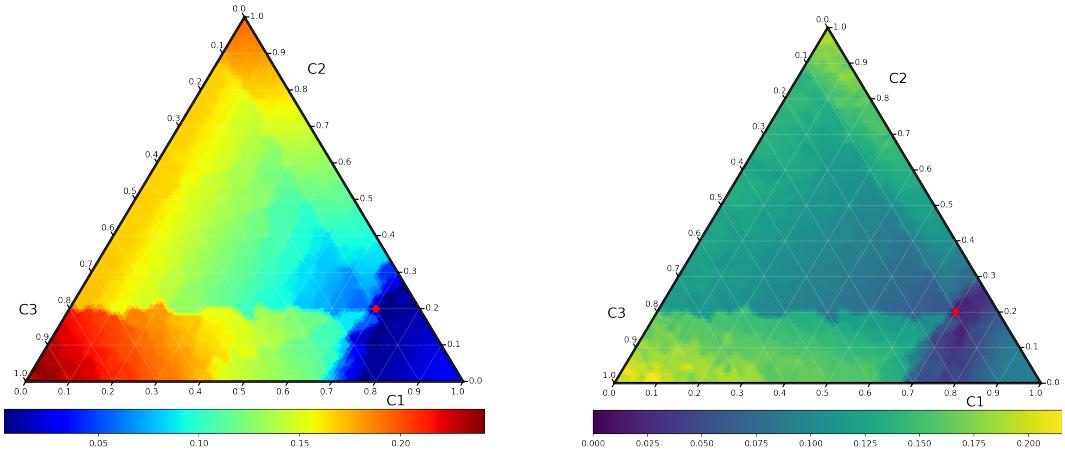
Zertifizierter Fehler ϵ $\quad$ $|\hat{L}_S + \epsilon - \hat{L}_\mathcal{T} + \epsilon_\mathcal{T}|$

Abbildung A.40: bng: $||P_0||_\infty \cdot ||\ell_h||_1$, Logistische Regression, 01-Loss



Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_T + \epsilon_T|$

Abbildung A.41: optdigits: *Signed Certificate*, Entscheidungsbaum, 01-Loss



Zertifizierter Fehler ϵ $|\hat{L}_S + \epsilon - \hat{L}_T + \epsilon_T|$

Abbildung A.42: optdigits: *Signed Certificate*, Entscheidungsbaum, L1-Loss

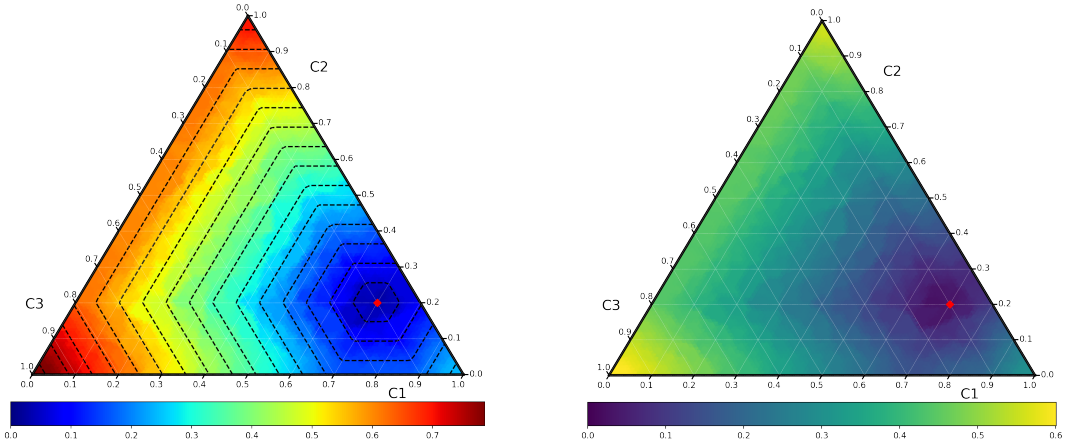


Abbildung A.43: optdigits: $||P||_1 \cdot ||\ell_h||_\infty$, Entscheidungsbaum, HingeLoss

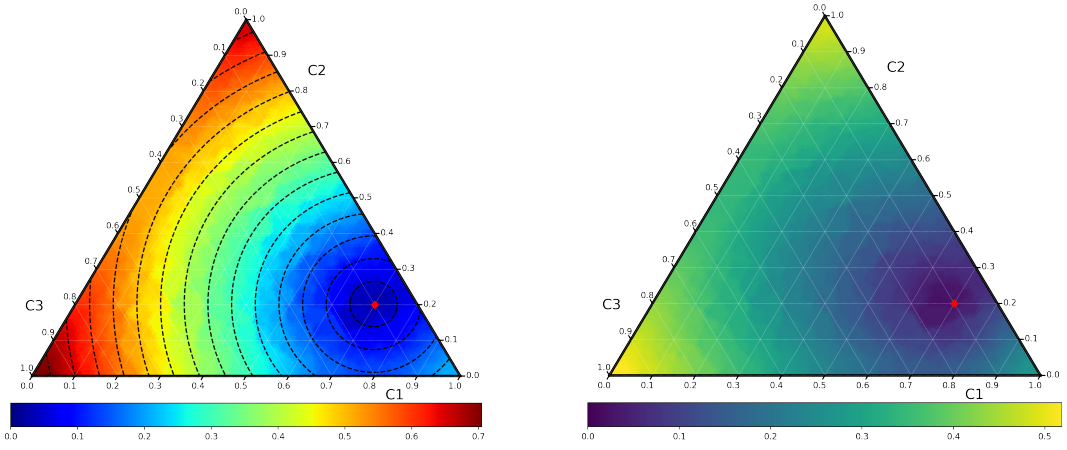


Abbildung A.44: optdigits: $||P||_2 \cdot ||\ell_h||_2$, Entscheidungsbaum, HingeLoss

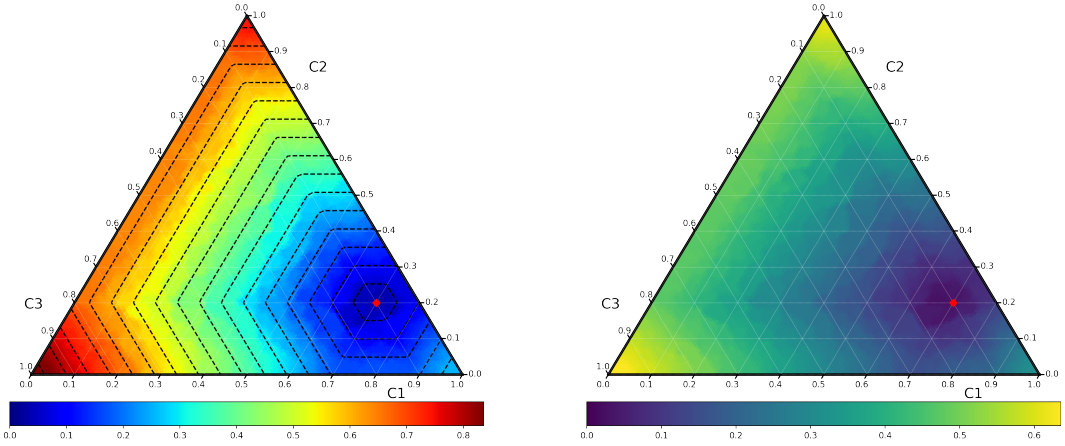


Abbildung A.45: optdigits: $||P||_\infty \cdot ||\ell_h||_1$, Entscheidungsbaum, HingeLoss

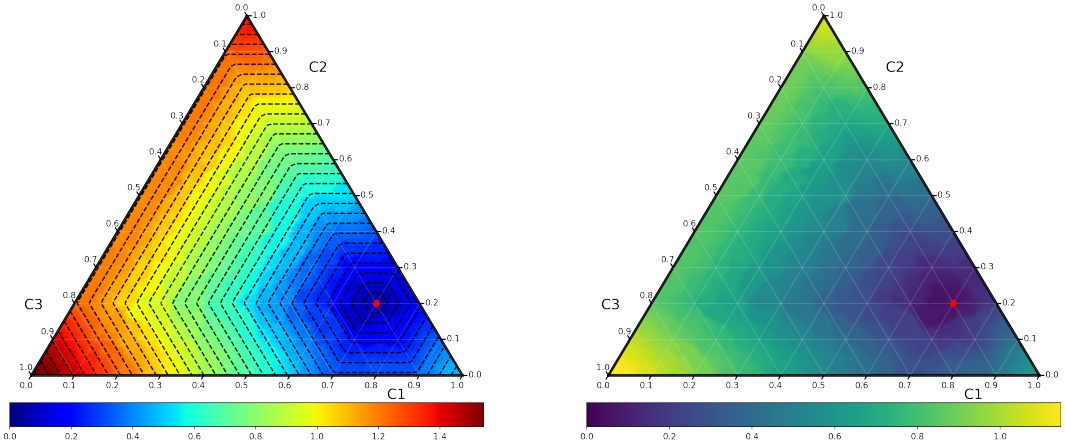


Abbildung A.46: optdigits: $||P||_\infty \cdot ||\ell_h||_1$, Entscheidungsbaum, L2-Loss

Abbildungsverzeichnis

Literaturverzeichnis

[1] Arnold, Matthew, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski et al.: *FactSheets: Increasing trust in AI services through supplier's declarations of conformity.* IBM Journal of Research and Development, 63(4/5):6–1, 2019.

[2] Benavoli, Alessio, Giorgio Corani und Francesca Mangili: *Should we really use post-hoc tests based on mean-ranks?* The Journal of Machine Learning Research, 17(1):152–161, 2016.

[3] Bousquet, Olivier, Stéphane Boucheron und Gábor Lugosi: *Introduction to statistical learning theory.* In: *Summer school on machine learning*, Seiten 169–207. Springer, 2003.

[4] Bunse, Mirko, Christian Bockermann, Jens Buss, Katharina Morik, Wolfgang Rhode und Tim Ruhe: *Smart control of monte carlo simulations for astroparticle physics.* Astronomical Data Analysis Software and Systems XXVII, 2017.

[5] Bunse, Mirko und Katharina Morik: *Active Class Selection with Uncertain Deployment Class Proportions.* In: *Workshop on Interactive Adaptive Learning*, Seite 70, 2021.

[6] Bunse, Mirko und Katharina Morik: *Certification of model robustness in active class selection.* In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Seiten 266–281. Springer, 2021.

[7] Bunse, Mirko, Dorina Weichert, Alexander Kister und Katharina Morik: *Optimal Probabilistic Classification in Active Class Selection.* In: *2020 IEEE International Conference on Data Mining (ICDM)*, Seiten 942–947. IEEE, 2020.

[8] Cakmak, Maya und Andrea L Thomaz: *Designing robot learners that ask good questions.* In: *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Seiten 17–24. IEEE, 2012.

[9] Demšar, Janez: *Statistical comparisons of classifiers over multiple data sets.* The Journal of Machine Learning Research, 7:1–30, 2006.

[10] Farahani, Abolfazl, Sahar Voghoei, Khaled Rasheed und Hamid R Arabnia: *A brief review of domain adaptation.* Advances in data science and information engineering, Seiten 877–894, 2021.

[11] Kottke, Daniel, Georg Krempl, Marianne Stecklina, Cornelius Styp von Rekowski, Tim Sabsch, Tuan Pham Minh, Matthias Deliano, Myra Spiliopoulou und Bernhard Sick: *Probabilistic active learning for active class selection.* arXiv preprint arXiv:2108.03891, 2021.

[12] Lomasky, Rachel, Carla E Brodley, Matthew Aernecke, Sandra Bencic und David Walt: *Guiding class selection for an artificial nose.* In: *NIPS Workshop on Testing of Deployable Learning and Decision Systems*, 2006.

[13] Lomasky, Rachel, Carla E Brodley, Matthew Aernecke, David Walt und Mark Friedl: *Active class selection.* In: *European Conference on Machine Learning*, Seiten 640–647. Springer, 2007.

[14] MacKay, David JC, David JC Mac Kay et al.: *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

[15] Mello, R Fernandes de und M Antonelli Ponti: *Statistical Learning Theory.* Rodrigo Fernandes de Mello, Seite 75, 2018.

[16] Mitchell, Margaret, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji und Timnit Gebru: *Model cards for model reporting.* In: *Proceedings of the conference on fairness, accountability, and transparency*, Seiten 220–229, 2019.

[17] Moreno-Torres, Jose G, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla und Francisco Herrera: *A unifying view on dataset shift in classification.* Pattern recognition, 45(1):521–530, 2012.

[18] Pan, Sinno Jialin und Qiang Yang: *A survey on transfer learning.* IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009.

[19] Parsons, Thomas D und James L Reinebold: *Adaptive virtual environments for neuropsychological assessment in serious games.* IEEE Transactions on Consumer Electronics, 58(2):197–204, 2012.

[20] Raji, Inioluwa Deborah, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel

Theron und Parker Barnes: *Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing.* In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*, Seiten 33–44, 2020.

[21] Shalev-Shwartz, Shai und Shai Ben-David: *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[22] Vapnik, Vladimir: *The nature of statistical learning theory.* Springer science & business media, 1999.

[23] Vapnik, Vladimir N: *An overview of statistical learning theory.* IEEE transactions on neural networks, 10(5):988–999, 1999.

[24] Weiss, Gary M und Foster Provost: *Learning when training data are costly: The effect of class distribution on tree induction.* Journal of artificial intelligence research, 19:315–354, 2003.

[25] Wu, Dongrui, Brent J Lance und Thomas D Parsons: *Collaborative filtering for brain-computer interaction using transfer learning and active class selection.* PloS one, 8(2):e56624, 2013.

[26] Yang, Wei H: *On generalized Hölder inequality.* Nonlinear Analysis: Theory, Methods & Applications, 16(5):489–498, 1991.

[27] Zhang, Kun, Bernhard Schölkopf, Krikamol Muandet und Zhikun Wang: *Domain adaptation under target and conditional shift.* In: *International Conference on Machine Learning*, Seiten 819–827. PMLR, 2013.