

Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit

Evaluation von Zwischenergebnissen in Entscheidungsbäumen

Julius Voggesberger

Studiengang: Softwaretechnik

Prüfer/in: PD Dr. rer. nat. habil. Holger Schwarz

Betreuer/in: Manuel Fritz, M. Sc.

Beginn am: 07. November 2018

Beendet am: 07. Mai 2019

Kurzfassung

Durch Technologien wie das Internet der Dinge und die Industrie 4.0 steigt die Menge an Daten auf der Welt rasant an. Klassifikationsalgorithmen werden von Analysten verwendet, um diese Menge an Daten zu analysieren. Eine Gruppe der populärsten Klassifikationsalgorithmen sind Entscheidungsbaumalgorithmen. Diese stellen erlernte Modelle menschenverständlich als einen Baum dar. Durch die steigende Menge an Daten kann es jedoch vorkommen, dass die erstellten Bäume immer größer, komplexer und unverständlicher für Analysten werden. Häufig werden die Bäume nachdem sie erstellt wurden gestutzt, um sie so kleiner und verständlicher zu machen. Methoden, die die Bäume nach der Erstellung kürzen, werden Post-Pruning-Methoden genannt. Jedoch benötigen Post-Pruning-Methoden eine hohe Laufzeit, da der Entscheidungsbaum erst komplett erstellt werden muss, ehe er gekürzt werden kann. Eine wenig erforschte Alternative sind Methoden, die während der Erstellung des Entscheidungsbaumes diesen kürzen. Diese Methoden werden auch Pre-Pruning-Methoden genannt.

In dieser Arbeit wird ein Verfahren vorgestellt, mit dem Pre-Pruning-Methoden allgemeingültig auf Entscheidungsbäume der Familie der Top-Down Induction of Decision Trees angewendet werden können. Viele Pre-Pruning-Methoden benötigen Schwellenwerte. Aus diesem Grund wurde weiterhin ein Ansatz entwickelt, der es einem Analysten ermöglicht für eine gewünschte Baumqualität einen Schwellenwert für die Pre-Pruning-Methoden zu erhalten. Dieser Ansatz soll es ermöglichen, die Pre-Pruning-Methoden evaluieren zu können. Eine Evaluation mithilfe dieses Ansatzes zeigt, dass hohe Laufzeiteinsparungen mithilfe der Pre-Pruning-Methoden möglich sind. Es konnten jedoch keine zuverlässigen Aussagen hinsichtlich der Qualität der Pre-Pruning-Methoden getroffen werden. Weitere Forschung hinsichtlich der Einflüsse von Datensätzen auf diese Methoden ist nötig, um zuverlässige Aussagen über die Qualität der Methoden treffen zu können.

Inhaltsverzeichnis

1. Einleitung	13
2. Grundlagen	15
2.1. Terminologie	15
2.2. Entscheidungsbäume	16
2.3. Pruning	25
3. Verwandte Arbeiten	31
3.1. Wissenschaftliche Arbeiten	31
3.2. Data-Mining-Werkzeuge	33
3.3. Zusammenfassung	35
4. Konzept	37
4.1. Erstellung der TDIDT-Entscheidungsbäume	37
4.2. Annäherung der Qualität	40
5. Methoden für Pre-Pruning	43
5.1. Split-Kriterium-Schwellenwert	44
5.2. χ^2 -Pruning	45
5.3. J-Pruning	46
5.4. Anzahl an Instanzen pro Knoten	47
5.5. Baumtiefe	47
6. Evaluation	49
6.1. Versuchsaufbau	49
6.2. Evaluation der Qualität	52
6.3. Evaluation der Performanz	64
6.4. Vergleich von Pre-Pruning und Post-Pruning-Methoden	66
7. Zusammenfassung	73
8. Ausblick	77
Literaturverzeichnis	79
A. Anhang	81
A.1. Zusammenhang zwischen Pre-Pruning-Methoden und der Macro F1-Measure	81
A.2. Evaluation der qualitativen Zusammenhänge mit der Macro F1-Measure	84
A.3. Qualität des J-Pruning mit der Macro F1-Measure	84
A.4. Vorhersage der Macro F1-Measure	85

A.5.	Vergleich von Pre-Pruning und Post-Pruning-Methoden anhand der Macro F1-Measure	87
A.6.	Laufzeit der Pruning-Methoden	88

Abbildungsverzeichnis

2.1.	Ein Entscheidungsbaum, der mit dem ID3-Algorithmus erzeugt wurde [Qui86]. . .	16
2.2.	Ein C4.5 Entscheidungsbaum, der auf dem labor-neg Datensatz aus dem UCI Machine Learning Repository bezogen wurde [Qui93]	21
2.3.	Ein aus Abbildung 2.2 abgewandelter Entscheidungsbaum, der den Regeln eines CART-Baumes entspricht	23
2.4.	Ein Entscheidungsbaum vor und nach dem kürzen durch Reduced Error Pruning .	27
2.5.	Ein Entscheidungsbaum vor und nach dem kürzen durch Error-Based Pruning . .	28
3.1.	Einstellungen von Data-Mining-Werkzeugen zu Pruning	34
4.1.	Der allgemeine Aufbau eines TDIDT-Entscheidungsbaumalgorithmus	38
4.2.	Eine Regressionskurve, die den Zusammenhang zwischen clustering-spezifischen Werten und der erzielten Qualität darstellt [FBS19].	41
5.1.	Anwendung der Pre-Pruning-Methoden an den Ablauf eines TDIDT-Algorithmus	43
5.2.	Ein Entscheidungsbaum, der mit dem ID3-Algorithmus aus dem Lenses-Datensatz erzeugt wurde.	44
6.1.	Scatterplots, die den Wert der Pre-Pruning-Methoden der Accuracy eines ID3-Entscheidungsbaumes gegenüberstellen.	55
6.2.	Scatterplots, die den Wert der Pre-Pruning-Methoden der Accuracy eines C4.5-Entscheidungsbaumes gegenüberstellen.	56
6.3.	Scatterplots, die den Wert der Pre-Pruning-Methoden der Accuracy eines CART-Entscheidungsbaumes gegenüberstellen.	58
6.4.	Vorhersagen der Accuracy, die mithilfe der Regressionen aus Kapitel 6.2.2 getroffen wurden.	62
6.5.	Einsparungen der Laufzeit durch Pre-Pruning-Methoden.	65
6.6.	Die Gesamtlaufzeiten der Entscheidungsbäume und der Pruning-Methoden. . . .	68
6.7.	Die Laufzeit des J-Pruning, EBP und REP.	70
A.1.	Scatterplots, die den Wert der Pre-Pruning-Methoden der Genauigkeit eines ID3-Entscheidungsbaumes gegenüberstellen.	81
A.2.	Scatterplots, die den Wert der Pre-Pruning-Methoden der Genauigkeit eines C4.5-Entscheidungsbaumes gegenüberstellen.	82
A.3.	Scatterplots, die den Wert der Pre-Pruning-Methoden der Genauigkeit eines ID3-Entscheidungsbaumes gegenüberstellen.	83
A.4.	Vorhersagen der Macro F1-Measure, die mithilfe der Regressionen aus Kapitel 6.2.2 getroffen wurden, für C4.5 und ID3	85
A.5.	Vorhersagen der Macro F1-Measure, die mithilfe der Regressionen aus Kapitel 6.2.2 getroffen wurden, für den CART-Algorithmus	86

A.6. Die Gesamtlaufzeiten der Entscheidungsbäume und der Pruning-Methoden. . . . 87

Tabellenverzeichnis

2.1.	Trainingsdatensatz des Baummodells aus Abbildung 2.1 [Qui86]	17
2.2.	Vergleich der Algorithmen ID3, C4.5 und CART	24
6.1.	Datensätze, die für den ID3-Algorithmus verwendet wurden.	51
6.2.	Datensätze, die für den C4.5- und CART-Algorithmus verwendet wurden.	51
6.3.	Die Regressionsmetriken der Regressionskurve. Die Werte sind auf drei Stellen nach dem Komma gerundet.	59
6.4.	Eine Tabelle mit Messwerten der Baummodelle ohne Pruning, die für das Training der Regressionskurve verwendet wurden.	59
6.5.	Eine Tabelle mit Messwerten der Baummodelle ohne Pruning, die für das Testen der Regressionskurve verwendet wurden.	60
6.6.	Accuracy der Entscheidungsbäume mit und ohne J-Pruning	64
A.1.	Die Regressionsmetriken der Regressionskurve der Macro F1-Measure und Pre-Pruning-Methoden. Die Werte sind auf drei Stellen nach dem Komma gerundet.	84
A.2.	Macro F1-Measure der ID3-Entscheidungsbäume mit und ohne J-Pruning	84
A.3.	Macro F1-Measure der C4.5-Entscheidungsbäume mit und ohne J-Pruning	84
A.4.	Macro F1-Measure der CART-Entscheidungsbäume mit und ohne J-Pruning	84

Akronyme

CART Classification and Regression Tree. 18

EBP Error-Based Pruning. 28

REP Reduced Error Pruning. 26

TDIDT Top-Down Induction of Decision Trees. 16

WEKA Waikato Environment for Knowledge Analysis. 33

1. Einleitung

Die Menge an Daten, die in der heutigen Zeit generiert wird, steigt rasant an. Neue Technologien wie das Internet der Dinge und die Industrie 4.0 erlauben es immer mehr Daten zu generieren. So soll von 130 Exabytes im Jahr 2005 die weltweite Menge an Daten auf bis zu 40000 Exabytes im Jahr 2020 steigen [GR12]. Im Jahr 2012 wurden bereits 23% dieser Daten als nützlich für Big Data eingestuft [GR12]. Jedoch wird nur ein geringer Teil der generierten Daten in der heutigen Zeit betrachtet und analysiert. Dieser Anteil lag im Jahr 2012 bei lediglich 0.5% [GR12]. Damit der Mensch die Daten analysieren und Wert daraus schöpfen kann, benötigt er Hilfe in Form von Algorithmen. Algorithmen, die Menschen dabei helfen diese Menge an Daten zu betrachten, sind im Bereich des Data Mining zu finden. Die populärsten Algorithmen des Data-Mining gehören dem Bereich der Klassifikation an [WKR+08]. Klassifikationsalgorithmen erlauben es Daten in Klassen einzuteilen. Insbesondere Entscheidungsbaumalgorithmen sind eine populäre Gruppe an Klassifikationsalgorithmen [WKR+08]. Diese unterscheiden sich von anderen Klassifikationsalgorithmen aufgrund ihrer, für menschliche Analysten verständliche, Darstellung als Baum. Entscheidungsbaumalgorithmen finden in der Medizin zur Prognostizierung und Diagnose von Krankheiten, sowie in der Spieltheorie zur Voraussage von Spielepartien ihre Anwendung [Qui86].

Für die Entscheidungsbaumalgorithmen existieren zwei große Probleme: Zum einen das Erstellen von Entscheidungsbäumen, die zu groß sind um verständlich für menschliche Analysten zu bleiben. Zum anderen das Problem der Bäume zu spezifisch auf den Trainingsdatensatz, auf dem sie erstellt wurden, angepasst zu sein. Das letztere Problem ist auch als Overfitting bekannt. Beide Probleme lassen sich durch Pruning behandeln. Pruning bezeichnet hierbei das Kürzen von Entscheidungsbäumen, um so kleinere Bäume zu erschaffen, in denen kein Overfitting vorkommt. Häufig werden Post-Pruning-Methoden angewendet um Pruning durchzuführen. Diese setzen nach der Erstellung des Entscheidungsbaumes an und benötigen einen ausgewachsenen Entscheidungsbaum. Hieran ist das Problem der Post-Pruning-Methoden zu erkennen. Sie benötigen eine lange Gesamtlaufzeit, da zuerst der Entscheidungsbaum erstellt werden muss und erst anschließend die Post-Pruning-Methoden diesen kürzen.

In dieser Arbeit werden Pre-Pruning-Methoden untersucht, um Entscheidungsbäume zu kürzen. Diese Pruning-Methoden setzen während der Erstellung des Entscheidungsbaumes an, um das Wachstum des Baumes auf dessen Knoten zu stoppen. Von diesen Methoden wird eine hohe Laufzeiteinsparung erhofft, da der Baum nicht auswachsen muss, sondern während der Erstellung das Wachstum des Baumes beschränkt wird. Die Schwierigkeit der Pre-Pruning-Methoden ist jedoch geeignete Knoten zum Stoppen des Wachstums zu finden. Für einige Pre-Pruning-Methoden besteht die Schwierigkeit insbesondere darin Schwellenwerte zu finden, mit denen diese Knoten zuverlässig ausgewählt werden können. Das Ziel dieser Arbeit ist es ein Verfahren zu entwickeln, mit dem Pre-Pruning-Methoden auf allen Entscheidungsbäumen der Familie der Top-Down Induction of Decision Trees angewendet werden können, da diese Familie die populärsten Entscheidungsbaumalgorithmen enthält. Mithilfe dieses Verfahrens sollen geeignete Schwellenwerte für die Pre-Pruning-Methoden

gefunden werden. Weiterhin soll das Verfahren es einem Analysten ermöglichen, für eine gewünschte Qualität für einen Entscheidungsbaum, einen Schwellenwert für eine Pre-Pruning-Methode zu erhalten.

Aufbau der Arbeit

Diese Arbeit ist wie folgt aufgebaut:

Kapitel 2 - Grundlagen: In den Grundlagen werden die in dieser Arbeit verwendeten Entscheidungsbaumalgorithmen und Post-Pruning-Methoden vorgestellt.

Kapitel 3 - Verwandte Arbeiten: Dieses Kapitel behandelt sowohl Literatur als auch Software, die sich mit dem Problem des Pruning beschäftigt hat.

Kapitel 4 - Konzept: In diesem Kapitel wird das in dieser Arbeit entwickelte Verfahren vorgestellt. Dieses erlaubt es, die in dieser Arbeit verwendeten Pre-Pruning-Methoden, auf allen Entscheidungsbäumen der Familie der Top-Down Induction of Decision Trees anzuwenden.

Kapitel 5 - Pre-Pruning-Methoden: In diesem Kapitel werden die fünf Pre-Pruning-Methoden, die in dieser Arbeit verwendet werden, vorgestellt. Weiterhin wird gezeigt, wie diese Methoden auf das Konzept aus Kapitel 4 übertragbar sind.

Kapitel 6 - Evaluation: Dieses Kapitel behandelt den Aufbau und die Durchführung des vorgestellten Verfahrens, sowie dessen Evaluation hinsichtlich Laufzeit und Qualität. Weiterhin werden die Pre-Pruning-Methoden in diesem Kapitel mit den Post-Pruning-Methoden verglichen.

Kapitel 7 - Zusammenfassung: In diesem Kapitel wird die Arbeit und die Ergebnisse der Evaluation zusammengefasst.

Kapitel 8 - Ausblick: In diesem Kapitel wird ein Ausblick auf weitere mögliche Ansätze, die zu diesem Thema verfolgt werden können, vorgestellt.

2. Grundlagen

In diesem Kapitel werden grundlegende Konzepte und Begriffe erklärt, die für das weitere Verständnis der Arbeit notwendig sind. Kapitel 2.1 legt die grundlegende Terminologie der Arbeit fest. In Kapitel 2.2 werden Entscheidungsbäume, sowie ausgewählte Entscheidungsbaumalgorithmen betrachtet und in Kapitel 2.3 wird das Pruning von Entscheidungsbäumen beschrieben.

2.1. Terminologie

Die Datensätze, die in dieser Arbeit verwendet werden, sind alle nach dem selben Muster aufgebaut, um von Entscheidungsbaumalgorithmen verwendet werden zu können. Ein Datensatz besteht aus vielen Datenpunkten, die in dieser Arbeit als Instanzen bezeichnet werden. Diese Instanzen werden durch die Attribute des Datensatzes beschrieben. Ein Attribut A enthält verschiedene Werte a_1, a_2, \dots, a_n mit $n = \text{Anzahl der Werte in } A$. Diese Werte können entweder kategorisch oder kontinuierlich vorliegen und das Attribut wird dementsprechend als kontinuierliches oder kategorisches Attribut bezeichnet. Ein Spezialfall eines Attributes ist das Klassenattribut C . Dieses kommt nur einmal in einem Datensatz vor und hat dabei eine Ausprägung aus mehreren Klassen c_1, c_2, \dots, c_k . Eine Instanz ist die Kombination von Werten verschiedener Attribute und einer Klasse. Anhand von Tabelle 2.1 kann dies beispielhaft betrachtet werden. In dieser Tabelle ist ein Datensatz zu sehen, bei dem die Zeilen Instanzen darstellen und Spalten die Attribute *Bewölkung*, *Temperatur*, *Luftfeuchtigkeit*, *Windig* und das Klassenattribut *Klasse* sind. Alle Attribute besitzen kategorische Werte. So sind die Werte des Attributes *Bewölkung*: *sonnig*, *bewölkt* und *Regen*. Das Klassenattribut *Klasse* besteht aus zwei Klassen N und P .

Ein Entscheidungsbaum besteht aus verschiedenen Klassifikationsregeln, die den Pfaden des Baumes entsprechen. Diese Pfade bestehen aus Knoten und unidirektionalen Kanten, wobei die Knoten in Wurzelknoten, innere Knoten und Blattknoten unterschieden werden. Ein Knoten t wird durch einen Namen repräsentiert, der einem Attribut A entspricht, auf dem der Datensatz an dieser Stelle geteilt wurde. Die Ausnahme davon sind Blattknoten, die mit einer Klasse gekennzeichnet werden. Ein Knoten t und alle seine Nachfolgeknoten, werden als Teilbaum T_t bezeichnet. Der Entscheidungsbaum aus Abbildung 2.1 wird im Folgenden als Beispiel für einen Entscheidungsbaum betrachtet. Der Wurzelknoten aus 2.1 ist durch den Namen *Bewölkung* gekennzeichnet und ist in Blau markiert zu sehen. Blattknoten sind durch gepunktete Linien gekennzeichnet, in Abbildung 2.1 violett markiert zu sehen und werden durch eine Klasse gekennzeichnet. Ein möglicher Pfad ist die Abfolge der Knoten *Windig* und N , und ein Teilbaum $T_{\text{Luftfeuchtigkeit}}$ ist grün markiert zu sehen.

Die Erstellung eines Entscheidungsbaumes geschieht durch das Training eines Entscheidungsbaumalgorithmus auf einem Datensatz. Bei der Erstellung eines Entscheidungsbaumes werden Muster in dem Trainingsdatensatz gesucht [Qui86]. Anhand der entdeckten Muster wird der Datensatz

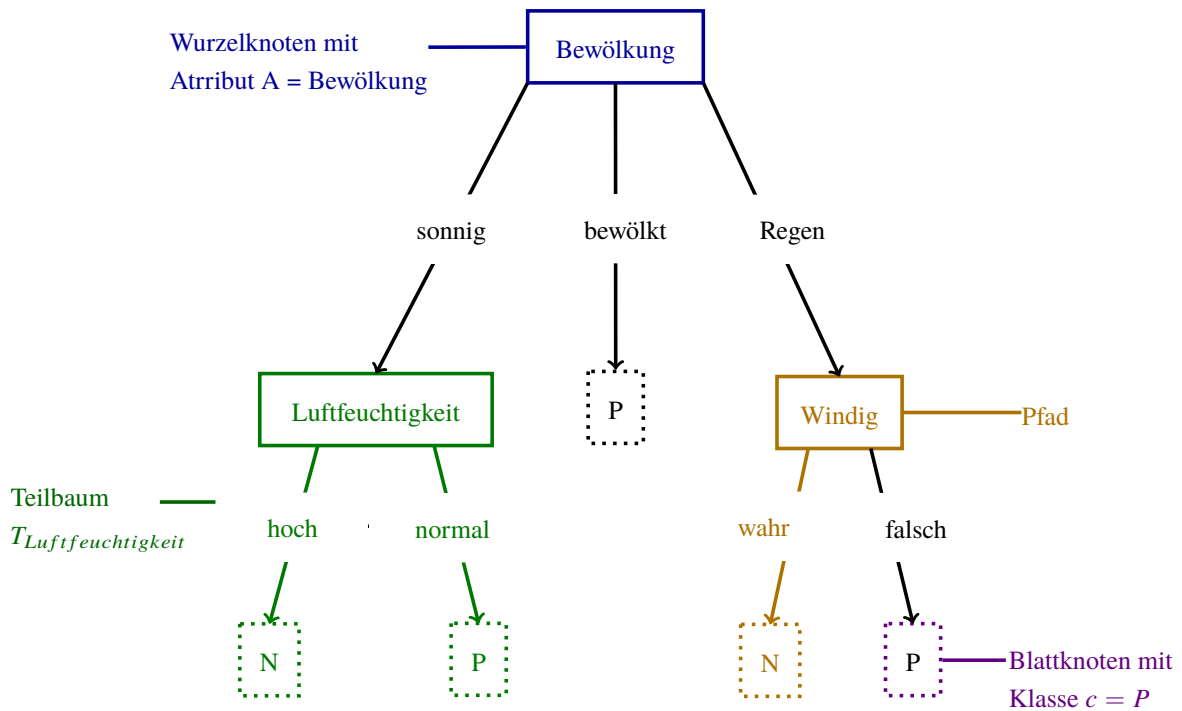


Abbildung 2.1.: Ein Entscheidungsbaum, der mit dem ID3-Algorithmus erzeugt wurde [Qui86].

horizontal in kleinere Teildatensätze aufgeteilt, die neue Knoten bilden. Auf diesen neu erstellten Knoten wird der Vorgang wieder ausgeführt. Dies erfolgt so lange auf allen durch diesen Vorgang neu erstellten Knoten, bis ein Abbruchkriterium des Algorithmus eintritt. In der Praxis erfolgt die Erkennung von Mustern anhand von Split-Metriken. Dies sind informationstheoretische Metriken, die den vermuteten Informationsgewinn einer Zerteilung des Datensatzes berechnen. Diese Zerteilung eines Datensatzes wird *Split s* genannt. Split-Metriken sind abhängig von den jeweiligen konkreten Entscheidungsbaumalgorithmen. Der Datensatz wird anhand des Attributes *A* zerteilt, das den höchsten Wert der Split-Metrik erhält. Dies kann in Abbildung 2.1 gesehen werden. Der erste Split erfolgt auf dem Attribut *Bewölkung* nach den einzelnen Werten *sonnig*, *bewölkt* und *Regen*. Der ursprüngliche Datensatz wird in drei Teildatensätze, je einer für die Instanzen mit einem der drei Werte, aufgeteilt. Jeder dieser Teildatensätze bildet einen neuen Knoten.

2.2. Entscheidungsbäume

Entscheidungsbäume sind Klassifikationsmodelle aus dem Bereich des maschinellen Lernens. Sie werden verwendet, um Daten mit unbekannter Klassenzugehörigkeit einer Klasse zuzuweisen. Mögliche Anwendungsdomänen sind die Medizin, um Patienten zu diagnostizieren oder Krankheiten zu prognostizieren, sowie die Spieltheorie, um den möglichen Gewinner einer Partie vorrauszusagen. Im Folgenden wird die Familie der Top-Down Induction of Decision Tree betrachtet und drei Algorithmen aus dieser vorgestellt, die im weiteren Verlauf der Arbeit verwendet werden.

Tabelle 2.1. Trainingsdatensatz des Baummodells aus Abbildung 2.1 [Qui86]

No.	Attribute				
	Bewölkung	Temperatur	Luftfeuchtigkeit	Windig	Klasse
1	sonnig	heiß	hoch	falsch	N
2	sonnig	heiß	hoch	wahr	N
3	bewölkt	heiß	hoch	falsch	P
4	Regen	mild	hoch	falsch	P
5	Regen	kalt	normal	falsch	P
6	Regen	kalt	normal	wahr	N
7	bewölkt	kalt	normal	wahr	P
8	sonnig	mild	hoch	falsch	N
9	sonnig	kalt	normal	falsch	P
10	Regen	mild	normal	falsch	P
11	sonnig	mild	normal	wahr	P
12	bewölkt	mild	hoch	wahr	P
13	bewölkt	heiß	normal	falsch	P
14	Regen	mild	hoch	wahr	N

2.2.1. Top-Down Induction of Decision Trees

Die Familie der Top-Down Induction of Decision Trees (TDIDT) beschreibt eine Gruppe an Algorithmen aus dem Bereich des maschinellen Lernens, die dem selben Aufbau und Ablauf folgen. Der Begriff wurde erstmals von J.R. Quinlan genannt [Qui86]. Algorithmen, die der Familie zugeordnet werden, erfüllen folgende Eigenschaften:

- Das Modell wird in Form eines Entscheidungsbaumes dargestellt.
- Die Erstellung eines Entscheidungsbaumes erfolgt nach dem Top-Down Prinzip.
- Die Häufigkeit, in der Informationen in Trainingsdaten vorkommen, ist relevant.
- Die Reihenfolge, in der die Trainingsdaten vorliegen, ist irrelevant.
- Der Lernvorgang ist nicht-inkrementell

Das erste Kriterium beschreibt eine Zuordnung des Algorithmus zu der Klasse der Entscheidungsbaume. Dies grenzt die Familie der TDIDT von anderen Klassifikationsalgorithmen, wie zum Beispiel den regelbasierten Algorithmen wie FOIL [Qui90] ab. Die weiteren vier Kriterien beschäftigen sich mit der Erstellung des Entscheidungsbaumes durch *Top-Down Induction*. Dies bezeichnet die Erstellung eines Entscheidungsbaumes in einem Top-Down-Vorgehen - beginnend vom Wurzelknoten bis zu den Blattknoten des Baumes. Weiterhin ist die Reihenfolge, in der Trainingsdaten

vorliegen nicht wichtig für die Erstellung des Entscheidungsbaumes, da lediglich die Häufigkeit, mit der Informationen in den Trainingsdaten vorkommen, Auswirkungen auf die Erstellung hat. Die letzte Eigenschaft ist die Erstellung des Baumes durch nicht-inkrementelles Lernen. Dies bedeutet, dass der Baum von Beginn des Lernvorganges an Zugriff auf den gesamten Trainingsdatensatz hat. Dies steht im Gegensatz zu inkrementellen Entscheidungsbaumalgorithmen, wie dem *Very Fast Decision Tree*-Algorithmus [DH00].

Im Folgenden wird die allgemeine Erstellung eines TDIDT-Baumes beschrieben. Dazu dient Abbildung 2.1 als Beispiel einer Darstellung eines TDIDT-Entscheidungsbaumes. Das Attribut *Bewölkung* aus der Tabelle 2.1 wurde anhand einer Split-Metrik als bestes Attribut für den Wurzelknoten aus Abbildung 2.1 ausgewählt. Die Werte des Attributes sind kategorisch und der Algorithmus zerteilt den Datensatz anhand der einzigartigen Werte des Attributes. Dies resultiert in drei Kindknoten, die den drei Attributwerten *sonnig*, *bewölkt* und *Regen* entsprechen. Dies wird an den benannten Kanten zwischen Eltern- und Kindknoten deutlich gemacht. Der linke Knoten enthält alle Instanzen des Datensatzes mit dem Wert *sonnig*, der mittlere alle Instanzen mit dem Wert *bewölkt* und der rechte Knoten enthält die restlichen Instanzen mit dem Wert *Regen* in der Spalte des Attributes *Bewölkung*. Der mittlere Knoten wird zu einem Blattknoten, da er rein ist. Dies bedeutet, dass alle Instanzen des Knotens der selben Klasse angehören. Im Fall des Beispiels ist dies die Klasse *P*. Auf den restlichen beiden Kindknoten wird rekursiv der selbe Ablauf wiederholt, bis zuletzt nur Blattknoten existieren.

Ist der Entscheidungsbaum vollständig erstellt, so können bisher ungesehene Instanzen durch den Baum zu Klassen zugeordnet werden. Als Beispiel wird eine Instanz mit den Werten *Bewölkung: sonnig, Temperatur: heiß, Luftfeuchtigkeit: normal, Windig: falsch* durch den Baum mit der Klasse *P* klassifiziert. Die Instanz wird dem Entscheidungsbaum am Wurzelknoten übergeben. Anschließend wird der Baum von dem Wurzelknoten bis zu einem Blattknoten wie folgt traversiert: Im Wurzelknoten wird bei der Instanz der Wert des Attributes *Bewölkung* überprüft. Da dieser *sonnig* beträgt, wird die Instanz an den linken Kindknoten weitergeleitet. In diesem Knoten wird der Wert des Attributes *Luftfeuchtigkeit* geprüft und an die Instanz an den rechten Kindknoten übergeben. Dieser ist ein Blattknoten mit der Klasse *P* und die Instanz wird dieser Klasse zugewiesen.

2.2.2. Entscheidungsbaumalgorithmen

Für den weiteren Verlauf dieser Arbeit werden drei Entscheidungsbaumalgorithmen in diesem Abschnitt genauer betrachtet und verglichen. Das Auswahlkriterium der drei Algorithmen ist deren Popularität. Diese ist zu einem anhand der Zitierungen auf Google Scholar festgemacht und zum anderen ist die Popularität der beiden Algorithmen C4.5 und Classification and Regression Tree (CART) weiterhin durch eine Arbeit von Wu et al. belegt, welche die populärsten Algorithmen im Bereich Data-Mining vorstellt[WKR+08]. Die ausgewählten Algorithmen sind:

- ID3: Quinlan „Induction of decision trees“ - Google Scholar Zitierungen, Stand 02.04.2019: 20250
- C4.5: Quinlan *C4.5: Programs for Machine Learning* - Google Scholar Zitierungen, Stand 02.04.2019: 35031
- CART: Breiman et al. *Classification and Regression Trees* - Google Scholar Zitierungen, Stand 02.04.2019: 40111

ID3

Der ID3-Algorithmus wurde von Quinlan entwickelt und hat den Anspruch mit wenig Rechenaufwand einen angemessen guten Entscheidungsbaum zu erstellen [Qui86]. ID3 erfordert Datensätze, deren Attribute alle kategorisch sind. Kontinuierliche Attribute werden von dem Algorithmus hingegen nicht unterstützt. Die Durchführung eines Splits auf einem kategorischen Attribut erzeugt so viele Kindknoten, wie das Attribut einzigartige Werte enthält. Dies ist in dem Datensatz aus Tabelle 2.1 und dem Baum aus Abbildung 2.1 zu sehen und ist in dem Beispiel aus Abschnitt 2.2.1 beschrieben, bei dem ein ID3-Baum erstellt wird. ID3 verwendet als Split-Kriterium den *Information Gain*, welcher auf der Shannon-Entropie [Sha48] basiert. Der Information Gain beschreibt den vermuteten Informationsgewinn, wenn der aktuell betrachtete Knoten t auf einem Attribut A geteilt wird. Die Entropie eines Knotens t wird durch $I(t)$ berechnet:

$$I(t) = - \sum_{i=1}^k \frac{|c_i|}{|t|} \log_2 \frac{|c_i|}{|t|} \quad (2.1)$$

$|c_i|$ bezeichnet die Häufigkeit des Vorkommens der Klasse c_i im Knoten t und $|t|$ ist die Anzahl aller Instanzen in t .

Die erwartete gewichtete Entropie der möglichen Kindknoten von t bei einem Split auf A wird in Gleichung 2.2 berechnet.

$$E_A(t) = \sum_{i=1}^{v_A} \frac{|t_i|}{|t|} I(t_i) \quad (2.2)$$

Hierbei bezeichnet t_i die einzelnen Kindknoten von t , mit $i \in \{1, 2, \dots, v_A\}$ für $v_A = \text{Anzahl der Kinder}$, wenn diese durch den Knoten t mit dem Attribut A erstellt werden.

Der Information Gain $gain(A)$ ist die Differenz der Entropie des Elternknotens und der erwarteten gewichteten Entropie der möglichen Kindknoten:

$$gain(A) = I(t) - E_A(t) \quad (2.3)$$

Der Information Gain wird für jedes Attribut auf dem Knoten t berechnet. Das Attribut mit dem höchsten Information Gain wird als das beste Attribut im Knoten t bezeichnet und auf diesem wird der Datensatz geteilt. Da durch den Split auf dem Attribut die gleiche Anzahl an Kindknoten entsteht, wie das Attribut einzigartige Werte hat, ist die mögliche Information des Attributes vollständig ausgeschöpft. Für weitere Splits, in dem auf den Knoten folgenden Teilbaum, wird das Attribut nicht wiederverwendet. Die Suche nach dem besten Attribut und der Split auf diesem, wird solange auf den Kindknoten wiederholt, bis eines der Abbruchkriterium des Algorithmus eintritt und ein Blattknoten erstellt wird.

Die Abbruchkriterien auf einem Knoten t des ID3 sind:

- Der Knoten ist pur
- Der Knoten enthält keine Attribute
- Der Knoten enthält keine Instanzen

Eines dieser Abbruchkriterien muss zutreffen, damit ein Knoten t terminiert wird und ein Blattknoten erstellt wird. Im ersten Fall wird der Knoten mit der einzigen Klasse die er besitzt assoziiert. Dies bedeutet, dass alle Instanzen die an den Knoten geleitet werden, mit dieser Klasse klassifiziert werden. Der zweite Fall tritt ein, wenn in dem Pfad von dem Elternknoten des betrachtete Knotens bis zum Wurzelknoten jedes Attribut aus dem Trainingsdatensatz bereits für einen Split verwendet wurde. Ein weiterer Split ist folglich nicht möglich und ein Blattknoten wird erstellt. Dem Blattknoten wird die Klasse nach dem Mehrheits-Kriterium zugewiesen. Dies bedeutet, dass die Klasse, die in den Instanzen des Knotens am häufigsten vorkommt, mit diesem assoziiert wird. Der letzte Fall tritt ein, wenn der Teildatensatz des aktuellen Knotens Werte des Split-Attributes nicht enthält. Dadurch werden für Werte, die in dem Teildatensatz nicht vorkommen, Kindknoten erstellt die keine Instanzen enthalten. Für diesen Fall wird dem Knoten die Klasse 'null' zugewiesen und es wird keine Instanz von diesem Knoten korrekt klassifiziert.

C4.5

Der C4.5 Algorithmus wurde von Quinlan als Nachfolger und Verbesserung des ID3 entwickelt [Qui93]. Die Verbesserungen enthalten die Unterstützung von kontinuierlichen Attributen sowie ein verbessertes Split-Kriterium.

Das Information-Gain-Kriterium (vgl. Gleichung 2.3) des ID3 beschreibt Quinlan als mangelbehaftet, da Attribute bevorzugt werden, die viele verschiedene mögliche Kinder erzeugen [Qui93]. Quinlan schlägt vor das Problem durch eine Weiterentwicklung des Information Gain-Kriterium, dem Gain-Ratio-Kriterium, zu beheben. Dieses führt eine Normalisierung des Information-Gain-Kriterium ein, um die Bevorteilung dessen gegenüber Attributen mit vielen möglichen Kindern zu korrigieren. Diese Normalisierung wird *split info* genannt:

$$\text{splitInfo}(A) = - \sum_{i=1}^{v_A} \frac{|t_i|}{|t|} \log_2 \frac{|t_i|}{|t|} \quad (2.4)$$

Hierbei beschreibt die $\text{splitInfo}(A)$ den möglichen Informationsgehalt eines jeden möglichen Kindes t_i , das auf dem Split mit dem Attribut A auf dem Knoten t entsteht. Das Gain Ratio-Kriterium wird als das durch die $\text{splitInfo}(A)$ normalisierte Information Gain-Kriterium beschrieben:

$$\text{gainRatio}(A) = \frac{\text{gain}(A)}{\text{splitInfo}(A)} \quad (2.5)$$

Durch die $\text{gainRatio}(A)$ wird der erwartete Informationsgewinn $\text{gain}(A)$ eines Split auf einem Attribut A in Verhältnis zu der generierten Information durch den Split $\text{splitInfo}(A)$ gesetzt.

Im Vergleich zu seinem Vorgänger unterstützt C4.5 die Verwendung von kontinuierlichen Attributen. Im Gegensatz zu kategorischen Attributen ist die Anzahl an Kindern eines Splits auf einem kontinuierlichen Attribut immer binär. Der durchgeführte Split findet einen Schwellenwert z , mit dem ein kontinuierliches Attribut A in zwei Gruppen aufgeteilt werden kann. Eine Gruppe enthält alle $a_i \in \{a_1, a_2, \dots, a_n\}$ aus A , mit $a_i \leq z$, die andere Gruppe enthält alle $a_i > z$. Dies ist in der

¹<https://archive.ics.uci.edu/ml/machine-learning-databases/labor-negotiations/C4.5/>

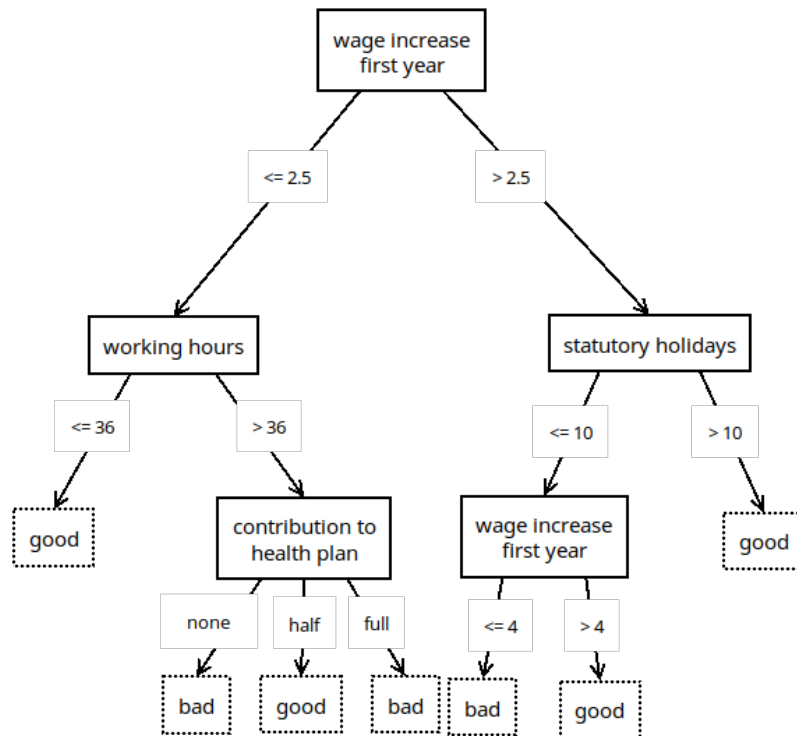


Abbildung 2.2.: Ein C4.5 Entscheidungsbaum, der auf dem labor-neg Datensatz¹ aus dem UCI Machine Learning Repository erzeugt wurde [Qui93]

Abbildung 2.2 an dem Knoten *wage increase first year* zu sehen. Der ausgewählte Schwellenwert $z = 2.5$ spaltet den Datensatz in zwei Teildatensätze. Der linke Knoten enthält alle Instanzen für die *wage increase first year* ≤ 2.5 gilt und der rechte enthält alle Instanzen mit *wage increase first year* > 2.5 .

Um das Finden des Schwellenwertes zu erleichtern wird das Attribut A zuerst sortiert. Dies ermöglicht es zwischen jeden aufeinanderfolgenden Werten a_i und a_{i+1} aus A , für $i \in \{0, 1, \dots, n-1\}$ einen Schwellenwert zu finden. C4.5 findet den Schwellenwert, in dem der Zwischenwert $\frac{a_i + a_{i+1}}{2}$ berechnet wird. Von der Mengen an Werten aus A , die unter dem Zwischenwert liegen, wird das Supremum als Schwellenwert gewählt. Dies verhindert, dass ein Schwellenwert gewählt wird, der nicht Teil des Attributes ist. Der Schwellenwert muss für alle $n-1$ möglichen Zwischenwerte berechnet werden. Der Schwellenwert, der die höchste $gainRatio(A)$ erzielt, wird als repräsentativ für das Attribut angenommen und mit denen der anderen Attributen verglichen. Es gilt wie beim ID3, dass auf dem Attribut mit dem höchsten Split-Kriterium-Wert der Datensatz des Knoten t geteilt wird.

Um dies zu verdeutlichen wird ein weiteres Mal der erste Split aus Abbildung 2.2 betrachtet. Der hierfür verwendete Zwischenwert z wurde zwischen den Werten 2.5 und 2.8 aus dem Attribut *wage increase first year* berechnet. Dies ergibt $z = \frac{2.5+2.8}{2} = 2.65$. Damit ein Schwellenwert gewählt wird, der in dem Trainingsdatensatz vorkommt, wird das Supremum des Attributes unter z gesucht. Dies ist der Wert 2.5, welcher folglich als Schwellenwert verwendet wird.

Der C4.5 verwendet nahezu die gleichen Abbruchkriterien, wie der ID3-Algorithmus. Die einzige Veränderung betrifft den Fall, in dem keine Instanzen in einem Knoten vorkommen. Für den C4.5 schlägt Quinlan vor, dass anstatt den Knoten mit 'null' zu bezeichnen, der Knoten die Klasse des Elternknotens nach dem Mehrheitskriterium erhält.

CART

Der von Breiman et al. entwickelte Entscheidungsbaumalgorithmus CART unterstützt sowohl Bäume zur Klassifikation, als auch zur Regression. In dieser Arbeit wird nur der Teil des CART betrachtet, der Klassifikationsbäume erzeugt.

Die Besonderheit des CART-Algorithmus ist die Darstellung von Splits in Form von Fragen. Abbildung 2.3 zeigt einen durch den CART-Algorithmus erstellten Baum. An diesem ist zu sehen, dass ein Split nicht anhand von Attributwerten durchgeführt wird. Stattdessen wird an einem Knoten eine Frage gestellt. Wird diese für eine Instanz mit *Ja* beantwortet, so wird die Instanz an das linke Kind geleitet, bei einem *Nein* wird sie an das rechte Kind gegeben. So wird bei einem kontinuierlichen Attribut eine Frage, wie beim Wurzelknoten aus 2.3 gestellt: *Is the wage increase ≤ 2.5* und bei kategorischen Attributen wird eine Frage gestellt, wie in dem Knoten *Does the employee contribute half to the health plan?*. Die letztere der Fragen wird so gestellt, dass gefragt wird ob ein Wert des Attributes, auf dem gesplittet wird, zu einer Menge an Werten gehört. In diesem Fall besteht diese Menge nur aus dem Wert *half* aus dem Attribut *contribution to health plan*. Auf beide Frage existieren nur die Antworten *Ja/Yes* oder *Nein/No*. Die Splits sind dadurch binär, daher generiert der CART-Algorithmus auch binäre Bäume. CART prüft alle Fragen als Split, die folgende Form haben:

- Für ein kontinuierliches Attribut A , mit $a_i \in \{a_1, a_2, \dots, a_n\}$ aus A sind alle Fragen der Art *Ist $a_i \leq z$?* enthalten
- Für ein diskretes Attribut A mit den Werten $a_i \in \{a_1, a_2, \dots, a_n\}$ werden alle Fragen der Art *Ist $a_i \in S$?* gestellt, wobei dies für jede Untermenge S aus $\{a_1, a_2, \dots, a_n\}$ gefragt wird.

In der Praxis wird der Split auf einem kontinuierlichen Attribut ähnlich zu C4.5 erstellt. Der einzige Unterschied liegt darin, dass der Zwischenwert $\frac{a_i + a_{i+1}}{2}$ gleichzeitig den Schwellenwert z darstellt und kein im Trainingsdatensatz vorhandener Wert als Schwellenwert gewählt wird. Anhand des C4.5-Entscheidungsbaumes aus Abbildung 2.2 und des CART-Entscheidungsbaumes aus Abbildung 2.3 wird der Unterschied deutlich. Der Split auf dem Wurzelknoten der beiden Bäume hat den Zwischenwert $z = 2.65$. Dieser wird von dem CART-Algorithmus als Schwellenwert verwendet, der C4.5-Algorithmus verwendet hingegen den Wert 2.5 als Schwellenwert, da dies ein im Trainingsdatensatz im Attribut *wage increase first year* real vorkommender Wert ist, der kleiner als der Zwischenwert z ist.

Als Split-Kriterium für CART schlagen die Autoren das *Gini*-Kriterium und das *Twoing*-Kriterium vor. Breiman et al. bevorzugen das Gini-Kriterium in ihrer Arbeit, da die so erzeugten Splits im Allgemeinen besser erscheinen. Aufgrund dessen beschränkt sich diese Arbeit auf die Verwendung des Gini-Kriteriums für CART.

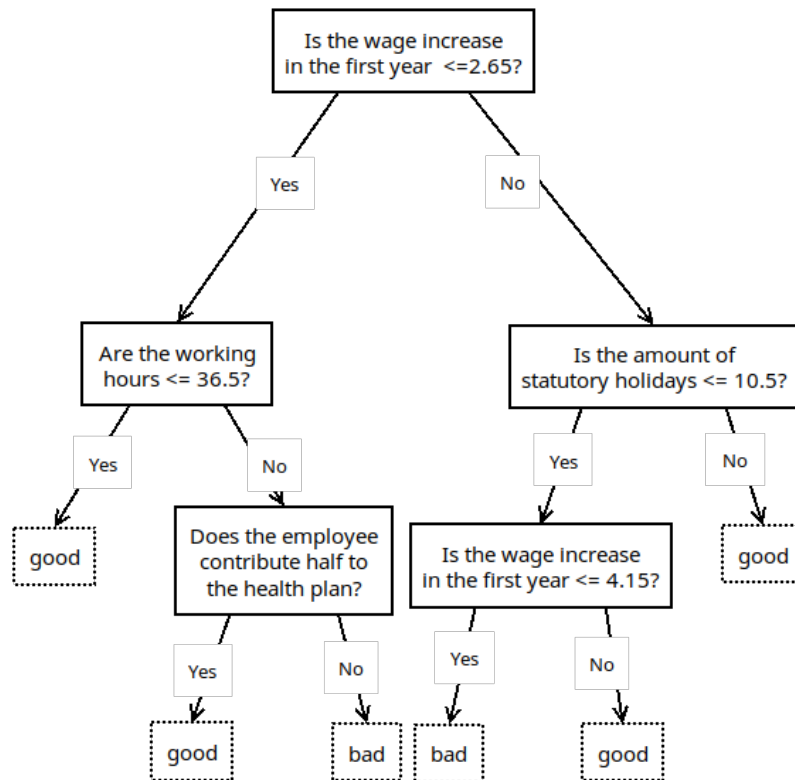


Abbildung 2.3.: Ein aus Abbildung 2.2 abgewandelter Entscheidungsbaum, der den Regeln eines CART-Baumes entspricht

Das Gini-Kriterium basiert auf dem Gini-Diversity-Index. Je geringer der Gini-Diversity-Index ist, desto purer ist der Knoten t und desto dominierender ist eine Klasse in t . Der Gini-Diversity-Index für einen Knoten t wird wie folgt berechnet [BFOS84]:

$$\begin{aligned}
 i(t) &= \left(\sum_c p(c_i|t) \right)^2 - \sum_c p^2(c_i|t) \\
 &= 1 - \sum_{c_i} p^2(c_i|t)
 \end{aligned} \tag{2.6}$$

$p(c_i|t)$ stellt das Verhältnis der Klasse c_i in dem Knoten t dar. Das Gini-Kriterium beschreibt die Abnahme des Gini-Diversity-Index bei einem Split:

$$\Delta i(s, t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{2.7}$$

L, R stehen für je das linke/rechte Kind des Knotens t und p_L, p_R bezeichnen die Verhältnisse der Instanzen, die von t in t_L, t_R übergehen, also $p_L = \frac{|t_L|}{|t|}$ und $p_R = \frac{|t_R|}{|t|}$. Der aktuelle Split auf einer Frage wird durch s bezeichnet. Das Gini-Kriterium stellt die Differenz des Gini-Diversity-Index des Elternknotens zu seinen Kindknoten bei einem Split auf dem Attribut A dar. Je höher der erzielte Wert des Gini-Kriterium ist, desto besser ist ein Split s für Knoten t geeignet.

Splits werden so lange ausgeführt, bis der Baum vollständig ausgewachsen ist. CART terminiert auf einem Knoten, wenn der Knoten rein ist oder keine Veränderung des Gini-Diversity-Index festgestellt werden kann, das Gini-Kriterium folglich den Wert null hat.

2. Grundlagen

Tabelle 2.2. Vergleich der Algorithmen ID3, C4.5 und CART

Algorithmus	ID3	C4.5	CART
Unterstützte Attribute	Kategorisch	Kategorisch, Kontinuierlich	Kategorisch, Kontinuierlich
Split-Metrik	Information Gain	Information Gain Ratio	Gini, Twoing
Anzahl der Kinder	#Werte des Attributes	Kontinuierliche: binär, Kategorisch: #Werte des Attributes	binär
Unterstützung fehlender Werte	Nein	Ja	Ja
Post-Pruning-Algorithmus	Keiner	Error-Based-Pruning	Minimal Cost-Complexity-Pruning

Vergleich

Dieser Abschnitt stellt die Entscheidungsbaumalgorithmen und ihre Eigenschaften gegenüber. In Tabelle 2.2 ist eine kurze Zusammenfassung der Eigenschaften der Algorithmen zu sehen. Die Kategorien sind zu einem Charakteristiken der Entscheidungsbäume, wie die Unterstützten Attribute, die verwendete Split-Metrik und die Anzahl der Kinder. Zum anderen sind es weitere Funktionen, die die Entscheidungsbaumalgorithmen anbieten, wie die Unterstützung von Datensätzen mit fehlenden Werte und ein für den Algorithmus erstellen Post-Pruning-Algorithmus. Die Entscheidungsbaumalgorithmen werden in diesem Abschnitt unter den genannten Kategorien miteinander verglichen.

In Spalte *Anzahl Kinder pro Attribut* ist die Verwandtschaft von ID3 und C4.5 anhand der Anzahl Kinder von kategorischen Attributen deutlich zu sehen, da sich die beiden Algorithmen dort von dem CART Algorithmus unterscheiden. Dieser erstellt seine Kinder in einer anderen Weise als C4.5 und ID3. Anstatt das Attribut anhand seiner Werte zu zerteilen, werden Fragen gestellt, anhand derer der Datensatz zerteilt wird, wie in Abschnitt 2.2.2 beschrieben. Dadurch ist ein durch CART erstellter Entscheidungsbaum ein Binärbaum, wohingegen durch ID3 und C4.5 erzeugte Entscheidungsbäume auch Mehrfachverzweigungen enthalten können.

Die Split-Metriken geben weiterhin einen Aufschluss über die Ähnlichkeit von ID3 und C4.5, da eine deutliche Verwandtschaft zwischen Information Gain und Information Gain Ratio besteht. Beide Metriken basieren auf der Shannon-Entropie[Qui93]. Das Gini- und Twoing-Kriterium unterscheiden sich von diesen, da sie auf anderen Informationstheroretischen Ansätzen basieren.

Ein weiterer Unterschied existiert zwischen ID3 und C4.5/CART. Die unterstützten Attribute der beiden Algorithmen C4.5 und CART schließen kontinuierliche Attribute mit ein. Dies bedeutet einen großen Vorteil, da die Daten, die in der heutigen Zeit gesammelt werden, oft kontinuierlich sind. Die Unterstützung von kontinuierlichen Attributen erlaubt es die Kategorisierung dieser als Vorverarbeitungsschritt zu umgehen, die für die Verwendung solcher Daten für den ID3 ansonsten nötig wäre.

Weiterhin unterstützen CART und C4.5 Datensätze, die fehlende Werte enthalten. Dies ist ein großer Vorteil, da Daten aus der Echtwelt oft fehlerhaft sind und fehlende Werte enthalten können. Quinlan schlägt für den C4.5 vor das Problem in drei Teilproblemen zu betrachten [Qui93]: Das erste Teilproblem ist die Berechnung des Split-Kriterium. Hier wird zu der Information Gain Ratio die Wahrscheinlichkeit hinzugerechnet, dass der Wert des Attributes A bekannt und unbekannt ist. Das zweite Teilproblem behandelt die Partitionierung des Datensatzes in Kindknoten, nach dem Finden des Splits. Hierzu wird eine Instanz, die bei dem Splitattribut keinen Wert besitzt, in

alle Kindknoten gewichtet verteilt. Zuletzt werden Instanzen während der Klassifikation behandelt. Anstatt eine eindeutige Klasse zurückzugeben, für eine Instanz mit fehlenden Attributwerten, erhält diese Instanz eine gewichtete Klassenverteilung über die möglichen Klassen. Breiman et al. schlagen für CART ein anderes Vorgehen vor. Für die Berechnung des besten Attributes werden nur die Werte der Attribute betrachtet, die bekannt sind. Das Aufteilen der Instanzen in die Kindknoten erfolgt, bei fehlenden Attributwerten eines Attributs A , über einen Ersatz-Split s_m für den Split s , der auf A den Datensatz teilt. Dies ist ein Split s_m auf einem anderen Attribut A_m , der am ähnlichsten zu Split s in der Aufteilung des Datensatzes ist. Die Instanz wird über s_m dem Kindknoten zugeteilt.

Zuletzt ist die Unterstützung von Post-Pruning-Methoden für die Entscheidungsbaumalgorithmen zu nennen. Post-Pruning wird in Kapitel 2.3 näher besprochen. Sowohl für C4.5 als auch CART wurde eine Post-Pruning-Methode in den jeweiligen Veröffentlichungen vorgestellt [Qui93][BFOS84]. Der Post-Pruning-Algorithmus des C4.5 ist das Error-Based Pruning. Dieses sagt anhand einer Fehlerwahrscheinlichkeit die vermuteten Klassifikationsfehler der Knoten voraus und kürzt anhand dieser den Entscheidungsbaum. Breiman et al. schlagen für CART das Minimal Cost-Complexity Pruning vor. Dieses erzeugt eine Reihe an Entscheidungsbäumen, startend mit dem maximalen ausgewachsenen Entscheidungsbaum, bis zu dem Entscheidungsbaum, der lediglich aus dem Wurzelknoten besteht. Die verschiedenen Entscheidungsbäume dazwischen entstehen durch das Prunen des jeweiligen Vorgänger Baumes. Dabei wird der Teilbaum mit der niedrigsten Cost-Complexity-Measure abgeschnitten. Der Entscheidungsbaum aus der Reihe an Bäumen, der die niedrigste Fehlerrate liefert, wird als der beste Entscheidungsbaum gewählt.

Im Allgemeinen lässt sich sagen, dass alle drei Algorithmen ähnlich sind, da sie alle dem selben Aufbau folgen, der auf Splits auf Attributen basiert. Jedoch sind die Kriterien, in denen sie sich unterscheiden, definierend für den Aufbau eines Entscheidungsbaumes. Unterschiedliche Split-Metriken erstellen unweigerlich unterschiedliche Bäume und die Anzahl der möglichen Kinder eines Knotens trägt dazu bei, wie stark ein Datensatz fragmentiert wird. Von allen drei Algorithmen stellt der ID3 jedoch die wenigsten Funktionen bereit. Der ursprüngliche ID3-Algorithmus unterstützt weder fehlende Werte in Datensätzen, noch wurde bei seiner Veröffentlichung eine Post-Pruning-Methode für den Algorithmus vorgeschlagen.

2.3. Pruning

Pruning bezeichnet das Beschneiden von Bäumen. Laut Breiman et al. [BFOS84] ist ein wichtiges Problem von Entscheidungsbäumen zu entscheiden, wann das Wachstum des Baumes gestoppt werden soll. Das Problem kann in zwei Teilprobleme aufgeteilt werden:

1. Wird das Wachstum zu früh gestoppt, so enthält der Baum zu wenige Informationen um ungesehene Daten mit einer hohen Genauigkeit zu klassifizieren. Dadurch können Knoten, die wichtig für die Klassifikation sind, fehlen. Dies ist als Underfitting bekannt [RM08].
2. Wird das Wachstum zu spät gestoppt, so ist das Resultat ein sehr großer Baum, der zu spezifisch an den Trainingsdatensatz angepasst ist. Dies wird auch als Overfitting bezeichnet [RM08]. Die daraus resultierende Genauigkeit fällt bei Testdaten schlechter aus als möglich wäre, da das Modell lediglich darauf trainiert wird bekannte Daten darzustellen, anstatt ungesehene Daten zu klassifizieren.

Es bestehen zwei Lösungsansätze für dieses Problem:

Pre-Pruning ist der Ansatz das Wachstum von Knoten während der Ausführung des Entscheidungsbaumalgorithmus zu stoppen, mit dem Ziel einen möglichst kompakten Baum zu erstellen [Qui93]. Folglich wird versucht, das Auftreten des zweiten Teilproblems zu reduzieren. Methoden die Pre-Pruning durchführen, werden in Kapitel 5 vorgestellt. Pre-Pruning besitzt den offensichtlichen Vorteil, dass das Wachstum von Teilbäumen, die den Baum verschlechtern würden, frühzeitig gestoppt wird. Diese Teilbäume werden von dem Algorithmus vor ihrer Erstellung erkannt und werden nie erstellt. Ein Problem des Pre-Pruning ist der Horizon-Effekt [Ber73]. Es besteht das Risiko bei einer Pre-Pruning-Methode, dass diese einen Teilbaum abschneidet, wenn dieser augenscheinlich das Baummodell verschlechtert. Die Möglichkeit besteht jedoch, dass der Teilbaum in späteren Splits die Genauigkeit des Baumes verbessert. Folglich kann durch Pre-Pruning das erste Problem verursacht werden.

Post-Pruning-Methoden werden nach der Erstellung der Baummodelles eingesetzt. Diese Methoden umgehen das anfangs erwähnte Problem, in dem der Entscheidungsbaum maximal ausgewachsen wird und zum Schluss ungewollte Knoten abgeschnitten werden. Es wird im Folgenden von einem vollständig ausgewachsenen Entscheidungsbaum ausgegangen, bei dem keine Methode zum frühen Wachstumsstopp eingesetzt wird. Dies bedeutet, dass alle Blätter des Baumes durch die Abbruchbedingungen der Entscheidungsbaumalgorithmen entstanden sind. Post-Pruning-Methoden schneiden ausgehend von diesem Entscheidungsbaum Teilbäume ab, die den Baum verschlechtern.

Der Vorteil von Post-Pruning-Methoden besteht darin, dass alle Informationen des Baumes vollständig vorliegen, da der Baum vollständig ausgewachsen ist. Der Nachteil des Post-Prunings liegt darin, dass der Baum vollständig auswachsen muss, was eine längere Laufzeit des Entscheidungsbaumalgorithmus zur Folge hat. Das bedeutet, dass auch Teilbäume von dem Entscheidungsbaumalgorithmus erstellt werden müssen, die im späteren Verlauf wieder abgeschnitten werden. Des Weiteren ist ein Post-Pruning-Algorithmus erst nach Ausführung des Entscheidungsbaumalgorithmus anwendbar. Die Folge ist eine längere Gesamtlaufzeit.

Alle Post-Pruning-Methoden lassen sich auf alle TDIDT-Baummodelle anwenden. Dies wird durch die Struktur und Klassifikationsaufgabe der Entscheidungsbäume gewährleistet. Die Post-Pruning-Methoden versuchen einen Baum zu erstellen, der möglichst wenige Klassifikationsfehler erzeugt. Dies wird durch das Abschneiden von Teilbäumen erreicht. Da alle Entscheidungsbäume aus Teilbäumen aufgebaut sind und aufgrund der Klassifikation Fehler entstehen können, ist eine Anwendung der Post-Pruning-Methoden auf verschiedenen Entscheidungsbäumen möglich.

In den beiden folgenden Abschnitten werden zwei Post-Pruning Algorithmen vorgestellt. Post-Pruning Algorithmen können anhand von verschiedenen Kriterien betrachtet werden. Quinlan [Qui93] unterscheidet zwischen Algorithmen, die separate Testdatensätze benötigen und Esposito et al. [EMS97] betrachtet die Algorithmen ausgiebig in ihrer Art einen Entscheidungsbaum zu wenig (Underpruning) oder zu stark (Overpruning) zu kürzen. Underpruning/Overpruning bedeutet hierbei, dass der Baum nach dem Beschneiden zu klein oder zu groß ist und der optimale Baum, also ein Baum der die bestmögliche Genauigkeit erreicht, nicht erreicht wurde. Im Folgenden werden zwei Post-Pruning-Methoden vorgestellt. Einer der Algorithmen tendiert zum Underpruning und der andere zum Overpruning.

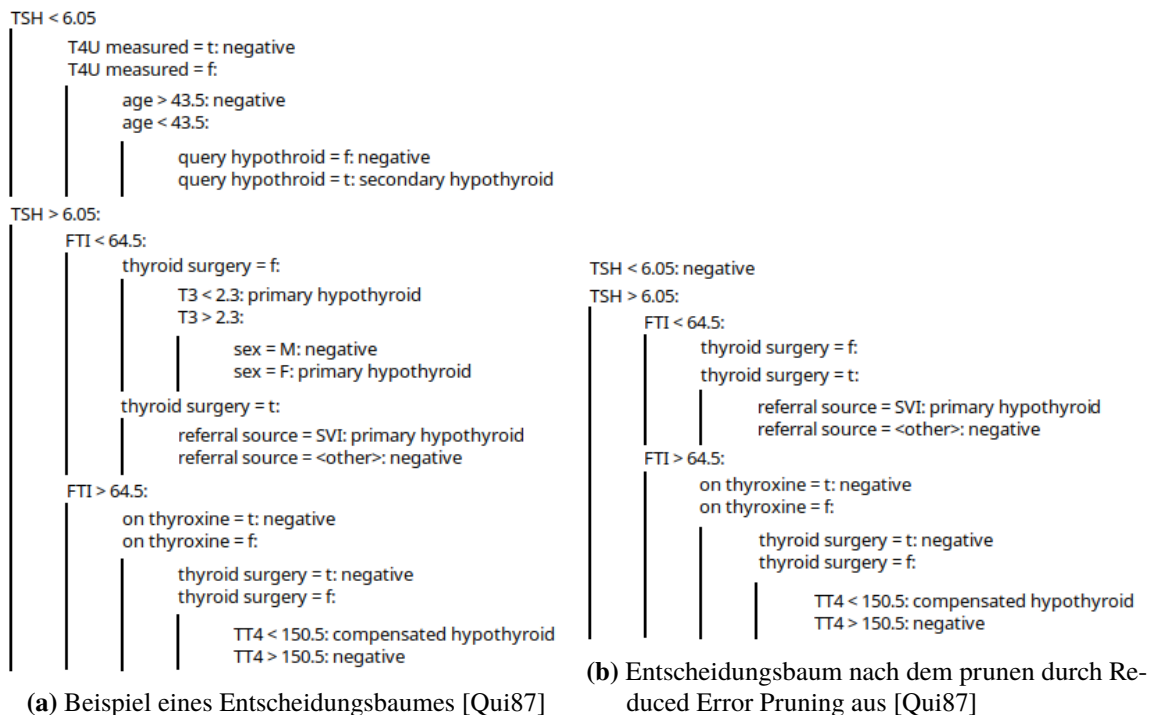


Abbildung 2.4.: Ein Entscheidungsbaum vor und nach dem kürzen durch Reduced Error Pruning

2.3.1. Reduced Error Pruning

Reduced Error Pruning (REP) ist ein von Quinlan entwickeltes Post-Pruning-Methoden [Qui87]. Dieses Verfahren tendiert zum Overpruning von Entscheidungsbäumen [EMS97]. Aufgrund seines einfachen Konzeptes ist dieses Verfahren das simpelste Post-Pruning-Methoden [EMS97]. REP erlaubt es einen Teilbaum T_t durch einen Blattknoten t zu ersetzen. Dies trifft ein, wenn der Klassifikationsfehler von T_t größer oder gleich groß wie der des Blattes t ist. Die mit t assoziierte Klasse stellt dabei die nach dem Mehrheitskriterium am häufigsten in t vorkommende Klasse dar. Ferner gilt, dass T_t keinen Unterteilbaum enthalten darf, für den das im vorigen Satz beschriebene gilt. Folglich wird der Baum für REP in einer Bottom-Up-Weise durchlaufen. Die Bestimmung des Klassifikationsfehlers erfolgt durch die Klassifizierung eines zusätzlichen Testdatensatzes auf dem vollständig ausgewachsenem Entscheidungsbaum. Bei jedem Knoten wird dabei die Anzahl an falschen Klassifikationen als Klassifikationsfehler angenommen. Bei einem inneren Knoten wird die mit ihm assoziierte Klasse nach dem Mehrheits-Kriterium ausgewählt und die bei dem Knoten ankommende Instanz mit dieser Klasse klassifiziert und überprüft, ob diese Klasse mit der eigentlichen Klasse der Instanz übereinstimmt.

Die Anwendung von REP wird anhand des Baumes aus der Abbildung 2.4a deutlich gemacht. Dieser Baum klassifiziert einen Testdatensatz, anhand dessen den Knoten Klassifikationsfehler zugewiesen werden. Als Beispiel zu betrachten ist der erste Teilbaum $T_{TSH < 6.05}$. Der Testdatensatz weist jedem der Blattknoten einen Klassifikationsfehler zu. Für das Beispiel wird angenommen, dass

2. Grundlagen

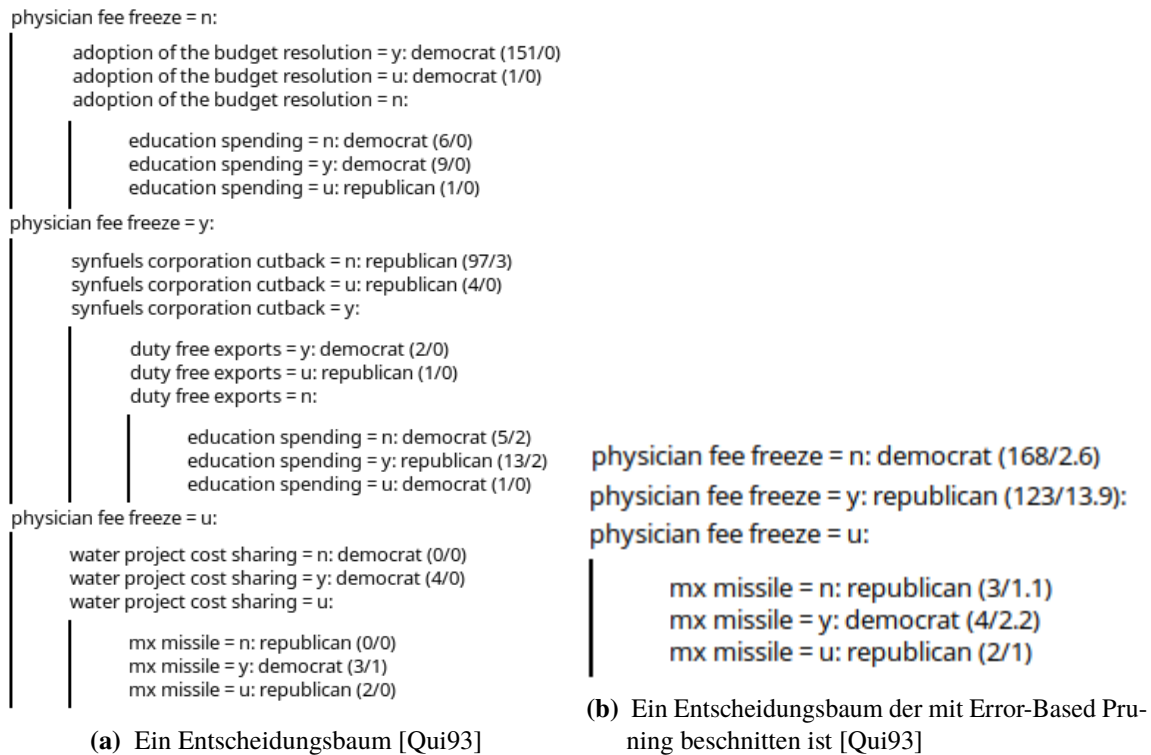


Abbildung 2.5.: Ein Entscheidungsbaum vor und nach dem kürzen durch Error-Based Pruning

der Klassifikationsfehler von $T_{TSH < 6.05}$ 4 beträgt. Dem Knoten $TSH < 6.05$ mit der populärsten Klasse *negative* wird der Klassifikationsfehler 1 zugewiesen. Dadurch ist der Klassifikationsfehler von $T_{TSH < 6.05}$ größer als der des Knoten $TSH < 6.05$ und $T_{TSH < 6.05}$ wird durch den Blattknoten $TSH < 6.05 : negative$ ersetzt. Der gekürzte Baum ist in Abbildung 2.4b zu sehen.

2.3.2. Error-Based Pruning

Error-Based Pruning (EBP) ist das von Quinlan für den C4.5 Algorithmus entwickelte Post-Pruning Verfahren [Qui93]. Dieses Verfahren tendiert zum Underpruning von Entscheidungsbäumen [EMS97]. EBP ist eine Weiterentwicklung des Pessimistic Pruning, das Quinlan einige Jahre zuvor entwickelte [Qui87], jedoch wird im Gegensatz zum Pessimistic Pruning der Baum in einem Bottom-Up-Verfahren durchlaufen und es wird kein zusätzlicher Testdatensatz benötigt. Zusätzlich zum Ersetzen der Teilbäume durch Blätter, erlaubt EBP das Ersetzen eines Teilbaumes durch seinen am stärksten durch Instanzen besuchten Unterteilbaum.

Für das EBP wird der zu kürzende Entscheidungsbaum in einem Bottom-Up-Verfahren durchlaufen. Bei jedem Teilbaum T_t wird überprüft, ob dieser durch einen Blattknoten t oder den am stärksten besuchten Unterteilbaum von T_t ersetzt werden kann. Dies ist der Fall, wenn der vermutete Klassifikationsfehler des Blattknoten t oder des Unterteilbaumes kleiner oder gleich ist, als der vermutete Klassifikationsfehler des Teilbaumes T_t . Die vermuteten Klassifikationsfehler werden über die Fehlerwahrscheinlichkeit der Knoten berechnet. Die Fehlerwahrscheinlichkeit eines Knotens kann aus einer (A-Posteriori) Wahrscheinlichkeitsverteilung über die Anzahl der Instanzen N und Anzahl

der Klassifikationsfehler E des Knotens berechnet werden. Die Anzahl der Durchläufe für diese Wahrscheinlichkeitsverteilung kann als N angenommen werden und E als die Ereignisse, die in N vorkommen. Diese Wahrscheinlichkeitsverteilung besitzt zwei Konfidenzlimits, von denen das obere Konfidenzlimit U_{CF} von hohem Interesse ist, da EBP einen pessimistischen Ansatz verfolgt. Gesucht ist dieses Konfidenzlimit für ein Konfidenzniveau CF , welches für den C4.5 standardmäßig als $CF = 25\%$ gegeben ist [Qui93]. Als Wahrscheinlichkeitsverteilung wird für diese Post-Pruning-Methode eine Binomialverteilung gewählt. Das obere Konfidenzlimit dieser Binomialverteilung wird als $U_{CF}(E, N)$ dargestellt und kann als Fehlerwahrscheinlichkeit für den Knoten t angenommen werden. Für die Berechnung des Klassifikationsfehlers wird angenommen, dass der Entscheidungsbaum einen unbekanntem Testdatensatz der selben Größe wie der Trainingsdatensatz klassifiziert. Dies erlaubt es den Klassifikationsfehler eines Knoten t mithilfe der Fehlerwahrscheinlichkeit als $N * U_{CF}(E, N)$ zu berechnen. Der Klassifikationsfehler eines Teilbaumes wird als die Summe aller vorhergesagten Klassifikationsfehler seiner Blätter berechnet. Ist der Klassifikationsfehler eines Teilbaumes T_t größer als der seines Elternknotens t , so wird t zum Blatt und ersetzt T_t , wobei die mit t assoziierte Klasse nach dem Mehrheits-Kriterium ausgewählt wird. Ist die Fehlerrate von T_t und die des Elternknotens t größer als die des größten Unterteilbaumes aus T_t , so wird T_t durch seinen größten Unterteilbaum ersetzt.

Deutlich gemacht wird dies anhand von Abbildung 2.5a und 2.5b. Die Zahlen in den Klammern bezeichnen die Anzahl der Instanzen N , die diesen Knoten erreicht haben und die hintere Zahl ist die Anzahl der Klassifikationsfehler E auf diesem Knoten. Die Klassifikationsfehler im gekürzten Baum bezeichnen den vorhergesagten Klassifikationsfehler des Knotens. Im Folgenden wird der Teilbaum *adoption of the budget resolution = n* betrachtet. Anhand dessen wird exemplarisch gezeigt, wie durch EBP ein Teilbaum durch einen Blattknoten ersetzt wird. Der Klassifikationsfehler der Blattknoten des Teilbaumes ist jeweils 0, somit beträgt der Klassifikationsfehler des Teilbaumes ebenso 0. Der Klassifikationsfehler des Knotens *adoption of the budget resolution = n* beträgt 1, wenn er durch einen Blattknoten mit der Klasse *democrat*, die nach dem Mehrheitskriterium gewählt wurde, dargestellt wird. Die durch die Binomialverteilung berechneten Klassifikationsfehler ergeben einen höheren Fehler für den Teilbaum als für den Blattknoten, deswegen wird der Teilbaum durch das Blatt *adoption of the budget resolution = n: democrat* ersetzt, wie in Abbildung 2.5b zu sehen. In Abbildung 2.5b ist an dem gekürzten Baum zu sehen, wie das Ersetzen eines Teilbaumes durch seinen größten Unterteilbaum funktioniert. Hierfür wird der Teilbaum *physician fee freeze = u* betrachtet in Abbildung 2.5a. Der größte Unterteilbaum ist *water project cost sharing = u*. Dieser wird an die Stelle des Teilbaumes *physician fee freeze = u* gesetzt, indem die Kinder des Unterteilbaumwurzelknotens die Kinder des Teilbaumwurzelknotens ersetzen.

3. Verwandte Arbeiten

Das in Abschnitt 2.3 genannte Problem des Prunings von Entscheidungsbäumen ist ausgiebig im Bereich des Post-Pruning behandelt. Es existieren sowohl Veröffentlichungen in denen Post-Pruning-Methoden vorgestellt werden, sowie ausführliche Vergleiche und Evaluationen der verschiedenen Methoden. Im Bereich der Entscheidungsbäume existieren nur wenige Arbeiten, die Pre-Pruning-Methoden betrachtet haben. Einige Arbeiten, die diese Methoden behandelt haben werden im Folgenden vorgestellt.

In Kapitel 3.1 wird Literatur vorgestellt, in der Post-Pruning und Pre-Pruning-Methoden vorgeschlagen werden. Anschließend werden in Kapitel 3.2 Data-Mining-Werkzeuge betrachtet, die Pre-Pruning für Entscheidungsbaumalgorithmen implementiert haben.

3.1. Wissenschaftliche Arbeiten

Dieser Abschnitt behandelt Arbeiten, die Post-Pruning- oder Pre-Pruning-Methoden vorstellen, sowie Arbeiten, die diese evaluieren und vergleichen. In Kapitel 3.1.1 werden entsprechende Arbeiten für Post-Pruning-Methoden vorgestellt und im darauffolgenden Kapitel 3.1.2 wird das selbe für Pre-Pruning vorgenommen.

3.1.1. Post-Pruning

Minimal Cost-Complexity Pruning ist der Post-Pruning-Algorithmus, der von Breiman et al. für den CART-Algorithmus vorgeschlagen wurde[BFO84]. Der Algorithmus basiert auf der sogenannten Cost-Complexity-Measure. Dies ist eine Metrik, die eine Kombination der Missklassifikationsrate und der Komplexität des Baumes darstellt. Der initiale Baum, von dem im Rahmen dieses Algorithmus ausgegangen wird, ist der maximal ausgewachsene Baum, der durch CART erzeugt wurde. Ausgehend von diesem Baum wird eine Reihe an Bäumen erzeugt. Jeder Baum unterscheidet sich von seinem Vorgänger, in dem ein Teilbaum abgeschnitten wurde, der mithilfe der Cost-Complexity-Measure ausgewählt wurde. Der letzte Baum der Reihe besteht nur aus dem Wurzelknoten. Der beste Baum der Reihe wird aufgrund der Fehlerrate ausgewählt. Diese wird entweder mit einem extra Testdatensatz oder über cross-validation bestimmt.

Error-Based Pruning ist der von Quinlan für den C4.5 vorgeschlagene Entscheidungsbaumalgorithmus[Qui93]. Diese Post-Pruning-Methode kommt ohne zusätzlichen Testdatensatz für die Ermittlung des Klassifikationsfehler aus. Der vermutliche Klassifikationsfehler eines Knoten t wird anhand der Fehlerwahrscheinlichkeit des Knotens vorhergesagt. Die Fehlerwahrscheinlichkeit kann über die obere Grenze einer Binomialverteilung mit dem Klassifikationsfehler E und den Instanzen N des Knoten berechnet werden. Diese obere Grenze wird durch $U_{CF}(E, N)$ dargestellt, mit CF dem Konfidenzlevel der oberen Grenze. Dieses schlägt Quinlan als $CF = 25\%$ vor. Für die Berechnung

des vermutlichen Klassifikationsfehlers eines Knotens wird angenommen, dass der Baum ein Testdatensatz klassifiziert, welches den selben Umfang wie der Trainingsdatensatz besitzt. Dies erlaubt einen Klassifikationsfehler für einen Knoten t als $N * U_{CF}(E, N)$ zu berechnen.

Einen Vergleich mehrerer Post-Pruning-Methoden stellt Quinlan hinsichtlich der Genauigkeit und Einfachheit der gekürzten Entscheidungsbäume an [Qui87]. In der selben Veröffentlichung stellt Quinlan auch zwei weitere Post-Pruning-Methoden namens *Reduced Error Pruning* und *Pessimistic Pruning* vor. Diese werden mit dem Minimal Cost-Complexity Pruning und der *Simplifying to Production Rules*-Methode, welche einen Entscheidungsbaum in eine Menge an Produktionsregeln umwandelt, verglichen. Quinlan kommt dabei zu dem Schluss, dass neben der Vereinfachung der Bäume durch die Reduzierung deren Größe durch Pruning, auch die Genauigkeit der gekürzten Entscheidungsbäume ansteigt.

Eine ausführliche Evaluation von sechs verschiedenen Post-Pruning-Methoden führt Esposito et al. durch [EMS97]. Diese Methoden sind Minimal Cost-Complexity-Pruning [BFOS84], EBP [Qui93], REP [Qui87], Pessimistic Pruning [Qui87], Critical Value Pruning [Min87] und Minimum Error Pruning [NB87]. Die Ergebnisse zeigen eine gewisse Tendenz einiger Post-Pruning-Methoden zum Overpruning oder Underpruning von Entscheidungsbäumen. Weiterhin wird die Größe der entstandenen Bäume, sowie deren Genauigkeit verglichen. Esposito et al. kommen zu dem Ergebnis, dass Post-Pruning meistens die Genauigkeit des Baumes nicht mindert und dass das Beiseitelegen von Daten des Gesamtdatensatzes für Post-Pruning-Methoden von Nachteil ist.

Im Bereich der Post-Pruning-Methoden existieren zahlreiche Arbeiten, die sowohl verschiedene Post-Pruning-Methoden, sowie die Auswertung und Evaluation dieser ausführlich behandeln. Es ist bereits viel über die Vor- und Nachteile dieser Methoden bekannt, im Gegensatz zu Methoden aus dem Bereich des Pre-Prunings. In Kapitel 3.1.2 werden existierende Werke aus dem Bereich des Pre-Pruning genauer betrachtet.

3.1.2. Pre-Pruning

Sowohl Quinlan als auch Breiman et al. erörtern die Möglichkeit des Pre-Prunings in den Veröffentlichungen ihrer Entscheidungsbaumalgorithmen [BFOS84; Qui93]. Breiman et al. und Quinlan schlagen einen Schwellenwert für ihr Split-Kriterium als Pre-Pruning Methode vor. Dieser Schwellenwert soll Splits mit geringen Werten des Split-Kriterium verhindern, um einen zu geringen Informationsgewinn zu vermeiden. Ab einem zu geringen Informationsgewinn ist es zweifelhaft, dass dieser noch informativ ist [BFOS84]. Das Finden eines Schwellenwertes, der weder zu klein noch zu groß ist, stellte sich laut Breiman et al. jedoch als schwierig heraus. Bei einem zu geringen Schwellenwert sind die Resultate zu große Bäume, bei denen Overfitting auftritt, und bei einem zu hohen Schwellenwert entstehen Bäume, die zu klein sind und deren Äste zu früh abgeschnitten werden.

Quinlan betrachtet die Möglichkeit der Verwendung des χ^2 -Test für Pre-Pruning [Qui86]. Im Bereich der Entscheidungsbäume kann damit die Irrelevanz eines Attributes A zu der Klassenverteilung, in den durch den Split s auf A erstellten Kindknoten beschrieben werden. Die Irrelevanz beschreibt, ob keine Abhängigkeit zwischen dem Attribut A und der Klasse C besteht. Kann diese Irrelevanz nicht mit einem hohen Konfidenzlevel abgelehnt werden, so schlägt Quinlan vor dieses Attribut

nicht für s zu betrachten. Quinlan beschreibt dieses Verfahren als effektiv, um das Wachstum der Bäume zu vermindern. Jedoch verwirft er dies in einer späteren Veröffentlichung, da die erzielten Ergebnisse zu unregelmäßig waren [Qui93].

Eine Arbeit von Bramer schlägt J-Pruning als Pre-Pruning-Methode vor. Anstatt einen Knoten zu betrachten, schlägt Bramer vor den Informationsgehalt einer Regel zu berechnen. Eine Regel wird in einem Entscheidungsbaum durch einen Ast dargestellt. Über die J-Measure [GS91] wird der Informationsgehalt des aktuell betrachteten Astes berechnet und mit dem Informationsgehalt des Astes nach dem Split verglichen. Wenn der Informationsgehalt abfällt ist die Annahme, dass sich weitere Splits auf diesem Ast nicht als sinnvoll erweisen. Bei einem Vergleich eines Entscheidungsbaumalgorithmus mit und ohne J-Pruning stellte sich heraus, dass die durchschnittlichen Genauigkeit der durch J-Pruning erzielten Ergebnisse nur geringfügig schlechter ist, als die des Entscheidungsbaumes. Im Ausgleich dazu fallen die mit J-Pruning erzeugten Entscheidungsbäume wesentlich simpler aus. Dies wird an der Anzahl der Regeln festgemacht, die die Bäume enthalten.

Die verfügbare Literatur im Bereich des Pre-Pruning ist aktuell spärlich. Meistens werden Pre-Pruning-Methoden die vorgeschlagen wurden, als zu ungenau abgetan, ohne eine genaue Begründung zu liefern. Eine genaue Evaluation der Methoden fehlt aktuell. Diese ist jedoch nötig, um Pre-Pruning-Methoden anwenden zu können, da viele der Methoden Werte benötigen, die von einem Nutzer übergeben werden. Diese Arbeit beschäftigt sich mit diesem Thema und evaluiert dafür diverse Pre-Pruning-Methoden.

3.2. Data-Mining-Werkzeuge

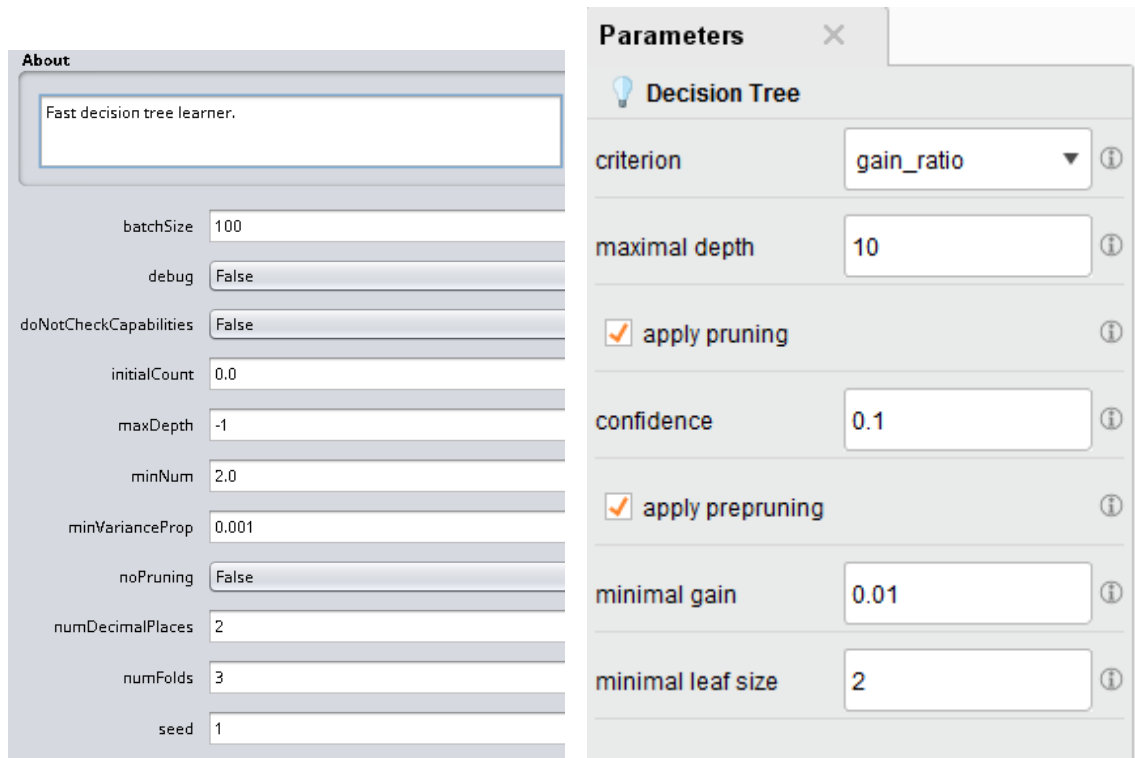
Dieser Abschnitt betrachtet die Verwendung von Pruning-Methoden in Data-Mining-Werkzeugen. Betrachtet werden sowohl die Verfügbarkeit von Post- als auch Pre-Pruning-Methoden für Entscheidungsbaumalgorithmen. Trotz der geringen wissenschaftlichen Abdeckung des Pre-Pruning bieten einige Werkzeuge dem Nutzer die Möglichkeit dieses für Entscheidungsbäume zu verwenden. Im folgenden Abschnitt werden die beiden Data-Mining-Werkzeuge Waikato Environment for Knowledge Analysis (WEKA) und RapidMiner vorgestellt, da diese Pre-Pruning-Methoden für ihre Entscheidungsbaumalgorithmen beinhalten.

3.2.1. Waikato Environment for Knowledge Analysis

WEKA¹ ist ein weit verbreitetes Data-Mining-Werkzeug, mit einer großen Anzahl an Mining-Algorithmen [HFH+09]. Für den Bereich der Entscheidungsbäume bietet WEKA acht Entscheidungsbaumalgorithmen an, darunter Random Trees und eine eigene Implementierung des C4.5 Algorithmus von Quinlan. Als Optionen für die Entscheidungsbaumalgorithmen existieren je nach Algorithmus Parameter für das Pre- und Post-Pruning der generierten Entscheidungsbaummodelle. In Abbildung 3.1a ist eine Auswahl an Einstellungen für einen Entscheidungsbaumalgorithmus zu sehen, der ausgewählt wurde, da er in WEKA zwei Pre-Pruning-Methoden beinhaltet. WEKA unterstützt in der aktuellen Version 3.8.3 nicht die beiden Pre-Pruning-Methoden auf jedem Entscheidungsbaumalgorithmus. Es ist möglich über die Einstellung *noPruning* das Post-Pruning des

¹<https://www.cs.waikato.ac.nz/ml/weka/>

3. Verwandte Arbeiten



(a) Einstellungen eines Entscheidungsbaum-Knotens in Weka (b) Einstellungen eines Entscheidungsbaum-Knotens in RapidMiner

Abbildung 3.1.: Einstellungen von Data-Mining-Werkzeugen zu Pruning

Baumes zu ermöglichen. Für diesen ausgewählten Algorithmus wird REP als Post-Pruning-Methode genutzt. Die Pre-Pruning-Einstellungen *maxDepth* und *minNum* erlauben es die maximale Tiefe des Baumes sowie die minimal benötigte Anzahl an Instanzen in einem Knoten festzulegen. Wird die Anzahl an Instanzen in einem Knoten unterschritten oder erreicht ein Knoten die maximale Tiefe, so wird ein Blattknoten erstellt. WEKA bietet für die beiden Einstellungen feste Standardparameter für alle Datensätze. Allerdings hat der Nutzer selbst keinen Ansatz welcher Wert für die Einstellungen einen guten Baum erzeugen wird. Zum aktuellen Zeitpunkt (Version 3.8.3) bietet das Werkzeug dem Nutzer keinerlei Unterstützung einen nützlichen Wert zu finden.

3.2.2. RapidMiner

RapidMiner² ist ein häufig genutztes Data-Mining-Werkzeug, das den gesamten Data-Mining-Prozess von der Vorverarbeitung der Daten bis zu der Auswertung der erlernten Modelle unterstützt. Das Werkzeug enthält Implementierungen³ der Entscheidungsbaumalgorithmen Chi-square Automatic Interaction Detectors (CHAID), ID3 und Random Forest. Für jeden der Algorithmen sind die Parameter aus Abbildung 3.1b anwendbar. Die Einstellung *apply pruning* erlaubt es Pessimistic

²<https://rapidminer.com/>

³https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/parallel_decision_tree.html

Error Pruning von Quinlan aus [Qui87] auf den generierten Entscheidungsbaum anzuwenden. Das Feld *confidence* beschreibt dabei das Konfidenzlevel, dass für das Pessimistic Error Pruning verwendet werden soll. Die Felder *maximal depth*, *minimal gain* und *minimal leaf size* erlauben es Pre-Pruning-Einstellungen auf die Erzeugung des Entscheidungsbaumes anzuwenden, wenn *apply prepruning* aktiviert ist. Diese begrenzen die maximale Tiefe des Baumes (*maximal depth*), setzen einen Schwellenwert für die Split-Metrik (*minimal gain*) und begrenzen die minimale Anzahl an Instanzen, die ein Knoten enthalten muss, um kein Blatt zu erstellen (*minimal leaf size*). RapidMiner verhält sich ähnlich wie WEKA. Es existieren Standardparameter für die Pre-Pruning Methoden, dem Nutzer wird in der aktuellen Version (9.2) ansonsten kein Anhaltspunkt für effektive Einstellungen bereitgestellt.

3.3. Zusammenfassung

Im Allgemeinen lässt sich sagen, dass Pre-Pruning-Methoden in Entscheidungsbäumen als Möglichkeit betrachtet, jedoch ohne tiefere Forschung verworfen wurden. Keine der Arbeiten führt weder eine umfassende Auswertung der Methoden durch, noch wird die Auswirkung auf die Entscheidungsbäume ausgiebig diskutiert. Im Gegensatz dazu existieren Arbeiten, die Post-Pruning-Methoden ausführlich vorstellen oder evaluieren. Zu nennen sind speziell die Arbeiten von Esposito et al. und Quinlan, die jeweils mehrere Post-Pruning-Methoden vergleichen [Qui87] [EMS97]. Insbesondere erstere Veröffentlichung stellt eine ausführliche Übersicht zu diversen Post-Pruning Verfahren dar. Data-Mining-Werkzeuge bieten ein ähnliches Bild. Wenn auch Einstellungen zur Durchführung von Pre-Pruning auf Entscheidungsbäumen existieren, so fehlen Informationen zu Parameterwerten, die es erlauben die Methoden effektiv zu nutzen. Diese Arbeit löst das Problem, indem eine ausführliche Evaluation von Pre-Pruning-Methoden auf verschiedenen Entscheidungsbaumalgorithmen durchgeführt wird und einem Analysten ein Werkzeug gibt, mit dem er geeignete Werte für Pre-Pruning-Methoden finden kann.

4. Konzept

Entscheidungsbäume sind eine Gruppe an Klassifikationsalgorithmen und gehören zu den populärsten Algorithmen im Bereich des Data Mining [WKR+08]. Sie sind durch ihre Visualisierung als Baumdiagramm charakterisiert. Durch diese Art der Visualisierung sind sie gut für menschliche Analysten verständlich. Anhand der Visualisierung und der Laufzeit ist ein weitreichendes Problem der Entscheidungsbäume erkennbar. Durch die steigende Komplexität und Menge der Daten werden immer größere Entscheidungsbäume erstellt, die für Analysten schwieriger verständlich sind. Die Erstellung der Bäume benötigt weiterhin aufgrund der zunehmenden Menge an Daten mehr Laufzeit.

Aktuell existieren zwei Verfahrensgruppen für das Pruning (vgl. Kapitel 2.3), mit deren Hilfe die Größe der Bäume verringert werden kann. Eine der beiden Verfahrensgruppen ist das Post-Pruning. Das Post-Pruning setzt nach der Erstellung der Entscheidungsbäume an und schneidet von einem maximal ausgewachsenen Entscheidungsbaum Teilbäume ab, die die Qualität des Baumes verschlechtern. Dadurch entsteht eine längere Laufzeit für die Erstellung des Entscheidungsbaumes und eine hohe Gesamtlaufzeit, da die jeweilige Post-Pruning-Methode nach der Erstellung des Baumes zusätzlich ausgeführt werden muss. Die zweite Verfahrensgruppe ist das Pre-Pruning. Diese Methoden setzen bereits während der Erstellung eines Entscheidungsbaumes an und verhindern das Wachsen von Teilbäumen, die die Qualität des Baumes verschlechtern würden. Die Gesamtlaufzeit für die Erstellung eines Entscheidungsbaumes fällt folglich geringer aus, da Teile des Baumes nicht erstellt werden müssen.

In diesem Kapitel wird ein allgemeingültiges Verfahren vorgestellt, mit dem TDIDT-Entscheidungsbäume durch Pre-Pruning-Methoden gekürzt werden können. Das Ziel dieses Verfahrens ist es, die Pre-Pruning-Methoden auf allen TDIDT-Entscheidungsbäumen anwenden zu können. Dies erlaubt es diese evaluieren zu können und so Schwellenwerte für die Methoden zu finden, mit denen Entscheidungsbäume effektiv gekürzt werden können. In Abschnitt 4.1 wird die allgemeine Erstellung eines TDIDT-Entscheidungsbaumes vorgestellt und mögliche Ansätze für Pre-Pruning-Methoden vorgeschlagen. Die Pre-Pruning-Methoden müssen mit der Genauigkeit der Entscheidungsbäume gegenübergestellt werden, um so Schwellenwerte für die Methoden zu bestimmen, mit denen eine gewünschte Genauigkeit erreicht werden kann. Die Relation der Werte und der Genauigkeit soll dann für das Pre-Pruning von Entscheidungsbäumen auf bisher ungesehenen Datensätze verwendet werden. In Kapitel 4.2 wird ein Verfahren vorgestellt, mit dem dieser Zusammenhang zwischen den Pre-Pruning-Methoden und der Genauigkeit gefunden werden kann.

4.1. Erstellung der TDIDT-Entscheidungsbäume

In diesem Abschnitt wird der allgemeine Ablauf der Erstellung eines TDIDT-Entscheidungsbaumes vorgestellt. Dieser ist nötig, um ein Verfahren zu entwickeln, mit dem Pre-Pruning-Methoden auf allen TDIDT-Entscheidungsbaumalgorithmen angewendet werden können. Der Erstellungsvorgang

4. Konzept

eines TDIDT-Entscheidungsbaum kann auf die Erstellung der einzelnen Knoten des Baumes zurückgeführt werden. Dies ist der Fall, da der selbe Ablauf für die Erstellung eines Knotens rekursiv auf seinen Kindknoten ausgeführt wird. Somit werden alle Knoten des Baumes und folglich der ganze Baum anhand dieses Ablaufes erstellt. Die Erstellung ist in drei Schritte unterteilbar. Diese drei Schritte sind:

S1 Auswahl des besten Attributes:

Im ersten Schritt wird das Attribut ausgewählt, das am besten für den Split auf diesem Knoten geeignet ist. Das geschieht durch das Split-Kriterium des Entscheidungsbaumalgorithmus wie zum Beispiel das Gini-Kriterium für CART. Dazu werden alle Attribute des Datensatzes von dem Split-Kriterium nach ihrer Nützlichkeit für den Baum evaluiert. Die Nützlichkeit für einen Baum ist höher, je reiner die Kindknoten sind, die durch den Split auf einem Attribut entstehen. Ein Knoten ist rein, wenn die Instanzen, die er enthält, nur einer Klasse angehören. Das Attribut mit dem besten Wert des Split-Kriteriums wird für die Durchführung des Splits ausgewählt.

S2 Durchführung des Splits:

Dieser Schritt bezeichnet die Durchführung eines Split auf einem Knoten, mit dem ausgewählten besten Attribut aus Schritt S1 für diesen Knoten. Der Split erstellt mehrere Kindknoten aus dem Datensatz des aktuellen Knotens, auf dem der Split durchgeführt wird. Dazu werden die Instanzen des Datensatzes des aktuellen Knotens auf die Kindknoten aufgeteilt. Die Anzahl der Kinder unterscheidet sich je nach Attribut und Algorithmus.

S3 Überprüfung der Abbruchkriterien:

Die neu erstellten Kindknoten aus Schritt S2 werden zuletzt auf Abbruchkriterien überprüft. Sollte eines der Abbruchkriterien des TDIDT-Algorithmus eintreffen, so wird das Wachstum des Baumes an diesem Knoten terminiert. Ein mögliches Abbruchkriterium ist, dass der erstellte Knoten rein ist. Folglich wird ein Blattknoten aus dem Knoten erstellt und kein Split wird auf diesem ausgeführt. Trifft hingegen kein Abbruchkriterium zu, so fängt der Ablauf auf jedem der Kindknoten bei Schritt S1 an.

Abbildung 4.1 zeigt den Ablauf der Erstellung eines Entscheidungsbaumes. Pre-Pruning-Methoden müssen an einem dieser Schritte ansetzen, um auf einen TDIDT-Entscheidungsbaum angewendet werden zu können. Auf jedem in Schritt S2 erzeugten Kindknoten wird Schritt S3 ausgeführt. Hierbei teilt sich der Vorgang für jeden Kindknoten auf, und Schritt S1 wird auf jedem der Kindknoten getrennt ausgeführt. Die drei Schritte werden im Folgenden ausführlich betrachtet.

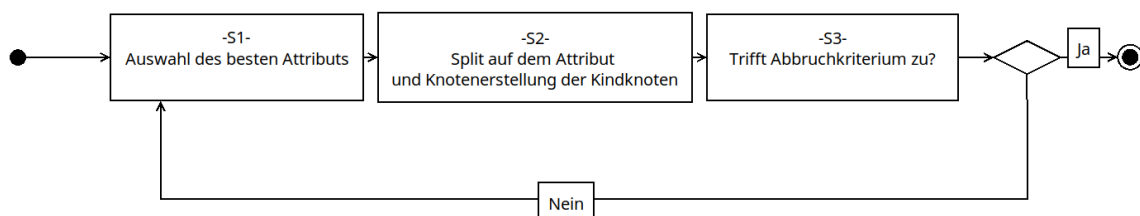


Abbildung 4.1.: Der allgemeine Aufbau eines TDIDT-Entscheidungsbaumalgorithmus

4.1.1. Auswahl des besten Attributes

Der erste Schritt, der auf einem Knoten ausgeführt wird, ist die Auswahl des besten Attributes in dem aktuellen Knoten. Die Auswahl des besten Attributes ist nötig für das Wachstum des Entscheidungsbaumes, da auf dem Attribut der Datensatz geteilt wird und so neue Knoten entstehen. Das Teilen auf einem Attribut ist zu einem durch das Format der Datensätze und zum anderen durch die Eigenschaften der TDIDT-Bäume (vgl. Kapitel 2.2.1) bedingt. Durch die Auswahl eines Attributes kann der Datensatz nach den Häufigkeiten der Attributwerte in den Instanzen und deren Auswirkung auf die Klassenverteilung der Instanzen geteilt werden.

Die Split-Metrik, mit der das beste Attribut identifiziert wird, unterscheidet sich je nach Algorithmus. Somit kann sich das beste Attribut eines Knotens je nach Algorithmus unterscheiden. Dies hat jedoch keinerlei Auswirkung auf den allgemeinen Ablauf des Entscheidungsbaumes, da dies lediglich die individuelle Struktur eines konkreten Beispiels ändert. Im Allgemeinen lässt sich sagen, dass je besser der Wert der Split-Metrik ist, desto reiner sind die Kinder die erstellt werden und die Aussagekraft des ausgewählten Attributes ist folglich hoch. Ist jedoch der Wert niedrig, so ist es fraglich wie informativ der Split ist [BFOS84].

Um Pruning auf diesem Schritt auszuführen, sind zwei Ansatzpunkte denkbar:

- **Stoppen des Algorithmus:**

Die Split-Metrik des betrachteten Splits wird überprüft und aufgrund dessen wird entschieden, ob der Split durchgeführt wird. Dies zieht eine Terminierung des Algorithmus auf dem Knotens nach sich, da somit keine Kindknoten erstellt werden können.

- **Ausschließen von Attributen:**

Aufgrund einer zusätzlichen Metrik wird die Auswahl von bestimmten Attributen verhindert. Wird das ausgewählte beste Attribut in einem Knoten als zu schlecht angesehen, so terminiert der Knoten.

Für beide Ansätze stehen nur begrenzt viele Informationen in diesem Schritt zur Verfügung. Diese sind der Datensatz des Knoten und damit zusammenhängende Informationen, sowie die Split-Metrik des Algorithmus. Pre-Pruning-Methoden, die an diesem Schritt ansetzen, müssen sich diese Informationen zu nutze machen, um über das Stoppen des Wachstums an Knoten oder das Weiterwachsen von Knoten zu entscheiden.

4.1.2. Durchführung des Splits

Der zweite Schritt ist die Durchführung des Splits. Dieser ist nötig, um weitere Knoten für den Entscheidungsbaum zu erzeugen. Aufgrund der Top-Down-Erstellung der TDIDT-Bäume sind die neu erstellten Knoten die Kinder des Knotens, auf dem der Split durchgeführt wird. Der Split kann erst nach Schritt S1 durchgeführt werden, da für die Durchführung des Splits das beste Attribut nötig ist. Mithilfe dessen wird der Datensatz in mehrere Teildatensätze aufgeteilt, die durch Knoten repräsentiert werden. Diese Aufteilung erfolgt abhängig von dem jeweiligen Algorithmus und es werden dabei mindestens zwei Kindknoten erstellt.

In diesem Schritt werden keine Entscheidungen getroffen, er ist lediglich eine Folge aus Schritt S1. Die Aufteilung der Kindknoten erfolgt aus dem in Schritt S1 ausgewählten Attribut, an dieser Auswahl wird nichts geändert. Aufgrund dessen ist dieser Schritt für das Pre-Pruning nicht von Interesse. Schritte, die für das Ansetzen von Pre-Pruning von Interesse sind, müssen Entscheidungen über das Wachstum des Baumes treffen können.

4.1.3. Überprüfung der Abbruchkriterien

Der dritte Schritt wird auf jedem der neu erstellten Knoten aus Schritt S2 ausgeführt. Die Knoten werden auf Abbruchbedingungen überprüft, um das Wachstum des Baumes, wenn nötig, zu stoppen. Abbruchkriterien sind spezifisch für jeden Algorithmus. Ein Kriterium muss jedoch von jedem Algorithmus abgedeckt werden. Dieses ist die Terminierung des Wachstums auf einem Knoten, sobald dieser rein ist. Tritt dieser Fall ein, so kann keine neue Information für die Klassifikation gewonnen werden. Tritt kein Abbruchkriterium ein, so beginnt der Ablauf bei Schritt S1 von neuem.

Dieser Schritt ist für das Ansetzen von Pre-Pruning-Methoden interessant, da hier von Grund auf von den Entscheidungsbaumalgorithmen wachstumsstoppende Methoden eingesetzt werden. Es liegt somit nahe, dass in diesem Schritt das Wachstum der Entscheidungsbäume beschränkt werden kann. Das Besondere dieses Schrittes ist, dass die in Schritt S2 neu erstellten Knoten betrachtet werden. Diese Knoten haben einen wichtigen Einfluss auf den Entscheidungsbaum, da in diesem Schritt entschieden wird, ob der Algorithmus auf diesen Knoten terminiert oder auf ihnen das Wachstum des Baumes fortgeführt wird. Aus diesem Grund ist es an dieser Stelle vorteilhaft, Pre-Pruning-Methoden anzusetzen. Mögliche zu betrachtende Kriterien sind:

- **Strukturelle Veränderungen des Entscheidungsbaumes:**
Durch das Hinzufügen neuer Knoten verändert sich der Entscheidungsbaum. Diese Veränderung betrifft die Anzahl der Knoten, sowie die Baumtiefe und die Breite des Baumes. Aufgrund dieser Veränderungen kann eine Entscheidung für das Wachstum des Baumes getroffen werden.
- **Veränderungen der Knoten zu den Elternknoten:**
Durch das Hinzufügen der Kindknoten können Unterschiede zwischen diesen und deren Elternknoten betrachtet werden. So kann etwa der Informationsgehalt der Kind- und Elternknoten überprüft werden, um etwa Overfitting zu entdecken.

Das erste Kriterium verweist auf die globale Veränderung des Baumes durch das Hinzufügen neuer Knoten, wohingegen das zweite Kriterium lokale Veränderungen betrachtet. Aufgrund dieser können Pre-Pruning-Methoden Entscheidungen für das weitere Wachstum des Baumes treffen.

4.2. Annäherung der Qualität

Im vorigen Abschnitt 4.1 wurden Schritte der TDIDT-Algorithmen aufgezeigt, bei denen Pre-Pruning-Methoden angesetzt werden können. Es ist jedoch unklar, für welche Schwellenwerte die Pre-Pruning-Methoden Entscheidungsbäume mit einer hohen Genauigkeit erzeugen, da die Genau-

igkeit der Bäume erst nach deren Erstellung ableitbar ist. In diesem Abschnitt wird ein Verfahren vorgestellt, mit dem ein Zusammenhang zwischen der Genauigkeit der Entscheidungsbäume und der Schwellenwerte der Pre-Pruning-Methoden hergestellt werden kann.

Breiman et al. und Quinlan betrachteten in Zusammenhang mit den von ihnen entwickelten Entscheidungsbaumalgorithmen die Möglichkeit von Pre-Pruning, jedoch wurde dies für beide Algorithmen verworfen [BFOS84; Qui93]. Diese Entscheidung wurde mit Verweis auf die Schwierigkeit des Findens von Schwellenwerten getroffen, mit denen genaue Entscheidungsbäume erzielt werden können. Inzwischen wurden Methoden entworfen, mit deren Hilfe das Finden dieser Werte für Analysten vereinfacht werden kann. Einer dieser Methoden ist das *Quality-Driven Early Stopping*, das von Fritz et al. entwickelt wurde [FBS19]. Diese Methode wurde ursprünglich verwendet, um Werte für Early-Stopping-Methoden in Clustering-Algorithmen nach jeder Iteration des Algorithmus zu berechnen und in den Zusammenhang mit der Qualität des Algorithmus zu setzen. Ein solcher Zusammenhang kann in Abbildung 4.2 betrachtet werden. Dort wird der Zusammenhang zwischen clustering-spezifischen Werten und der erzielten Qualität eines Clusters darstellt. In Rot ist eine Regressionskurve zu sehen, mit der der Zusammenhang bewertet wird. Dieses Verfahren soll im Folgenden für Entscheidungsbaumalgorithmen adaptiert werden. Die Erwartung an diese Methode ist, dass Zusammenhänge zwischen Schwellenwerten für Pre-Pruning-Methoden, sowie der Genauigkeit der Entscheidungsbäume, gefunden werden.

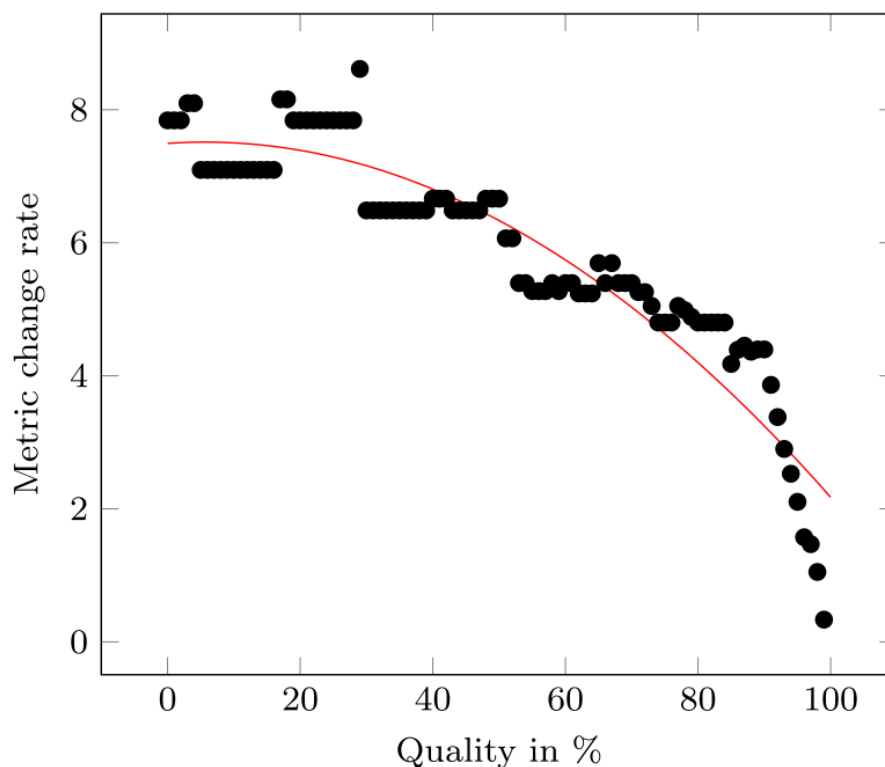


Abbildung 4.2.: Eine Regressionskurve, die den Zusammenhang zwischen clustering-spezifischen Werten und der erzielten Qualität darstellt [FBS19].

4. Konzept

Quality-Driven Early Stopping wird für jede Pre-Pruning-Methode angewendet, die einen Wert benötigt der von einem Analysten übergeben werden muss. Anhand des Konzeptes wird nach Ablauf aller drei Schritte aus Abschnitt 4.1 für jede dieser Pre-Pruning-Methoden der jeweilig Wert ausgegeben. Zusätzlich werden Qualitätsmetriken für den Entscheidungsbaum ausgegeben. Jede Pre-Pruning-Methode wird anhand ihrer Werte diesen Qualitätsmetriken gegenübergestellt. Diese Werte und die Qualität des Entscheidungsbaumes werden darauffolgend mit einer Regression ähnlich Abbildung 4.2 evaluiert und auf Zusammenhänge geprüft. Dies soll einem Analysten ermöglichen, eine Relation zwischen der zu erwarteten Qualität des Entscheidungsbaumes und den Werten der Pre-Pruning-Methode herzustellen und entsprechend zu verwenden.

5. Methoden für Pre-Pruning

Ein primärer Bestandteil dieser Arbeit stellen Pre-Pruning-Methoden dar. In diesem Kapitel werden die in dieser Arbeit verwendeten Pre-Pruning-Methoden vorgestellt, sowie auf das in Kapitel 4 vorgestellte Konzept angewendet. Die Zuordnung der Methoden auf die verschiedenen Schritte des allgemeinen Ablaufes der TDIDT-Entscheidungsbäume (vgl. Kapitel 4.1) ist äußerst relevant, da Pre-Pruning-Methoden in die Erstellung des Baumes eingreifen müssen. Dieser Ablauf ist in Abbildung 5.1 zu sehen. Die Pre-Pruning-Methoden, die in diesem Kapitel vorgestellt werden, sind alle mit diesem allgemeingültigen Ablauf vereinbar. Aufgrund des Konzeptes können diese Pre-Pruning-Methoden mit allen TDIDT-Algorithmen kombiniert werden.

Der Ablauf aus Kapitel 4.1 ist in Kombination mit möglichen Pre-Pruning-Methoden in Abbildung 5.1 dargestellt. Aufgrund des Aufbaus der Entscheidungsbäume kann Pre-Pruning nur an bestimmten Stellen angewandt werden, da es vom Ablauf des Aufbaus abhängt. Naheliegend sind die Auswahl des besten Attributes, sowie das Prüfen von Abbruchkriterien. Diese beiden Stellen des Ablaufes sind maßgeblich für die Erstellung des Baumes. Die Durchführung eines Splits ist für Pre-Pruning-Methoden nicht relevant. In diesem Schritt wird keine Entscheidung, die Auswirkungen auf den Baum hat, getroffen, sondern nur der im vorherigen Schritt vorbereitete Split ausgeführt. Während der Auswahl des besten Attributes und dem Prüfen von Abbruchkriterien können Pre-Pruning-Methoden nur mit im Baum vorhandenen Informationen und ableitbaren Metriken arbeiten.

Der Vorteil der Pre-Pruning-Methoden ist die mögliche hohe Laufzeiteinsparung. Dies geschieht durch die frühzeitige Begrenzung des Wachstums während der Erstellung des Entscheidungsbaumes. Lediglich das Horizon-Problem [Ber73] stellt eine Schwierigkeit der Pre-Pruning-Methoden dar. Aufgrund dessen kann nicht sichergestellt werden, dass im späteren Verlauf eines Teilbaumes, dessen Wachstum verhindert wird, wichtige Informationen für den Baum generiert werden. Es muss also die Qualität der Pre-Pruning-Methoden geprüft werden.

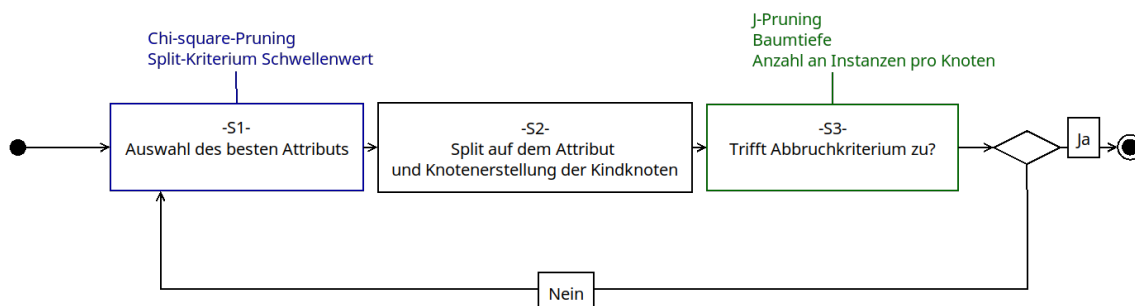


Abbildung 5.1.: Anwendung der Pre-Pruning-Methoden an den Ablauf eines TDIDT-Algorithmus

In diesem Kapitel werden fünf verschiedene Pre-Pruning-Methoden vorgestellt, die an der Auswahl des besten Attributes und dem Prüfen der Abbruchkriterien ansetzen. Diese werden an dem Entscheidungsbaum in Abbildung 5.2 erklärt. Die Zahlen in den Klammern kennzeichnen vier Metriken,

die in Pre-Pruning-Methoden verwendet werden. Der erste Wert ist der Wert der Split-Metrik des auf diesem Knoten ausgeführten Splits. Der zweite Wert ist der χ^2 -Wert des Split-Attributes des Knotens. Der dritte Wert kennzeichnet den Wert der J-Measure des Pfades bis zu diesem Knoten und der vierte Wert ist die Anzahl an Instanzen in dem Knoten.

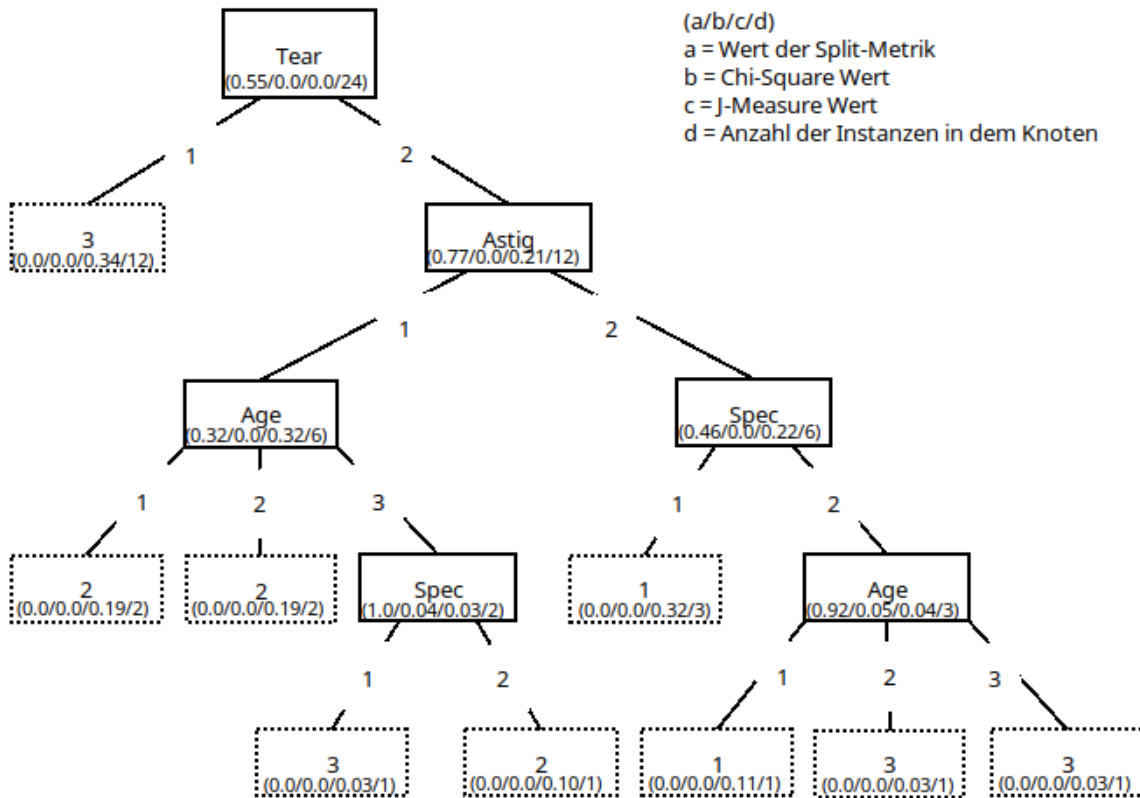


Abbildung 5.2.: Ein Entscheidungsbaum, der mit dem ID3-Algorithmus aus dem Lenses-Datenset¹ erzeugt wurde.

5.1. Split-Kriterium-Schwellenwert

Diese Methode setzt bei der Auswahl des besten Attributes in Schritt S1 an, wie in Abbildung 5.1 zu sehen. Dies ist der Fall, da das Verfahren auf dem Split-Kriterium basiert, welches nur in diesem Schritt verfügbar ist. Sie wurde von Breiman et al., sowie von Quinlan, in den jeweiligen Veröffentlichungen zu CART und C4.5 vorgeschlagen [BFOS84][Qui86]. Bei der Auswahl des besten Attributes wird für jedes mögliche Attribut des aktuellen Splits der Wert des Split-Kriterium berechnet. Das Attribut, das den besten Wert erzielt, wird als das beste Attribut ausgewählt. Interessant bei dieser Auswahl ist es, wenn der Wert des Split-Kriteriums des besten Attributes schlecht ist. Bei den Algorithmen ID3, C4.5 und CART bedeutet dies einen niedrigen Wert des Split-Kriteriums. Ein niedriger Split-Kriterium-Wert bedeutet einen geringen Informationsgewinn bezüglich der Klassifikation und es wird dadurch zweifelhaft, wie nützlich dieser Split ist [BFOS84]. Ein geringer

¹<http://archive.ics.uci.edu/ml/datasets/Lenses>

Informationsgewinn kann Overfitting andeuten, da dies sehr spezifische Anpassungen des Baumes auf einen Datensatz sind und deswegen keinen hohen Informationswert besitzen. Die vorgeschlagene Methode legt einen Schwellenwert β fest. Ein Knoten t , auf dem ein Split s ausgeführt wird, wird zu einem Blattknoten, wenn der beste Wert des Split-Kriterium x kleiner als der des Schwellenwertes ist:

$$\max_s x(s, t) < \beta \quad (5.1)$$

Die Schwierigkeit dieser Pre-Pruning-Methode ist das Finden eines Schwellenwertes. Ein zu niedriger Schwellenwert kann in Bäumen resultieren, die zu groß sind und zu sehr auf den Trainingsdatensatz angepasst sind. Bei einem zu hohen Schwellenwert können zu kleine Bäume erstellt werden, die eine hohe Ungenauigkeit bei der Klassifikation besitzen. Weiterhin besteht die Möglichkeit, dass der Split auf dem Knoten t aufgrund des niedrigen Wertes des Split-Kriteriums als schlecht erscheint, jedoch Splits auf darauf folgenden Knoten durchaus gut ausfallen können. Der Schwellenwert muss in dieser Methode von einem Analysten an den Entscheidungsbaumalgorithmus übergeben werden.

Für ein Beispiel wird der Schwellenwert $\beta = 0.4$ gesetzt. Anhand der Abbildung 5.2 und dem gewählten Schwellenwert kann diese Pre-Pruning-Methode angewendet werden. Der Knoten *Age*, mit dem Elternknoten *Astig*, enthält den Split-Wert $a = 0.32$, der unter den Schwellenwert fällt. Die darauffolgenden Knoten des Teilbaumes T_{Age} werden abgeschnitten und ein Blattknoten an der Stelle von *Age* erstellt. Diesem Blattknoten wird aufgrund des Mehrheitskriteriums die Klasse ≥ 2 zugeteilt.

5.2. χ^2 -Pruning

Das χ^2 -Pruning basiert auf Pearson's χ^2 -Test [Pea00] und wurde von Quinlan als Pre-Pruning-Methode für Entscheidungsbäume vorgeschlagen [Qui86]. Der χ^2 -Test soll im Bereich des Pre-Pruning dazu dienen ein mögliches Attribut A für einen Split s zu verhindern, falls die Werte $\{a_1, a_2, \dots, a_n\} \in A$ irrelevant für die Klassenverteilung in den durch A erstellten Kindknoten erscheinen. Die Wahrscheinlichkeit, dass ein Attribut A irrelevant zu einer Klassenverteilung einer Klasse c_h , mit $h \in \{1, 2, \dots, k\}$ in einem Kindknoten t_i , mit $i \in \{1, 2, \dots, v_A\}$ ist, wird wie folgt berechnet:

$$|c'_h|_i = |c_h| * \frac{|t_i|}{|t|} \quad (5.2)$$

$|c_h|$ steht hierbei für das Vorkommen der Klasse c_h im gesamten Trainingsdatensatz und $|c'_h|_i$ für das vermutete Vorkommen der Klasse in dem Kindknoten i . Für alle Klassen und alle Kindknoten ergibt sich ungefähr der χ^2 -Test mit $v_A - 1$ Freiheitsgraden [Qui86]:

$$y \approx \sum_{i=1}^{v_A} \sum_{h=1}^k \frac{(|c_h|_i - |c'_h|_i)^2}{|c'_h|_i} \quad (5.3)$$

Dabei steht $|c_h|_i$ für das reale Vorkommen der Klasse c_h in dem Kindknoten t_i . Der χ^2 -Test prüft also die Nullhypothese, dass A unabhängig von den Klassen der Instanzen in den Kindern von t ist. Kann dies nicht für das Attribut A mit einem hohen Signifikanzniveau abgewiesen werden, so wird auf A der Split s nicht ausgeführt. Quinlan schlägt ein Signifikanzniveau von 99% vor, mit dem Vermerk, dass dieser Wert gute Ergebnisse liefert [Qui86]. Der erzielte χ^2 -Wert y wird in einen

p-Wert umgerechnet, um mit dem Signifikanzniveau verglichen werden zu können. Der p-Wert ist dabei die Wahrscheinlichkeit $P(\chi^2 > y)$ für eine χ^2 -Verteilung mit den Freiheitsgraden $\nu_A - 1$. Um diese Pre-Pruning-Methode zu verwenden kann ein anderes Signifikanzniveau von einem Analysten angegeben werden. Das χ^2 -Pruning setzt an Schritt S1 aus Abbildung 5.1 an, da die Attribute für den Split überprüft werden.

In Abbildung 5.2 werden bei einem Signifikanzniveau von 3% (oder 0,03) zwei Teilbäume abgeschnitten. Der Knoten *Spec* mit dem Elternknoten *Age* und dem p-Wert $b = 0.04$ und der Knoten *Age* mit dem Elternknoten *Spec* und dem p-Wert $b = 0.05$ bestätigen die Nullhypothese und werden deswegen abgeschnitten und durch Blattknoten ersetzt.

5.3. J-Pruning

Das J-Pruning [Bra02] basiert auf der J-measure, die von Goodman und Smyth vorgeschlagen wurde, um den Informationsgehalt einer Regel zu berechnen [GS91]. Für das J-Pruning wird der Wert der J-Measure des Pfades vom Wurzelknoten bis zu dem zu prüfenden Knoten t_h berechnet. Dieser Wert wird mit dem Wert der J-Measure des Pfades vom Wurzelknoten bis zum Elternknoten t von t_h , verglichen. Aus diesem Grund setzt die Methode bei der Prüfung von Abbruchkriterien in Schritt S3 der Abbildung 5.1 an. Ab diesem Schritt sind die neuen Knoten erstellt und können für das J-Pruning verwendet werden. Die J-Measure wird wie folgt berechnet:

$$J(c_i; t) = p(t) * j(c_i; t) \quad (5.4)$$

Hierbei beschreibt $p(t)$ die Wahrscheinlichkeit, dass der Pfad bis zu dem aktuellen Knoten gewählt wurde und $j(c_i; t)$ beschreibt die j-measure oder auch cross-entropy:

$$j(c_i; t) = p(c_i|t) * \log_2\left(\frac{p(c_i|t)}{p(c_i)}\right) + (1 - p(c_i|t)) * \log_2\left(\frac{1 - p(c_i|t)}{1 - p(c_i)}\right) \quad (5.5)$$

$p(c_i|t)$ beschreibt die Wahrscheinlichkeit, dass die Klasse c_i in dem Knoten t gewählt wird und $p(c_i)$ ist die Wahrscheinlichkeit der Klasse im gesamten Datensatz. Der Wert der J-Measure beschreibt dabei den Informationsgehalt der Regel in bits. Da mehrere Klassen c_i in einem Knoten vorkommen gilt für die Auswahl der $JMeasure(t)$ eines Knotens t :

$$JMeasure(t) = \max_{c_i}(J(c_i; t)) \quad (5.6)$$

Ist die $JMeasure(t)$ größer, als die des auf Abbruchbedingungen zu prüfenden Kindknotens t_i , so wird t_i zu einem Blattknoten. Dies geschieht aufgrund der Annahme, dass ein Abfall des Informationsgehalts des Pfades durch Overfitting hervorgerufen wird. Das J-Pruning benötigt im Gegensatz zu den anderen vorgestellten Pre-Pruning-Methoden keine Eingabe eines Analysten.

In Abbildung 5.2 ist beispielweise der Pfad *Tear*, *Astig*, *Spec*, *Age* von Interesse für das J-Pruning. Von Knoten *Spec* zu Knoten *Age* ist ein Abfallen der J-Measure von $c = 0.22$ zu $c = 0.04$ zu erkennen. Dies lässt auf eine Verschlechterung des Informationsgehaltes des Baumes schließen. Der Teilbaum T_{Age} wird gekürzt und durch ein Blatt mit der Klasse 3 ersetzt.

5.4. Anzahl an Instanzen pro Knoten

Diese Pre-Pruning-Methode basiert darauf, dass im späteren Verlauf der Erstellung des Entscheidungsbaumes Overfitting auftritt. Wenige Instanzen haben dort eine höhere Auswirkung auf die Splits, die durchgeführt werden, als zu Beginn der Erstellung des Baumes. Somit findet hier eine zu starke Anpassung auf den Trainingsdatensatz statt. Diese Methode setzt bei der Überprüfung der Abbruchkriterien in dem Ablauf der TDIDT-Entscheidungsbäumen an. Diese Pre-Pruning-Methode folgt dieser Überlegung und legt einen Schwellenwert $Instanzen_{min}$ für die minimale Anzahl an Instanzen in einem Knoten t fest:

$$Instanzen(t) < Instanzen_{min} \quad (5.7)$$

Fällt die Anzahl an Instanzen in t , sodass die Gleichung 5.7 eintritt, so wird ein Blattknoten t erstellt. Die Klasse für t wird über das Mehrheitskriterium ausgewählt. Diese Methode setzt bei neu erstellten Knoten an, also bei der Überprüfung der Abbruchbedingungen. Der Schwellenwert $Instanzen_{min}$ muss von einem Analysten bereitgestellt werden.

Als Beispiel dient Abbildung 5.2. Der Schwellenwert wird auf $Instanzen_{min} = 3$ gesetzt. Der Teilbaum T_{Spec} mit $d = 2$ Instanzen fällt unter den Schwellenwert und wird aufgrund dessen gekürzt und durch einen Blattknoten mit der Klasse 2 oder 3 ersetzt.

5.5. Baumtiefe

Das Pruning per Baumtiefe ist eine sehr unkomplizierte Pre-Pruning-Methode. Overfitting in Entscheidungsbäumen tritt erst im späteren Verlauf des Lernvorgangs auf, da dort einzelne Instanzen mehr Auswirkung auf die Verzweigungen des Baumes haben. Folglich liegt es nahe den Baum ab einer bestimmte Tiefe zu kürzen, unter der Annahme, dass bis zu dieser Tiefe bereits ausreichend Informationen in das Modell eingeflossen sind. Diese Pre-Pruning-Methode setzt an der Überprüfung der Abbruchkriterien an. Bei einem neu erstellten Knoten t wird überprüft ob dessen Tiefe d_t die maximale Tiefe d_{max} erreicht, die von der Pre-Pruning-Methode übergeben wird.

$$d_t \geq d_{max} \quad (5.8)$$

Gilt für den Knoten t , dass Gleichung 5.8 eintritt, so wird t zu einem Blattknoten. Die Klasse für das Blatt t wird über das Mehrheitskriterium ausgewählt. Die maximale Baumtiefe muss von einem Analysten übergeben werden.

Diese Pre-Pruning-Methode wird anhand des Beispiels in Abbildung 5.2 erklärt. Ist die maximale Tiefe $d_{max} = 2$, so werden die Teilbäume T_{Age} und T_{Spec} auf Tiefe 2 abgeschnitten, wenn die Tiefe des Baumes bei dem Wurzelknoten mit 0 anfängt. Aus dem Knoten T_{Age} wird ein Blattknoten mit der Klasse 2 und aus dem Knoten T_{Spec} wird ein Blattknoten mit der Klasse 1 erstellt.

6. Evaluation

Dieses Kapitel behandelt die Evaluation des in Kapitel 4 vorgeschlagenen Konzepts. Hierbei werden die in Kapitel 5 vorgeschlagenen Pre-Pruning-Methoden auf den Entscheidungsbaumalgorithmen ID3, C4.5 und CART ausgewertet. Das Ziel der Evaluation ist es zu einem die Pre-Pruning-Methoden zu evaluieren und einen Zusammenhang der Qualität zu den Schwellenwerten der Pre-Pruning-Methoden zu finden. Anhand dieses Zusammenhanges soll es einem Analysten erlaubt werden für eine gewünschte Qualität eines Entscheidungsbaumes einen Schwellenwert für die jeweilige Pre-Pruning-Methode zu erhalten. Ein weiteres Ziel der Evaluation ist es, Pre-Pruning-Methoden mit bereits existierenden Post-Pruning-Methoden zu vergleichen.

In Kapitel 6.1 wird der Versuchsaufbau der Evaluation beschrieben. In Kapitel 6.2 wird die Qualität und in Kapitel 6.3 die Laufzeit der Pre-Pruning-Methoden evaluiert. Abschließend werden in Kapitel 6.4 die Pre-Pruning- mit bereits existierenden Post-Pruning-Methoden verglichen.

6.1. Versuchsaufbau

In diesem Kapitel wird der Aufbau und die Durchführung der Experimente erklärt. Zuerst wird in Abschnitt 6.1.1 die Implementierungsdetails der Entscheidungsbaumalgorithmen und Pruning-Methoden erläutert. In Abschnitt 6.1.2 wird anschließend auf die Auswahl der Datensätze eingegangen. Abschließend wird in Abschnitt 6.1.3 die verwendete Software und in Abschnitt 6.1.4 die verwendete Hardware genannt, die für die Ausführung der Experimente und der Entscheidungsbaumalgorithmen verwendet wurde.

6.1.1. Implementierung

Dieser Abschnitt behandelt die Implementierung des Konzeptes, sowie Anpassungen an die Algorithmen in der realen Implementierung. Die Entscheidungsbaumalgorithmen ID3, C4.5 und CART wurden basierend auf deren erster Publikation implementiert [BFOS84; Qui86; Qui93]. Die Entscheidungsbaumalgorithmen wurden nach dem in Kapitel 4 vorgestellten Konzept implementiert. Für den CART-Algorithmus existieren mehrere Split-Kriterien (vgl. Kapitel 2.2). Für diese Implementierung wurde das Gini-Kriterium implementiert, da dieses auch von den Verfassern des Algorithmus bevorzugt wird [BFOS84].

Die Algorithmen wurden in einigen Aspekten angepasst, die nicht als relevant für die Arbeit angesehen werden, um die Evaluation zu erleichtern. Der erste ist die Unterstützung von fehlenden Werten in Datensätzen, wie in Kapitel 2.2.2 beschrieben. Für diese Arbeit wurde in keinem Algorithmus die Möglichkeit implementiert, dass Datensätze mit fehlenden Werten unterstützt werden können, da diese Methoden nicht Teil der Arbeit sind. Die Instanzen mit fehlenden Werten wurden in einem Vorverarbeitungsschritt entfernt.

Der zweite Aspekt ist die Diskretisierung von kontinuierlichen Attributen, bevor der Split-Wert eines Attributes berechnet wird. Dies wurde aus dem Grund vorgenommen, da die Berechnung des Split-Wertes eines kontinuierlichen Attributes eine hohe Laufzeit hat. Um eine schnellere Ausführungszeit zu gewährleisten, werden die kontinuierlichen Attribute diskretisiert, bevor ihr Split-Wert berechnet wird. Dies geschah mit einem *Binning*-Verfahren. Unter einem Binning-Verfahren wird eine Diskretisierung von kontinuierlichen Werten durch deren Einteilung in Eimer (= Bins) verstanden. Ein Eimer bildet eine Zuordnungseinheit, in dem Fall des Binnings ist dies ein Wertintervall. Die Eimer, die gebildet werden, decken die selbe Reichweite an Werten ab. Ein Eimer bin_i mit $i \in 0, \dots, n - 1$ deckt dabei das Intervall $[min_i, max_i)$ ab. Dabei gilt $min_i = i * width$ und $max_i = (i + 1) * width$, mit $width$:

$$width = \frac{minWert + maxWert}{n} \quad (6.1)$$

$maxWert$ ist dabei der größte Wert des Attributes, $minWert$ der kleinste Wert des Attributes und n bezeichnet die Anzahl an gewünschten Eimern, die gebildet werden sollen. Die Anzahl der Eimer n , die für die Evaluation verwendet wird, beträgt $n = 100$. Das Binning-Verfahren hat keinerlei Auswirkung auf die Evaluation, da es sowohl bei Pre-Pruning, Post-Pruning, als auch bei Bäumen ohne Pruning angewendet wird.

Zuletzt ist die Implementierung der Post-Pruning-Methoden zu erwähnen, die implementiert wurden, um einen Vergleich mit den Pre-Pruning-Methoden zu ermöglichen. Implementiert wurden die beiden in Kapitel 2.3 vorgestellten Post-Pruning-Methoden EBP und REP. Diese wurden anhand ihrer Eigenschaft des Over-/Underpruning ausgewählt, da dies eine höhere Vergleichbarkeit der Algorithmen hinsichtlich ihrer Qualität anbietet. EBP ist tendenziell dem Underpruning zuzuordnen, wohingegen REP eher dem Overpruning angehört [EMS97].

Zu EBP sind zwei weitere Details für die Implementierung zu nennen. Der Schwerpunkt der Arbeit liegt nicht auf den Post-Pruning-Methoden, deswegen wurde nur eine einfache Implementierung von EBP gewählt. Diese erlaubt nicht das Ersetzen von Teilbäumen durch einen Ast des Teilbaumes, wie in Kapitel 2.3.2 beschrieben, lediglich das Ersetzen von Teilbäumen durch Blattknoten wird unterstützt. Weiterhin wurde für die Berechnung der Fehlerwahrscheinlichkeit der originale Code von Quinlan verwendet. Dieser ist aus dem Release 8¹ des C4.5 von Quinlan entnommen.

Zu REP ist zu erwähnen, dass dieser Algorithmus einen weiteren Datensatz benötigt, um die Klassifikationsfehler zu bestimmen. Dieser Datensatz muss ein anderer, als der für den Entscheidungsbaum verwendete Trainingsdatensatz, sein. Im Falle dieser Evaluation wurde hierfür der jeweilige zum Trainingsdatensatz zugehörige Testdatensatz verwendet.

6.1.2. Verwendete Datensätze

Für die Evaluation wurden Echtweltdatensätze verwendet. Breiman et al. vertreten die Meinung, dass simulierte Daten nicht die Komplexität von Echtweltdaten erreichen können [BFOS84]. Aufgrund dessen wurden für diese Arbeit keine synthetischen Daten verwendet. Auch Quinlan verwendet in seinen Veröffentlichungen zu ID3 und C4.5 in den meisten Fällen Echtweltdatensätze [Qui93].

¹<https://www.rulequest.com/Personal/>

Tabelle 6.1. Datensätze, die für den ID3-Algorithmus verwendet wurden.

Datensatz	Car Evaluation	Chess (King-Rook vs. King-Pawn)	Connect-4	Mushroom	Nursery	Soybean (Large)
Abkürzung	Car	Chess	Connect	Mushroom	Nursery	Soybean
Datentyp	Kategorisch	Kategorisch	Kategorisch	Kategorisch	Kategorisch	Kategorisch
Instanzen	1728	3196	67557	8124	12960	683
Trainingsinstanzen	1296	2396	50667	4233	9719	266
Testinstanzen	432	800	16890	1411	3241	296
Attribute	6	36	42	22	8	35
Klassen	4	2	3	3	5	19
Fehlende Werte	Nein	Nein	Nein	Ja	Nein	Ja
Verwendung Regression	Training	Test	Training	Test	Training	Test

Tabelle 6.2. Datensätze, die für den C45- und CART-Algorithmus verwendet wurden.

Datensatz	Avila	Census Income	Gas sensors for home activity monitoring	Poker Hand	Covertime	Skin Segmentation
Abkürzung	Avila	Census	Gas	Poker	Covertime	Skin
Datentyp	Real	Kategorisch, Integer	Real	Kategorisch, Integer	Kategorisch, Integer	Real
Instanzen	20867	45222	928991	1025010	581012	245057
Trainingsinstanzen	10430	30162	696743	25010	435759	183792
Testinstanzen	10437	15060	232248	1000000	145253	61265
Attribute	10	14	11	11	54	4
Klassen	12	2	3	10	7	2
Fehlende Werte	Nein	Ja	Nein	Nein	Nein	Nein
Verwendung Regression	Test	Training	Training	Test	Test	Training

Es wurden zwölf verschiedene Datensätze für die Evaluation ausgewählt. Sechs dieser Datensätze sind rein kategorisch und werden für den ID3 Algorithmus verwendet, da dieser Algorithmus nur kategorische Attribute unterstützt. Diese sind in Tabelle 6.1 zu sehen. Die restlichen sechs Datensätze werden für C4.5 und CART verwendet. Je drei der jeweils sechs Datensätze wurden verwendet, um Entscheidungsbäume zu erstellen, die für das Training einer Regression in Kapitel 6.2.2 verwendet wurden. Die jeweiligen restlichen drei Datensätze wurden verwendet, um Entscheidungsbäume zu erstellen, mit denen diese Regression trainiert werden kann. Diese Datensätze sind in Tabelle 6.2 zu sehen. Die Art der Attribute eines Datensatzes hat keine Auswirkung auf die Evaluation der Pre-Pruning-Methoden, da die im Rahmen der Arbeit vorgestellte Verfahren nicht in der Art des Attributes unterscheiden. Aufgrund dessen war die Art des Attributes für die Auswahl der Datensätze irrelevant, außer für die Auswahl der Datensätze des ID3, da dieser nur kategorische Attribute unterstützt.

Bei der Auswahl der Datensätze wurde Wert darauf gelegt, dass diese eine große Breite hinsichtlich der Anzahl der Instanzen, Attribute und Klassen abdecken. Dies erlaubt es mögliche Abhängigkeiten von Pre-Pruning-Methoden hinsichtlich der Datensätze zu analysieren. Ferner wurde jeder Datensatz in einen Test- und einen Trainingsdatensatz aufgeteilt. Einige der Datensätze wurden bereits in der Aufteilung nach Test- und Trainingsdatensatz zur Verfügung gestellt. Datensätze, die keine Aufteilung im Vorfeld haben, werden in Test- und Trainingsdatensätze unterteilt. Diese Unterteilung geschah in einer randomisierten Aufteilung, bei der 75% der Daten des Originaldatensatzes dem Trainingsdatensatz und 25% dem Testdatensatz zugeteilt wurden. Weiterhin enthalten die Datensätze Mushroom, Soybean und Census Instanzen mit fehlenden Werten. Diese Instanzen wurden aus den Datensätzen gelöscht, um eine einfachere Implementierung zu ermöglichen. Alle verwendeten Datensätze sind aus dem UCI Machine Learning Repository² abgerufen wurden (Stand 20.04.2019).

6.1.3. Software

Für die Implementierung in Java Version 1.8 wurde die Bibliothek Apache Commons math3 (Version 3.6.1)³ für die Verwendung diverser mathematischer Funktionen genutzt. Für die Erzeugung der Regression in Kapitel 6.2.2 wurde die Programmiersprache Python (Version 3.5) mit der Bibliothek auto-sklearn (Version 0.5.0)⁴ verwendet.

6.1.4. Hardware

Alle Experimente wurden auf einem IBM PureFlex Cluster, das durch OpenStack verwaltet wird, ausgeführt. Das Cluster hat 11 Berechnungsknoten mit je 256 GB RAM. 7 der Knoten nutzen 24 Intel Xeon E5-2630 CPUs mit je 2.30 GHz. Die restlichen 4 Knoten laufen auf 48 Intel Xeon E5-2680 v3 CPUs mit je 2.50 GHz. Die Experimente wurden auf einer Virtuellen Maschine (VM) auf diesem Cluster ausgeführt. Diese VM nutzt Ubuntu 18.04.01 LTS mit 8 VCPUS, 16GB RAM und eine HDD mit 160GB Speicherplatz. Die Experimente wurden dreimal ausgeführt, um mögliche Abweichungen der Laufzeit abzufangen. Im Rahmen der Evaluation wird der Medianwert der drei Ausführungen als Laufzeit dargestellt.

6.2. Evaluation der Qualität

In diesem Kapitel wird die Qualität, die mit den Pre-Pruning-Methoden aus Kapitel 5 erzielt wird, evaluiert. Diese Methoden sind auf den Entscheidungsbäumen ID3, C4.5 und CART nach dem Konzept aus Kapitel 4 implementiert. Vier der fünf Pre-Pruning-Methoden aus Kapitel 5 werden mithilfe des Quality-Driven Early Stopping, welches in Kapitel 4.2 vorgestellt wurde, evaluiert. Dies ist der Fall, da diese Methoden mit Werten arbeiten, die ihnen von einem Analysten übergeben

²<https://archive.ics.uci.edu/ml/index.php>

³<https://commons.apache.org/proper/commons-math/>

⁴<https://automl.github.io/auto-sklearn/stable/>

werden müssen. Die fünfte Pre-Pruning-Methode wird lediglich hinsichtlich des mit ihr erstellten Entscheidungsbaumes mit der Qualität verglichen, da sie keine Schwellenwerte benötigt. Die Evaluation der Qualität für die vier Methoden erfolgt in drei Schritten:

- E1 Anhand des Quality-Driven Early Stopping sollen Werte für die Pre-Pruning-Methoden in Zusammenhang mit der Qualität des erstellten Entscheidungsbaumes gesetzt werden.
- E2 Kann ein Zusammenhang zwischen den Werten und der Qualität festgestellt werden, so werden die Methoden mithilfe einer Regression evaluiert.
- E3 Anhand der Regression können Vorhersagen über die gewünschte Qualität des Modelles getroffen werden. Diese werden in diesem Schritt evaluiert.

Für den ersten Schritt werden die Ergebnisse der Entscheidungsbaumalgorithmen von drei der sechs Datensätze ausgewertet. Dies erlaubt es die in Schritt 2 erstellte Regression mit den jeweiligen restlichen drei Datensätzen zu überprüfen.

6.2.1. Zusammenhang zwischen Pre-Pruning-Methoden und der Baumqualität

In diesem Abschnitt wird die Qualität der Entscheidungsbäume mit den Schwellenwerten der Pre-Pruning-Methoden in Zusammenhang gesetzt. Die zu betrachteten Methoden sind: Pruning mit einem Split-Kriterium-Schwellenwert, χ^2 -Pruning, Pruning über die Baumtiefe und Pruning über die Anzahl der Instanzen. Dazu wird nach jeder Iteration des Ablaufes eines Entscheidungsbaumes (vgl. Kapitel 4.1) die Werte der Pre-Pruning-Methoden ausgegeben, sowie die Qualität der erstellten Modelle zu diesem Zeitpunkt. Die Werte der Pre-Pruning-Methoden sind:

- **p-Wert:**
Der aus dem χ^2 -Wert gewonnene Wert, der mit dem Signifikanzniveau verglichen wird.
- **Tiefe:**
Die maximale Baumtiefe des aktuellen Baumes.
- **Relative Instanzen:**
Die relative Anzahl an Instanzen ist die Anzahl Instanzen in einem Knoten im Verhältnis zu der Gesamtanzahl der Instanzen im gesamten Baum. In der Evaluation wird die maximale Anzahl an Instanzen der Knoten gemessen, die zu dem Zeitpunkt der Messung keine Kindknoten haben. Dies schließt Blattknoten mit ein. Es wird nicht die minimale Anzahl gemessen, um ein Szenario zu vermeiden, in dem sehr früh in der Erstellung eines Entscheidungsbaumes ein Blattknoten mit wenigen Instanzen erstellt wird. Dies würde verhindern, dass Veränderungen in späteren Blattknoten gemessen werden.
- **Wert der Split-Metrik:**
Der Wert der jeweiligen Split-Metrik eines Algorithmus, der für den aktuellen Split verwendet wurde.

Diese vier Werte werden der Qualität gegenübergestellt. Für die Messung der Qualität wurde sowohl die Accuracy als auch die Macro F1-Measure verwendet. Es wurde die Macro F1-Measure anstatt der Micro F1-Measure gewählt, da diese alle Klassen gleichwertig betrachtet in ihrer Berechnung. Die Micro F1-Measure beachtet hingegen die Klassengleichverteilung. Dies ist für diese Arbeit jedoch nicht von Relevanz. Im Folgenden wird die Qualität anhand der Accuracy betrachtet.

Die Accuracy ist das Verhältnis der korrekt klassifizierten Instanzen zu der Anzahl aller getesteter Instanzen. Die weiteren Diagramme mit der Macro F1-Measure sind im Anhang zu finden. Anhand der Accuracy wurden die Zusammenhänge zwischen den Werten und der Qualität klarer, weswegen sie im Folgenden betrachtet werden. Dies wird für die drei Entscheidungsbaumalgorithmen und für je drei der Datensätze der Algorithmen in diesem Abschnitt evaluiert.

ID3

Zuerst wird der Entscheidungsbaumalgorithmus ID3 betrachtet. In Abbildung 6.1 ist die Gegenüberstellung der Pre-Pruning-Werte zur Qualität des ID3-Baumes zu sehen. Auf der y-Achse sind hierbei die Werte der jeweiligen Pre-Pruning-Methoden zu sehen und die x-Achse stellt die Qualität in Form der Accuracy dar. Die in der Abbildung zu sehenden Werte sind mit den Datensätzen *Car*, *Connect* und *Nursery* erstellt worden.

Abbildung 6.1a stellt das χ^2 -Pruning dar. Hierbei ist eine starke Abhängigkeit dieser Pruning-Methode von einem Datensatzes zu sehen. Für den Connect Datensatz ändert sich der Wert der Accuracy wenig, wohingegen der p-Wert des χ^2 -Pruning eine sehr starke Streuung erfährt. Der Car und Nursery Datensatz hingegen zeigen eine starke Streuung hinsichtlich der Accuracy, die Werte des χ^2 -Pruning bleiben jedoch meist niedrig. Diese Abbildung lässt darauf schließen, dass ein Großteil der Werte des χ^2 -Tests sehr gering sind. Dies deutet darauf hin, dass die für den Split gewählten Attribute laut dem χ^2 -Test eine hohe Relevanz hinsichtlich der Klassenverteilung in den daraus entstandenen Kindern besitzen. Ein Zusammenhang zwischen dem χ^2 -Pruning und der Accuracy lässt sich jedoch nicht erkennen. Dafür sind die erzielten Ergebnisse zu unregelmäßig.

Abbildung 6.1b stellt die Baumtiefe der Accuracy gegenüber. Bei allen drei Datensätzen lässt sich ein Muster erkennen. Interessant ist vor allem der Connect Datensatz. Dieser zeigt mit steigender Tiefe einen Anstieg in der Accuracy, jedoch ist ab der Tiefe 13 nahezu keine Veränderung in der Accuracy des Baumes zu erkennen. Auch die anderen beiden Datensätze zeigen eine Abhängigkeit der Baumtiefe zu der Accuracy des Baumes. An beiden Datensätzen ist zu sehen, dass mit einer steigenden Baumtiefe auch die Accuracy des Baumes ansteigt. Weiterhin anzumerken ist die anfänglich nur geringe Veränderung der Accuracy durch die Baumtiefe. Dies kann durch die anfängliche Unsicherheit hinsichtlich der Klasse eines Knotens erklärt werden, da das Klassenverhältnis in einem Knoten noch keinen Schluss über die Klasse eines Knotens zulässt.

In Abbildung 6.1c ist die relative Anzahl der Instanzen der Knoten im Vergleich mit der Accuracy des Baumes zu sehen. Ähnlich wie in den vorigen beiden Abbildung zeigt sich auch hier ein Unterschied zwischen dem Datensatz Connect und den beiden anderen Datensätzen. Der Car und der Nursery Datensatz zeigen wenig Abhängigkeit der relativen Anzahl der Instanzen hinsichtlich der Accuracy. Dies kann auf die geringe Anzahl an Instanzen im Gesamtdatensatz zurückgeführt werden. Aufgrund der höheren Anzahl an Instanzen in dem Datensatz ist bei Connect ein stärkerer Zusammenhang hinsichtlich der Accuracy zu erkennen. So ist ab 4% der Instanzen in einem Knoten eine Verbesserung der Accuracy des Baumes zu erkennen.

Zuletzt ist in Abbildung 6.1d der Information Gain als Split-Kriterium zu betrachten. Ähnlich wie bei dem χ^2 -Pruning aus Abbildung 6.1a ist kein Zusammenhang zwischen dem Split-Kriterium des ID3 und der Accuracy des erstellten Baummodelles zu finden. Die Werte des Split-Kriterium scheinen von dem Datensatz abzuhängen, auf dem trainiert wurde. So sind die Werte des Connect Datensatzes sehr gestreut bei einer gleichbleibenden Accuracy, wohingegen die anderen beiden Datensätze eine starke Streuung sowohl hinsichtlich der Accuracy als auch des Split-Kriteriums darstellen.

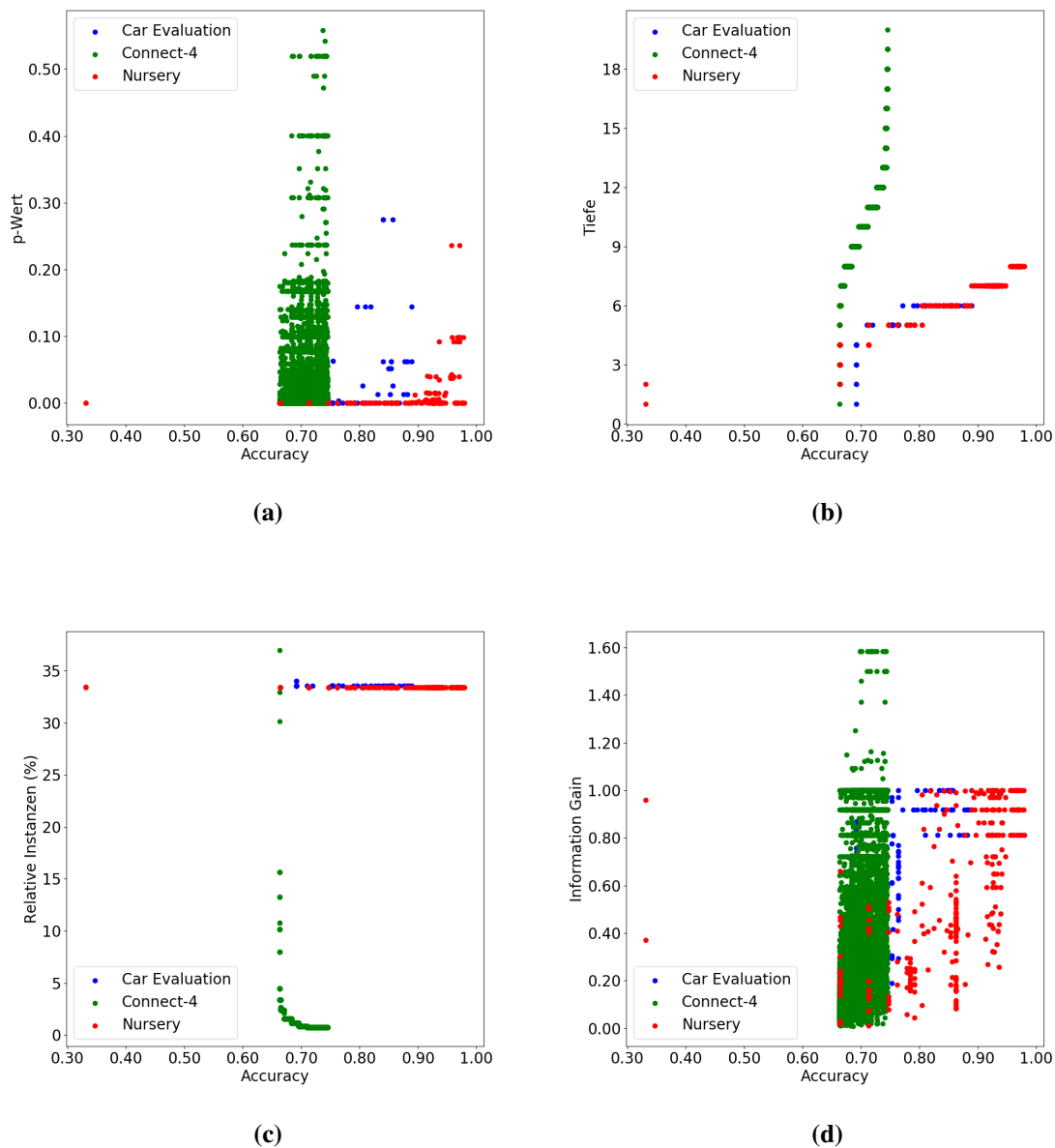


Abbildung 6.1.: Scatterplots, die den Wert der Pre-Pruning-Methoden der Accuracy eines ID3-Entscheidungsbaumes gegenüberstellen.

C4.5

Abbildung 6.2 stellt die Pre-Pruning-Methoden auf dem C4.5-Algorithmus dar. Generell bietet sich ein ähnliches Bild in Abbildung 6.2 zu der vorigen Abbildung 6.1. Die χ^2 - und Split-Pruning-Methoden zeigen auch für den C4.5 Algorithmus und den drei Datensätzen Census, Gas und Skin keine Relation zu der Accuracy der erstellten Entscheidungsbaume. Besonders deutlich ist dies in Abbildung 6.2d zu erkennen. Die Werte des Split-Kriteriums Information-Gain-Ratio lassen kein Muster erkennen, welches einen Schluss auf die Accuracy zulässt. Lediglich starke

6. Evaluation

Datensatzabhängigkeiten sind zu erkennen. So ist in Abbildung 6.2d zu sehen, dass die durch den Gas Datensatz erzeugten Messungen für das Split-Kriterium eine Streuung über das gesamte Spektrum der möglichen Werte der Accuracy und der Information Gain-Ratio darstellen. Für die beiden anderen Datensätze ist lediglich eine Streuung der Werte hinsichtlich des Split-Kriteriums zu erkennen, die Accuracy bleibt jedoch in einem festen Intervall.

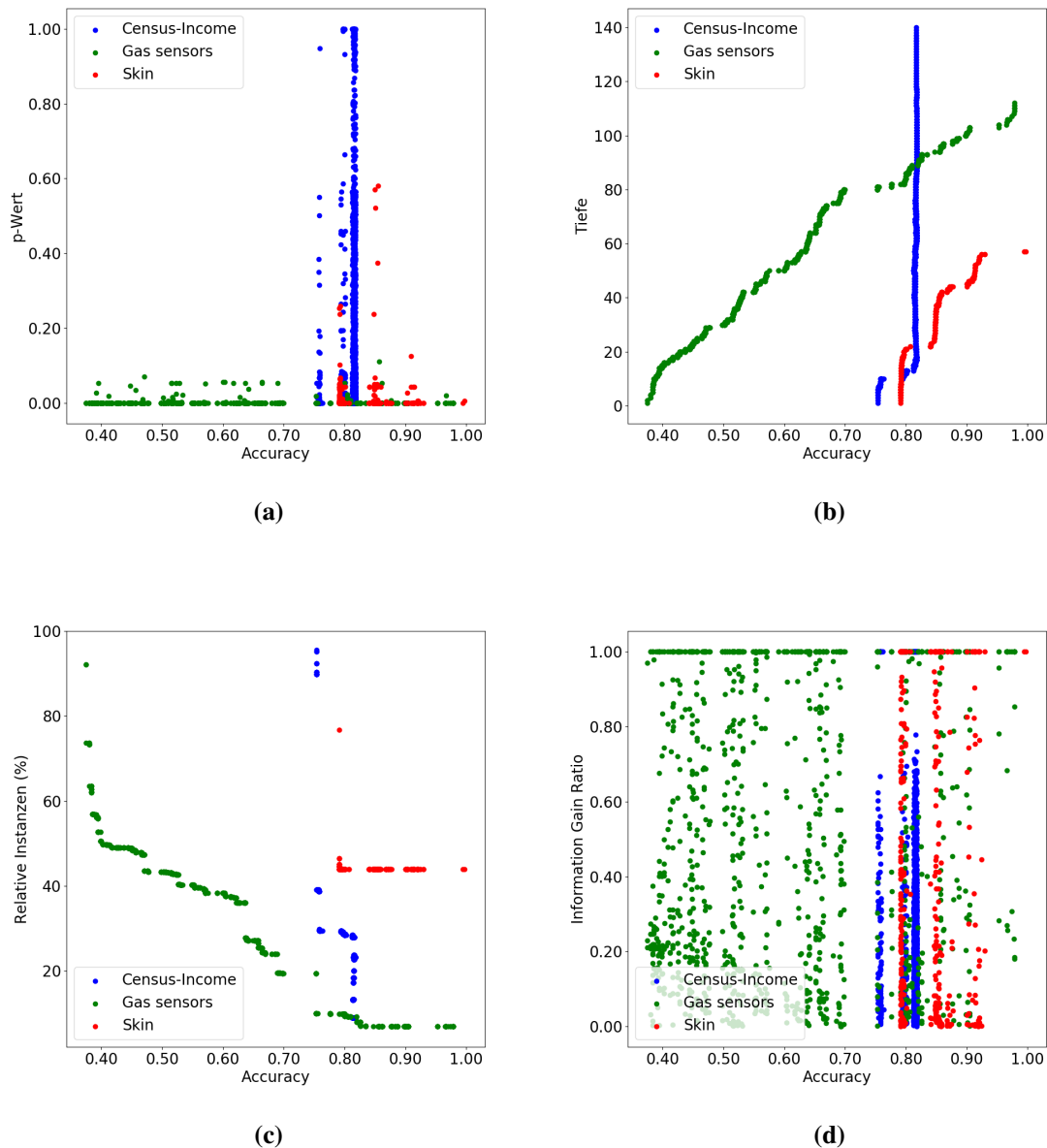


Abbildung 6.2.: Scatterplots, die den Wert der Pre-Pruning-Methoden der Accuracy eines C4.5-Entscheidungsbaumes gegenüberstellen.

Die beiden anderen Pre-Pruning-Methoden für die Baumtiefe und die Instanzen lassen, analog zum ID3-Algorithmus, einen Zusammenhang mit der Accuracy erkennen. Mit steigender Baumtiefe oder sinkender Anzahl an Instanzen lässt sich hier eine Verbesserung der Accuracy erkennen. Interessant

ist insbesondere der Census Datensatz für das Baumtiefe-Pruning, zu sehen in Abbildung 6.2b. Dieser lässt einen Knick in dem Verlauf der Baumtiefe erkennen. Erreicht der Entscheidungsbaum eine Tiefe von 13 Knoten, so ist keine nennenswerte Veränderung der Accuracy zu erkennen.

CART

Zuletzt werden die Messungen, die mit dem CART-Algorithmus erstellt wurden, betrachtet. Diese Messungen wurden auf den selben Datensätzen, wie die des C4.5, erstellt. Im Allgemeinen existieren keine maßgeblichen Unterschiede zu den vorigen beiden Messungen des ID3 und C4.5. Die χ^2 - und Split-Pruning-Methoden lassen auch bei diesen Messungen keinen Aufschluss über die Accuracy zu. Einige Unterschiede lassen sich hinsichtlich der Baumtiefe und der relative Anzahl an Instanzen erkennen. So ist in Abbildung 6.3c zu sehen, dass der Gas Datensatz ab 52% Accuracy keine Veränderung in der Anzahl an Instanzen aufweist. Eine Vermutung für diese Messung ist, dass bei dem Erreichen der Accuracy von 52% ein Blattknoten mit ca. 7% der Instanzen gebildet wurde. Ansonsten decken sich die Messungen des CART-Algorithmus mit denen des C4.5.

Abschließend lässt sich sagen, dass zwei der vier Pre-Pruning-Methoden in keinem der Entscheidungsbaumalgorithmen einen Zusammenhang zur Accuracy darstellen konnten. Eine der beiden Methoden ist das χ^2 -Pruning. Es konnte für diese Methode kein Zusammenhang der p-Werte zu der Accuracy des erstellten Modelles gefunden werden. Die p-Werte scheinen keinerlei Einfluss auf die Accuracy zu haben, die Verteilung der Werte erscheint zufällig. Weiterhin weist das χ^2 -Pruning eine Datensatzabhängigkeit auf. Der Connect Datensatz bei dem ID3-Algorithmus und der Census Datensatz bei CART und C4.5 weisen beide ein gänzlich anderes Verhalten als die restlichen Datensätze bei dem χ^2 -Pruning auf. Somit kann für diese Methode kein Wert für das Signifikanzniveau gefunden werden, der die Accuracy des Baumes zuverlässig verbessern kann. Die andere Methode ist das Pre-Pruning nach dem Split-Kriterium. Die Messungen dieser Methode weisen die selben Charakteristiken auf, wie die des χ^2 -Pruning. Starke Streuungen der Werte und die Datensatzabhängigkeit der Methode erlauben keinen Aufschluss über Schwellenwerte für diese Methode. Beide Pre-Pruning-Methoden erlauben keine genaue Vorhersage der Accuracy von Entscheidungsbäumen.

Umgekehrt verhält es sich bei dem Baumtiefen- und dem Instanzen-Pruning. Bei beiden Pre-Pruning-Methoden lassen sich Muster und ein Zusammenhang zur Accuracy des Entscheidungsbaumes erkennen. Insbesondere der Census Datensatz bei C4.5 und CART, sowie der Connect Datensatz bei ID3 lassen eine Anwendungsmöglichkeit des Baumtiefen-Prunings erkennen. Ab einer bestimmten Baumtiefe verändert sich die Accuracy des Baumes nahezu nicht. Beim CART-Algorithmus in Abbildung 6.3b ist zu bemerken, dass die Accuracy für eine bestimmte Tiefe kleiner wird. Ein Kürzen des Baumes ab einer Tiefe von 16 Knoten hätte das Wachstum frühzeitig bei einer gleichbleibenden Accuracy beendet. Ähnliches kann in Abbildung 6.2c für das Instanzen-Pruning erkannt werden. Ab einer relativen Anzahl an Instanzen von 30% in einem Knoten lassen sich keine Veränderungen der Accuracy des C4.5-Entscheidungsbaumes feststellen. Ein Schwellenwert von 30% der Instanzen des Datensatzes für diese Pre-Pruning-Methode hätte bei dem C4.5-Entscheidungsbaum auf dem Census Datensatz das Wachstum frühzeitig beenden können, ohne Verlust bei der Accuracy des Baumes.

6. Evaluation

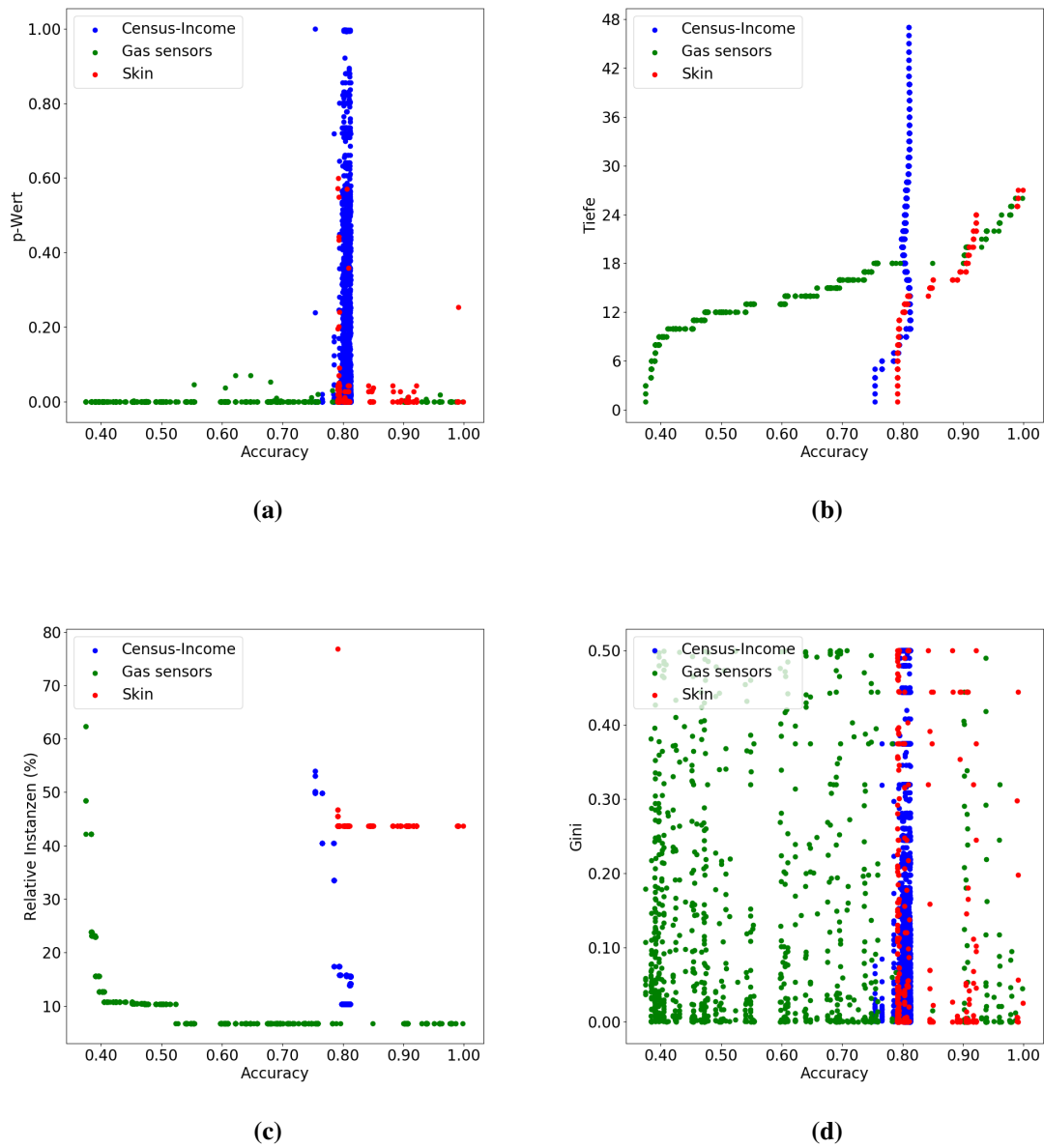


Abbildung 6.3.: Scatterplots, die den Wert der Pre-Pruning-Methoden der Accuracy eines CART-Entscheidungsbaumes gegenüberstellen.

Für die folgenden Abschnitte in diesem Kapitel 6.2 wird das χ^2 -Pruning, sowie das Split-Pruning verworfen, da keinerlei Anhaltspunkte für deren Auswirkung auf die Qualität der Entscheidungsbäume gefunden werden konnte. Das Baumtiefe- und das Instanzen-Pruning werden in diesem Kapitel hingegen genauer betrachtet, da ein Zusammenhang zwischen den Pruning-Methoden und der Qualität des erzeugten Modelles zu existieren scheint.

Tabelle 6.3. Die Regressionsmetriken der Regressionskurve. Die Werte sind auf drei Stellen nach dem Komma gerundet.

Metrik	ID3		C4.5		CART	
	Relative Instanzen	Baumtiefe	Relative Instanzen	Baumtiefe	Relative Instanzen	Baumtiefe
R^2	-0.516	0.478	-1.091	-0.143	-7.007	-0.572
Mean Squared Error	280.097	4.640	423.583	2660.225	93.285	52.092

Tabelle 6.4. Eine Tabelle mit Messwerten der Baummodelle ohne Pruning, die für das Training der Regressionskurve verwendet wurden.

	ID3			C4.5			CART		
	Car	Connect	Nursery	Census	Gas	Skin	Census	Gas	Skin
Maximale Tiefe	6	20	8	140	112	57	47	26	27
Min Relative Instanzen	33,565%	0,744%	33,378%	6,922%	6,849%	43,964%	10,384%	6,747%	43,691%
Max Relative Instanzen	34,028%	36,961%	33,429%	95,590%	92,174%	76,791%	54,018%	62,362%	76,791%
Änderungen der Relativen Anzahl an Instanzen	2	23	2	42	96	5	13	12	4

6.2.2. Evaluation der qualitativen Zusammenhänge

In diesem Abschnitt sollen mithilfe von Regressionskurven die Pre-Pruning-Methoden hinsichtlich ihres Zusammenhangs mit der Qualität eines Entscheidungsbaumes bewertet werden. Betrachtet werden die beiden Pre-Pruning-Methoden Baumtiefe- und Instanzen-Pruning, die in Kapitel 6.2.1 als relevant für die Qualität des Baumes befunden wurden. Für jeder der Pre-Pruning-Methoden wurde eine Regressionskurve für deren Schwellenwerte erstellt. Diese Regressionskurven wurden auf den Messungen von Baummodellen, die durch den ID3, C4.5 und CART erstellt wurden, trainiert. Die Eingabe der Regressionskurve ist die Accuracy des betrachteten Modelles und die Ausgabewerte sind die Schwellenwerte der zu betrachtenden Pre-Pruning-Methoden. Es ist für diese Arbeit nicht von Relevanz, welche Art von Regressionskurve verwendet wird, solange sie den Zusammenhang zwischen der Accuracy und der Pre-Pruning-Methode zeigt.

Die Regressionskurven wurden mit Hilfe von auto-sklearn, einer AutoML-Bibliothek, erstellt. Diese erstellt und evaluiert diverse Regressionsmethoden und wählt die nach der Evaluation beste Regressionskurve aus. Die Trainingszeit der Regression wurde als eine Stunde gewählt, da dies der Standardparameter der Bibliothek ist. Als Trainingsdatensätze für die Regressionskurven wurden die jeweiligen Messungen der Entscheidungsbäume, die auf den Datensätzen aus Kapitel 6.2.1 erstellt wurden, gewählt. Diese sind die Modelle, die auf den Datensätzen Car, Connect und Nursery für den ID3, sowie Census, Gas und Skin für C4.5 und CART erstellt wurden. Als Testdatensätze für die Regression wurden die Messungen der Entscheidungsbäume, die auf den jeweiligen übrigen Datensätzen erstellt wurden, gewählt. Dies soll es erlauben die Pre-Pruning-Methoden hinsichtlich der Accuracy eines Entscheidungsbaumes zu bewerten, indem die Qualität der Vorhersage über die Regression bewertet wird.

Für die Bewertung der Regression wurden diverse Metriken berechnet. Diese sind die R^2 -Metrik und der Mean Squared Error. Die Ergebnisse der Metriken sind für die jeweiligen Algorithmen und Pre-Pruning-Methoden in Tabelle 6.3 zu sehen. Im Folgenden Abschnitt werden diese Ergebnisse diskutiert.

Tabelle 6.5. Eine Tabelle mit Messwerten der Baummodelle ohne Pruning, die für das Testen der Regressionskurve verwendet wurden.

	ID3			C4.5			CART		
	Chess	Mushroom	Soybean	Avila	Poker	Covertime	Avila	Poker	Covertime
Maximale Tiefe	13	3	7	81	47	232	20	30	40
Min Relative Instanzen	23,038%	28,656%	6,015%	6,922%	0,120%	1,360%	6,922%	0,120%	1,227%
Max Relative Instanzen	81,302%	49,043%	54,511%	93,078%	92,123%	93,403%	93,078%	50,100%	56,957%
Änderungen der Relativen Anzahl an Instanzen	5	2	7	68	41	156	14	31	29

Im Allgemeinen sind die Ergebnisse eindeutig. Die Werte der R^2 -Metrik sind meist negativ. Negative R^2 -Werte bedeuten, dass die gewählte Regression nicht dem Trend der Daten folgt, die auf ihr getestet wurden. Die Regression kann folglich keine passenden Werte vorhersagen. Das selbe ist aus dem Mean Squared Error zu schließen. Die Werte sind in den meisten Fällen sehr hoch. Dies deutet auf einen hohen Fehler in der Vorhersage durch die Regression hin. Die einzige Ausnahme bildet das Baumtiefen-Pruning bei dem ID3-Algorithmus. Dieses hat einen R^2 -Wert von 0,478 und, verglichen zu den anderen Messungen, einen sehr geringen Mean-Squared-Error-Wert. Dies kann auf die Unterschiede in der Tiefe der Modelle, die durch die Trainingsdatensätze und Testdatensätze entstanden sind, zurückgeführt werden. In Tabelle 6.4 und Tabelle 6.5 ist zu sehen, dass sowohl die Test-, als auch Trainingsmodelle des ID3 von ähnlicher Tiefe sind. Bei dem CART-Algorithmus sind die Baumtiefen der Modelle in einem ähnlichen Bereich, wohingegen die des C4.5 sich sehr unterscheiden. Dies erklärt den großen Unterschied des Mean Squared Error der beiden Algorithmen. Die Regressionen für das Instanzen-Pruning liefern bei allen Algorithmen durchgehend schlechte Werte. An den Maximal und Minimal Instanzen Verhältnisse aus Tabelle 6.4 und Tabelle 6.5 ist dies nicht abzuleiten, da die Werte der Test- und Trainingsdaten sich ungefähr im gleichen Bereich bewegen. Vielmehr ist die Häufigkeit der Veränderung der Anzahl an Instanzen in den Knoten während der Erstellung des Entscheidungsbaumes für die Regression relevant. Je häufiger die Messungen der Anzahl an Instanzen sich ändern, desto höher ist die Anzahl an Datenpunkten auf denen die Regressionskurve trainiert werden kann. In den Tabellen 6.4 und 6.5 ist zu sehen, dass es große Unterschiede zwischen den Trainings- und Testdaten in diesem Punkt gibt. Dies erklärt die schlechten R^2 und Mean Squared Error Werte der Regression bei den Werten für das Instanzen-Pruning.

Im Allgemeinen lässt sich sagen, dass die Regressionskurven den in Kapitel 6.2.1 gefundenen Zusammenhang zwischen Pre-Pruning-Methoden und der Accuracy der Entscheidungsbäume, als schlecht bewerten. Dies liegt vornehmlich an den Messungen der erstellten Entscheidungsbäume, die sehr datensatzspezifisch ausfallen. Weiterhin sind die Regressionskurven für kleinere Werte unter ca. 35% Accuracy sehr unzuverlässig, da für diese Werte keine Trainingsdaten vorliegen. Dies liegt daran, dass der erste Split der Entscheidungsbäume zu einer höheren Accuracy als 35% führt.

6.2.3. Vorhersage der Qualität

In diesem Kapitel wird untersucht, ob mithilfe der Regression aus Kapitel 6.2.2, für eine gewünschte Accuracy, ein Schwellenwert für eine Pre-Pruning-Methode berechnet werden kann, mit dem diese Accuracy erreicht wird. Dabei folgt dieser Teil der Evaluation folgendem Szenario: Ein Analyst will

eine bestimmte Accuracy für einen Entscheidungsbaum erreichen. Um diese Accuracy zu erreichen, muss der jeweiligen Pre-Pruning-Methode ein Schwellenwert übergeben werden. Mit diesem Wert soll der Baum entsprechend gekürzt werden, damit die gewünschte Accuracy erreicht wird.

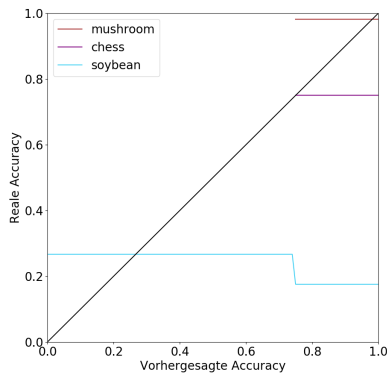
Im Folgenden wird dieses Szenario für Accuracy-Werte zwischen 0% und 100% Accuracy für das Baumtiefen- und Instanzen-Pruning betrachtet. Die Werte für die Pre-Pruning-Methoden wurden mithilfe der Regressionsmodelle aus Kapitel 6.2.2 berechnet. Diese Werte wurden mit den Testdatensätzen der Regressionsmodelle überprüft und der reale Accuracy-Wert wurde zurückgegeben. Sind die Mehrzahl der gewünschten Accuracy-Werte höher als die Werte der realen Accuracy, so wird die Tendenz der Vorhersage als optimistisch bezeichnet. Sind die Mehrzahl der gewünschten Accuracy-Werte hingegen kleiner oder gleich der realen Accuracy-Werte, so wird die Tendenz der Vorhersage als pessimistisch bezeichnet. Je pessimistischer die Vorhersage, desto höher ist folglich der reale Accuracy-Wert.

Die Ergebnisse dieses Experimentes sind für die drei Algorithmen ID3, C4.5 und CART, sowie für die beiden Pre-Pruning-Methoden Baumtiefen- und Instanzen-Pruning, in Abbildung 6.4 zu sehen. Die Abbildungen in einer Reihe gehören zu dem selben Algorithmus und die Abbildungen in einer Spalte sind der selben Pre-Pruning-Methode angehörig. Die x-Achse der jeweiligen Abbildungen ist die gewünschte Accuracy und die y-Achse ist die reale Accuracy. Die Accuracy wird für die x-Werte $\{0, 0.01, 0.02, \dots, 0.99, 1.0\}$ überprüft.

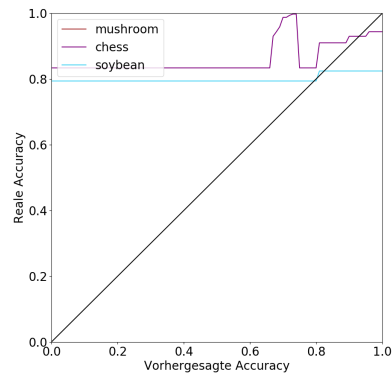
Zuerst werden die Ergebnisse des ID3-Algorithmus in Abbildung 6.4a und Abbildung 6.4b betrachtet. Die schwarze Linie in den Diagrammen beschreibt hierbei das optimale Ergebnis. Dieses tritt ein, wenn die gewünschte Accuracy durch das Pre-Pruning als realer Wert erreicht wird. Interessant in Abbildung 6.4a ist, dass sowohl der Chess als auch der Mushroom Datensatz bei ungefähr 70% gewünschter Accuracy beginnen Werte anzuzeigen. Dies kann darauf zurückgeführt werden, dass zu wenige Datenpunkte für das Erlernen der Regressionskurve zur Verfügung standen. Dadurch kann es passieren, dass die von der Regression zurückgegebene relative Anzahl an Instanzen geringer ist, als die des Testdatensatzes. Dies führt dazu, dass kein Accuracy-Wert in dem Testdatensatz gefunden werden konnte, der zu dem Rückgabewert der Regression passt. Im Allgemeinen verläuft die Vorhersage optimistisch. Der reale Wert des Entscheidungsbaumes ist für 101 Werte geringer als der gewünschte Wert und für 52 Werte größer oder gleich dem gewünschten Wert. Dies bedeutet, dass bei 52 Werten die Rückgabewerte der Regression eine schlechtere Accuracy besitzen, als gewünscht ist.

Unterschiedlich sieht es beim Baumtiefen-Pruning in Abbildung 6.4b aus. Der Großteil der realen Accuracy-Werte (176) sind größer oder gleich der gewünschten Accuracy-Werte und lediglich 26 Werte kleiner als die gewünschten Werte. Die Vorhersage ist folglich pessimistisch und die Entscheidungsbäume liefern meist bessere Werte als gewünscht. Interessant bei dieser Abbildung ist, dass die Werte des Mushroom Datensatzes nicht abgebildet sind. Dies ist der Fall, da der durch den Mushroom Datensatz erstellte Entscheidungsbaum eine sehr geringe Tiefe hat. Die Regression liefert jedoch höhere Werte, als die maximale Tiefe des Baumes, für das Baumtiefen-Pruning zurück. Folglich können hierzu keine Voraussagen hinsichtlich der Accuracy getroffen werden.

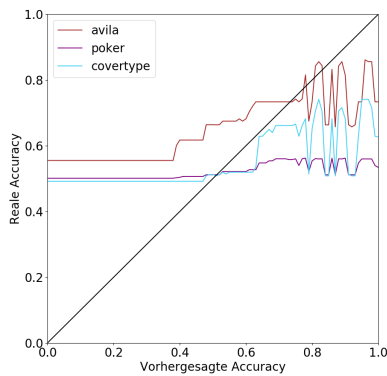
6. Evaluation



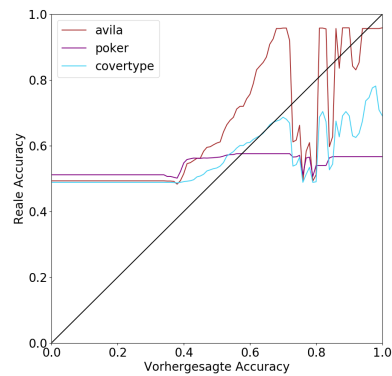
(a) Vorhersagen der Accuracy, mithilfe des Instanzen-Pruning für den ID3-Algorithmus



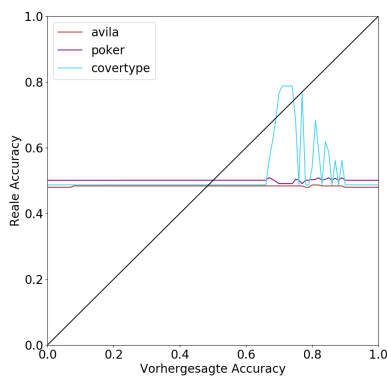
(b) Vorhersagen der Accuracy, mithilfe des Baumtiefen-Pruning für den ID3-Algorithmus



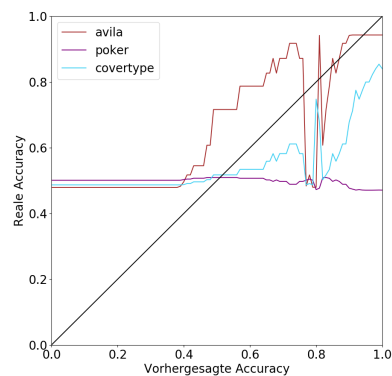
(c) Vorhersagen der Accuracy, mithilfe des Instanzen-Pruning für den C4.5-Algorithmus



(d) Vorhersagen der Accuracy, mithilfe des Baumtiefen-Pruning für den C4.5-Algorithmus



(e) Vorhersagen der Accuracy, mithilfe des Instanzen-Pruning für den CART-Algorithmus



(f) Vorhersagen der Accuracy, mithilfe des Baumtiefen-Pruning für den CART-Algorithmus

Abbildung 6.4.: Vorhersagen der Accuracy, die mithilfe der Regressionen aus Kapitel 6.2.2 getroffen wurden.

Als nächstes wird der C4.5-Algorithmus in Abbildung 6.4c und 6.4d betrachtet. Die Vorhersagen beider Pre-Pruning-Methoden fallen pessimistisch aus, die realen Accuracy-Werte sind folglich höher als die erwarteten. Jedoch sind bei beiden Abbildungen hohe Schwankungen ab dem Bereich größer 75% Accuracy zu sehen. Dies ist damit zu erklären, dass lediglich die Messungen des Entscheidungsbaumes des Gas Datensatzes im niedrigen Accuracy Wertebereich Datenpunkte enthalten und somit in dem Bereich unter 75% stark die Werte beeinflusst. Bei höheren Accuracy-Werten enthalten alle drei Messungen, die für das Training der Regression verwendet wurden, Datenpunkte. Da die Entscheidungsbäume, auf denen die Messungen erstellt wurden, unterschiedliche Baumtiefen und Instanzen Messungen enthalten (vgl. Tabelle 6.4), ist eine starke Schwankung in diesem Bereich der Accuracy zu erwarten.

Bei den Abbildungen 6.4e und 6.4f des CART-Algorithmus sieht dies ähnlich aus. Auch hier sind die Vorhersagen pessimistisch und im hinteren Wertebereich des Accuracy sind hohe Schwankungen zu erkennen. Dies kann auf die selben Ursachen zurückgeführt werden, die die Ergebnisse des C4.5-Algorithmus in Abbildung 6.4c und 6.4d beeinflusst haben.

Im Allgemeinen lässt sich sagen, dass die Vorhersagen der realen Accuracy anhand einer gewünschten Accuracy pessimistisch ausfallen. Die Schwellenwerte für die Pre-Pruning-Methoden, die durch die Regression berechnet wurden, haben meist eine besserer reale Accuracy zur Folge, als gewünscht war. Dies ist jedoch in den meisten Fällen nur für gewünschte Werte unter ca. 50% Accuracy der Fall. Bei gewünschten Werten über ca. 50% Accuracy sind die realen Werte oft geringer. Das ist die Folge der Regressionskurven, die schlecht auf die Werte der Testdatensätze angepasst sind (vgl. Tabelle 6.3).

6.2.4. Qualität des J-Pruning

Dieser Abschnitt behandelt die Qualität von Entscheidungsbäumen, die mit J-Pruning gekürzt wurden. J-Pruning wird getrennt von den anderen Pre-Pruning-Methoden betrachtet, da es keine Schwellenwerte benötigt. Die Accuracy der Entscheidungsbaumalgorithmen ID3, C4.5 und CART wird mit dieser in den Vergleich gesetzt. Die Accuracy der Entscheidungsbäume lässt in den meisten Fällen deutlich nach, wenn J-Pruning angewendet wird. Dies ist in Tabelle 6.6 zu sehen. Ausnahmen bilden der Census Datensatz in Tabelle 6.6, sowie der Poker Datensatz für den C4.5. Diese Steigungen in der Accuracy sind jedoch sehr gering und können als Ausnahme angesehen werden. In den meisten Fällen sinkt die Accuracy durch das Anwenden von J-Pruning. Bei einigen Entscheidungsbäumen des C4.5 und CART fällt die Accuracy um mehr als 30%, jedoch bleibt die Accuracy bei drei der sechs Datensätzen des C4.5 und CART in einem 10% Accuracy-Bereich um den Wert des ursprünglichen Entscheidungsbaumes. Die Accuracy der ID3-Entscheidungsbäume in Tabelle 6.6 lässt durch J-Pruning nicht stark nach, wie die bei einigen C4.5- und CART-Entscheidungsbäumen der Fall ist. Hier ist der höchste Verlust an Accuracy im Nursery Datensatz mit 19% zu verzeichnen.

Die sinkende Accuracy ist durch die große Anzahl an abgeschnittenen Knoten erklärt werden. So werden durch das J-Pruning deutlich kleinere Entscheidungsbäume erzeugt, als dies durch die originalen Entscheidungsbaumalgorithmen der Fall ist. Hervorzuheben ist etwa der Covertyp Datensatz in Tabelle 6.6. Dieser verliert bei sowohl C4.5 als auch CART mehrere zehntausende Knoten. Jedoch leidet die Accuracy der Entscheidungsbäume darunter, wie ebenso am Covertyp Datensatz zu sehen ist. Eine Ausnahme bildet der Connect Datensatz. Bei diesem werden über 30000 Knoten abgeschnitten und die Accuracy geht lediglich um 6% zurück.

Tabelle 6.6. Accuracy der Entscheidungsbäume mit und ohne J-Pruning

ID3-Datensätze	Car	Chess	Connect	Mushroom	Nursery	Soybean
Accuracy	93.981%	99.875%	74.506%	100%	100%	84.459%
Accuracy J-Pruning	76.157%	75.125%	68.620%	99.787%	81.950%	73.649%
Knoten	345	93	31750	22	1090	142
Knoten J-Pruning	33	5	28	14	21	60

C4.5-Datensätze	Avila	Census	Gas	Poker	Covertime	Skin
Accuracy	95.8705%	81.7065%	97.8682%	57.6088%	92.0821%	99.6997%
Accuracy J-Pruning	53.7415%	82.4050%	72.0958%	53.4639%	52.9676%	92.8974%
Knoten	1293	13931	2429	19371	62191	771
Knoten J-Pruning	49	79	103	65	23	47

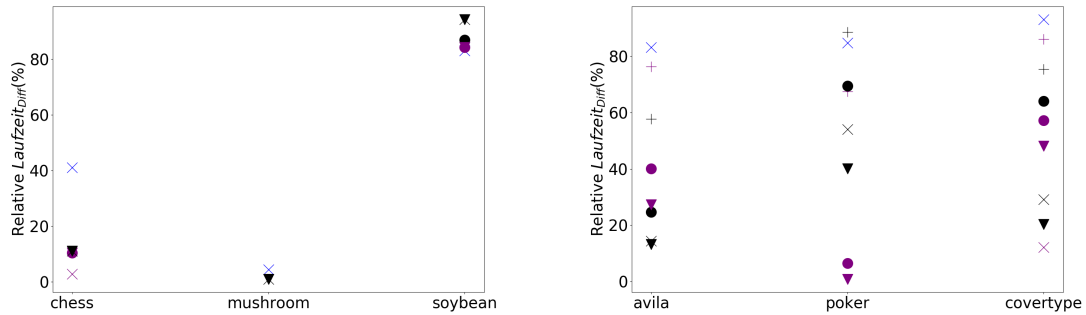
CART-Datensätze	Avila	Census	Gas	Poker	Covertime	Skin
Accuracy	96.0620%	81.0558%	99.7602%	47.1537%	93.2814%	99.8629%
Accuracy J-Pruning	50.4743%	82.4434%	62.6171%	49.8277%	67.5504%	93.6326%
Knoten	1241	8747	1761	8943	46927	585
Knoten J-Pruning	5	17	19	23	17	17

Im Allgemeinen lässt sich sagen, dass das J-Pruning sehr unzuverlässige Ergebnisse für die Accuracy der Entscheidungsbäume liefert, jedoch sehr kleine Bäume erstellt. Für den ID3-Algorithmus scheint es die besten Ergebnisse zu erzielen, was an der geringen Größe der ID3-Entscheidungsbäume gegenüber der C4.5- und CART-Entscheidungsbäume liegt. So besitzen bei C4.5 und CART die Entscheidungsbäume meist über tausend Knoten, wobei es bei dem ID3 deutlich weniger sind, wie in Tabelle 6.6 zu sehen. Durch die geringe Größe des ID3 kann nur wenig durch das J-Pruning abgeschnitten werden und der Verlust der Accuracy verbleibt im Vergleich zu den Ergebnissen des C4.5 und CART gering.

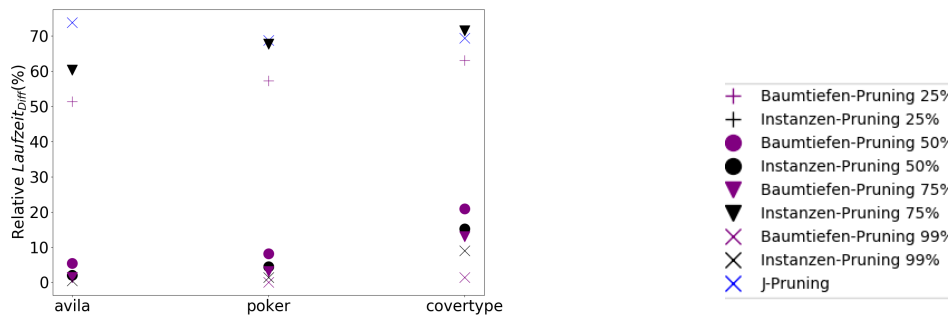
6.3. Evaluation der Performanz

In diesem Kapitel werden die Laufzeitersparnisse der Entscheidungsbaumalgorithmen, die durch Pre-Pruning-Methoden erreicht werden, betrachtet. Hierbei wird die Laufzeit zur Durchführung des Baumtiefen- und Instanzen-Prunings als vernachlässigbar gering angenommen, da diese Pre-Pruning-Methoden lediglich Abfragen sind und keine Berechnungen durchführen. Des Weiteren werden lediglich die Pre-Pruning-Methoden betrachtet, die in dem vorigen Kapitel 6.2 als interessant befunden wurden. Dies sind das Baumtiefen- und Instanzen-Pruning sowie das J-Pruning.

Im Folgenden wird die Abbildung 6.5 betrachtet. In dieser Abbildung wird die Laufzeitersparnis, die durch Pre-Pruning-Methoden erreicht wurde, dargestellt. Dafür wird die Gesamtlaufzeit der Entscheidungsbäume, auf die Pre-Pruning angewendet wurde, mit der Gesamtlaufzeit der Entscheidungsbäume, auf die kein Pruning angewendet wurde, verglichen. Dies wird für die drei Entscheidungsbaumalgorithmen und die Testdatensätze der Regression aus Kapitel 6.2.2 gezeigt. Um die Laufzeit der Methoden Baumtiefen- und Instanzen-Pruning zu ermitteln, wird diese mithilfe der Regression für die Werte 25%, 50%, 75% und 99% Accuracy berechnet. Die Unterschiede



(a) Die Einsparungen der Laufzeit auf dem ID3-Algorithmus. (b) Die Einsparungen der Laufzeit auf dem C4.5-Algorithmus.



(c) Die Einsparungen der Laufzeit auf dem CART-Algorithmus.

Abbildung 6.5.: Einsparungen der Laufzeit durch Pre-Pruning-Methoden.

der Laufzeit ($Laufzeit_{Diff}$) werden als die Differenz der Gesamtlaufzeit des Entscheidungsbaumes ($Laufzeit_{Gesamt}$) und der Laufzeit der Entscheidungsbäume mit Pruning ($Laufzeit_{Pruning}$) berechnet:

$$Laufzeit_{Diff} = Laufzeit_{Gesamt} - Laufzeit_{Pruning} \quad (6.2)$$

$Laufzeit_{Diff}$ wird relativ zur $Laufzeit_{Gesamt}$ betrachtet und stellt die y-Achse der Abbildungen dar. Die betrachteten Datensätze stellen die x-Achse der Abbildungen dar.

Die möglichen Laufzeiteinsparungen reichen von 1% beim Poker Datensatz in Abbildung 6.5b bis zu 94% beim Soybean Datensatz in Abbildung 6.5a. In sieben der neun Abbildungen erreicht das J-Pruning die höchsten Laufzeiteinsparungen. Diese liegen für den C4.5 in Abbildung 6.5b über 80% Einsparungen und für den CART in Abbildung 6.5d über 68%. Das J-Pruning setzt in diesen Fällen vor den beiden anderen Methoden zum Kürzen des Baumes an. Lediglich beim Soybean Datensatz in Abbildung 6.5a und dem Poker Datensatz in Abbildung 6.5b liefert das Instanzen-Pruning die höchsten Laufzeiteinsparungen. Beim Soybean Datensatz erreicht das J-Pruning sogar die geringste Laufzeiteinsparung. Es ist weiterhin anzumerken, dass beim ID3 in Abbildung 6.5a die Pre-Pruning-Methoden für den Chess und Mushroom Datensatz, deutlich geringere Laufzeiteinsparungen zu erkennen sind. Dies liegt zu einem an den kleinen Bäumen, die durch den ID3 erzeugt wurden und der dadurch geringen Laufzeit. Weiterhin setzen bei diesen beiden Entscheidungsbäumen die

Pre-Pruning-Methoden erst später an. In den restlichen Abbildungen ist zu sehen, dass meist das Baumtiefen- und Instanzen-Pruning mit einer Accuracy von 99% die geringste Laufzeiteinsparungen erzielt. Dies kann darauf zurückgeführt werden, dass diese Methoden erst sehr spät das Wachstum des Baumes stoppen, um eine möglichst hohe Accuracy zu erreichen. Weiterhin ist interessant, dass das Instanzen-Pruning höhere Laufzeiteinsparungen erreicht als das Baumtiefen-Pruning. Dies liegt daran, dass das Baumtiefen-Pruning erst ab einer bestimmten Tiefe das Wachstum des Baumes stoppt und das Instanzen-Pruning gezielter bei verschiedenen Knoten das Wachstum stoppen kann. Das Baumtiefen-Pruning erzielt in fünf der Abbildungen die schlechtesten Einsparungen, das Instanzen-Pruning hingegen nur in drei der Abbildungen. Diese sind die Abbildung des Mushroom Datensatzes in Abbildung 6.5a und die beiden des Avila Datensatzes in Abbildung 6.5b und 6.5d.

Die Laufzeiteinsparungen unterscheiden sich weiterhin zwischen den Algorithmen. So ist in den Abbildungen 6.5b und 6.5d zu sehen, dass die Messungen eine deutlich höhere Streuung zu den Werten in Abbildung 6.5a aufweisen. Dies ist auf die höhere Laufzeit des C4.5 und CART zurückzuführen. Durch die längere Laufzeit sind höhere Variationen in den Laufzeiteinsparungen möglich. Weiterhin existieren durch die Größe der beiden Algorithmen mehr Möglichkeiten den Baum zu kürzen. Dies ist bei den ID3-Bäumen nicht der Fall, da die Bäume deutlich kleiner als die des C4.5 und CART sind (vgl. Tabelle 6.4 und 6.5). In einigen Abbildungen sind nicht alle möglichen Pruning-Methoden mit zugehörigen Accuracy-Werten zu sehen. Ein Beispiel ist der Mushroom Datensatz in Abbildung 6.5a, in dem nur drei Datenpunkte abgebildet sind. Dies ist darauf zurückzuführen, dass für einige Accuracy-Werte bei der entsprechenden Pre-Pruning-Methode kein Wert über die Regression gefunden wurde.

Im Allgemeinen lässt sich sagen, dass durch das J-Pruning die höchsten Laufzeiteinsparungen gewonnen werden konnten. Für das Baumtiefen und Instanzen-Pruning gilt, dass je höher die gewünschte Accuracy ist, desto geringer fallen die Laufzeiteinsparungen aus. Weiterhin erzielt das Instanzen-Pruning in den meisten Fällen eine höhere Laufzeiteinsparung als das Baumtiefen-Pruning.

6.4. Vergleich von Pre-Pruning und Post-Pruning-Methoden

In diesem Kapitel werden die Pre-Pruning-Methoden J-Pruning, Baumtiefen-Pruning und Instanzen-Pruning mit den zwei Post-Pruning-Methoden REP und EBP unter den Aspekten der Laufzeit und Accuracy der erstellten Entscheidungsbäume verglichen. Dazu wird in Kapitel 6.4.1 die Accuracy, sowie die Gesamtlaufzeit und in Kapitel 6.4.2 die Laufzeit der Pruning-Methoden miteinander verglichen.

6.4.1. Qualität und Laufzeit von Entscheidungsbäumen mit Pruning-Methoden

In diesem Abschnitt wird die Laufzeit und Qualität der verschiedenen Pruning-Methoden auf den Entscheidungsbäumen miteinander verglichen. Die Abbildung 6.6 stellt die reale Laufzeit der Entscheidungsbaumalgorithmen, der Post-Pruning-Methoden und des J-Pruning dar. Die Laufzeit der Entscheidungsbäume mit Baumtiefen- und Instanzen-Pruning wurde über die Regression aus Kapitel 6.2.2 für die gewünschten Accuracy-Werte berechnet. Diese Laufzeit wurde wie folgt berechnet: Die Regression bietet die Möglichkeit, dass ein Analyst bei der Eingabe einer gewünschten

Accuracy von $x\%$ einen Pre-Pruning-Wert erhält, mit dem dieser Accuracy-Wert erreicht werden soll. Weiterhin kann über diesen Wert in den Messungen der Entscheidungsbäume, die auf den Testdatensätzen für die Regression erstellt wurden, die Laufzeit der Pre-Pruning-Methode erfragt werden. Dies gibt dem Analysten eine Möglichkeit zu bewerten, ob ein Accuracy-Wert aufgrund seiner Laufzeit für seine Zwecke gedacht ist. Die Regressionen aus Kapitel 6.2.2 haben eine sehr schlechte Qualität, weswegen das folgende Kapitel mit Vorsicht zu betrachten ist. Jedoch werden die Regressionskurven im Folgenden weiterhin verwendet. Mit diesen soll untersucht werden, welche Zeiteinsparungen durch die Pre-Pruning-Methoden möglich sind. Im Folgenden wird die Accuracy und Laufzeit dieser Methoden für die drei Algorithmen ID3, C4.5 und CART, sowie für die drei Testdatensätze der Regression betrachtet.

In der Abbildung 6.6 ist zu sehen, dass die Laufzeit der Entscheidungsbäume mit Pre-Pruning-Methoden oft höher ist, als die Laufzeit des Entscheidungsbaumalgorithmus. Dies ist auf die ungenaue Regression, die in Kapitel 6.2.2 diskutiert wurde, zurückzuführen. Das Instanzen- und Baumtiefen-Pruning besitzen eine zu vernachlässigende Laufzeit, die Gesamtlaufzeit dieser Methoden auf einem Entscheidungsbaum kann folglich nicht weit über der Laufzeit des Algorithmus liegen. Die Regression sorgt ebenso für die fehlenden Werte in Abbildung 6.6b.

Die Laufzeit der Post-Pruning-Methoden ist meist höher als die der Pre-Pruning-Methoden. Dies ist dadurch zu begründen, dass für die Post-Pruning-Methoden zuerst der gesamte Entscheidungsbaum erstellt werden muss. Auf diesem wird anschließend die Post-Pruning-Methode ausgeführt. Dies resultiert in einer höheren Gesamtlaufzeit für Entscheidungsbäume mit Post-Pruning, als ohne oder mit Pre-Pruning. Ausnahmen sind die Abbildung 6.6a, in der der Entscheidungsbaum mit Baumtiefen-Pruning und die Abbildung 6.6c in der der Entscheidungsbaum mit J-Pruning eine höhere Laufzeit als die Bäume mit den Post-Pruning-Methoden hat. Die Laufzeit des J-Prunings ist in dieser Abbildung deutlich höher, als die aller anderen Methoden und die des Algorithmus. Dies ist auf die kurze Laufzeit des ID3-Entscheidungsbaumes auf dem Soybean Datensatz zurückzuführen. Die Berechnungen, die das J-Pruning durchführen muss sind höher als diese. Im späteren Verlauf der Accuracy der Bäume sinkt die Laufzeit der Entscheidungsbäume mit Instanzen- und Baumtiefen-Pruning auf den Poker und Covertype Datensätzen stark unter die des Entscheidungsbaumalgorithmus, jedoch bei steigender Accuracy. Diese Werte sind ebenso auf die Regression zurückzuführen. Im Allgemeinen lässt sich jedoch sagen, dass die Laufzeit der Pre-Pruning Methoden mit den C4.5- und CART-Entscheidungsbäumen auf den Datensätzen geringer ist, als die der Post-Pruning-Methoden. Insbesondere das J-Pruning erzielt durchgehend sehr niedrige Laufzeitwerte.

Die Entscheidungsbäume mit Post-Pruning-Methoden erreichen meistens die höchste Accuracy. Ausnahmen sind der Poker-Datensatz auf dem C4.5 und CART in den Abbildungen 6.6e und 6.6h sowie der Soybean-Datensatz in Abbildung 6.6c. Jedoch sind diese Accuracy-Werte für das Baumtiefen und Instanzen-Pruning sehr unwahrscheinlich, da die Accuracy deutlich höher als die der Entscheidungsbäume mit Post-Pruning-Methoden ist. Die Werte der Entscheidungsbäume mit Pre-Pruning-Methoden, die höhere Accuracy-Werte als die der originalen Entscheidungsbäume erreichen, sind folglich mit hoher Vorsicht zu betrachten. Der Entscheidungsbaum mit J-Pruning erreicht jedoch bei einem Datensatz einen höheren Accuracy-Wert als der Entscheidungsbaum mit REP. Dies ist der Fall für den Poker Datensatz auf dem CART-Algorithmus in Abbildung 6.6h.

6. Evaluation

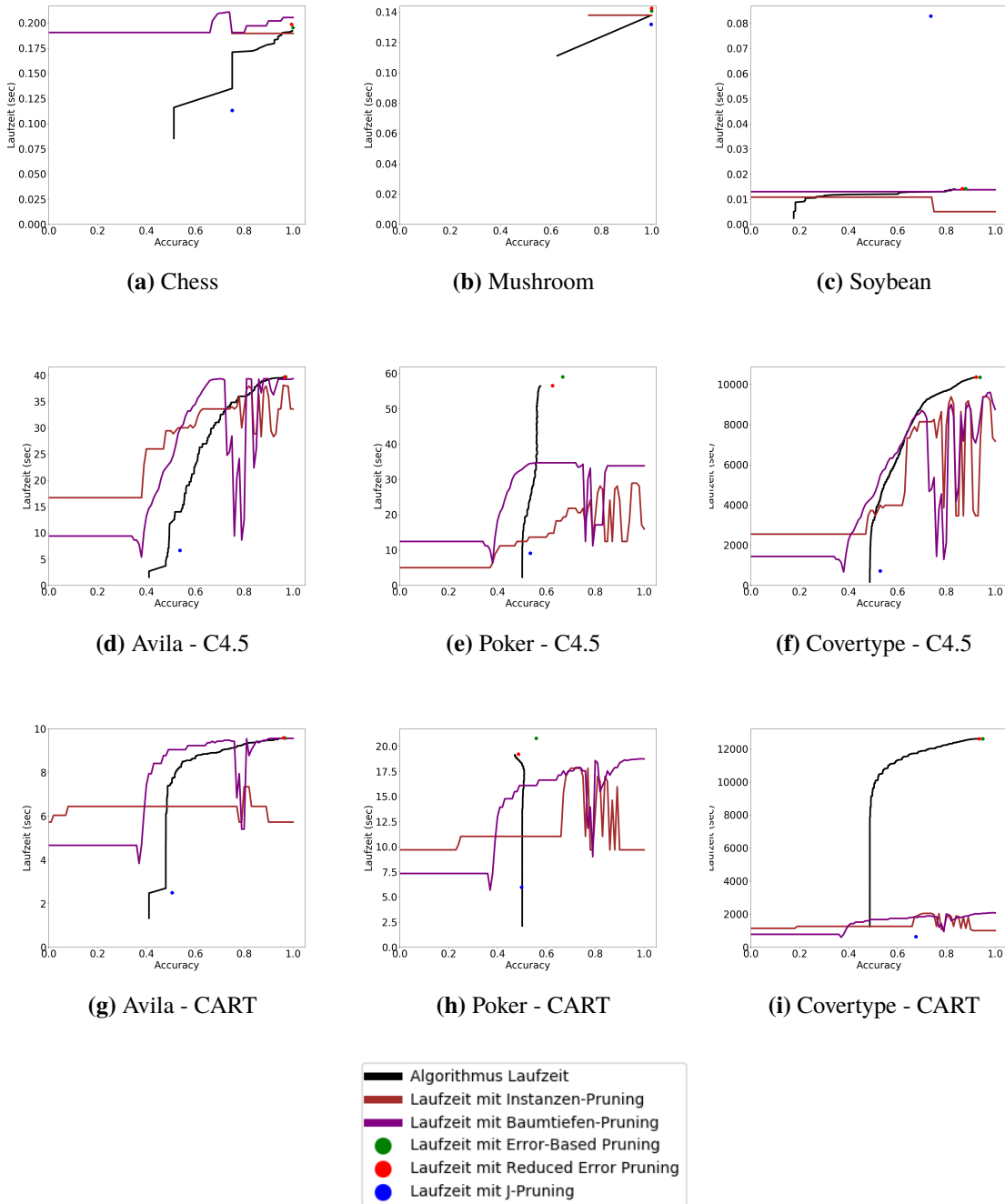


Abbildung 6.6.: Die Gesamtlaufrzeiten der Entscheidungsbäume und der Pruning-Methoden.

Abschließend lässt sich sagen, dass die Laufzeit der Pre-Pruning-Methoden meist geringer ist, als die des ursprünglichen Entscheidungsbaumes und der Post-Pruning-Methoden. Die höchste Laufzeit ist in allen Messungen bei den Post-Pruning-Methoden zu finden. Diese stellen in dem meisten Messungen jedoch auch die höchste Accuracy. Diese ist bei den Pre-Pruning-Methoden schwierig zu

erfassen, da die durch die Regression vorhergesagten Werte sehr ungenau sind. Lediglich die Werte des J-Pruning sind gut zu bewerten und hier ist die Accuracy des erlernten Entscheidungsbaumes meist deutlich schlechter, als die des ursprünglichen Baumes.

6.4.2. Laufzeit der Pruning-Methoden

In diesem Abschnitt wird die Laufzeit zur Durchführung der Pre- und Post-Pruning-Methoden verglichen. Dies bezeichnet die Laufzeit der Methoden ohne die Laufzeit des Entscheidungsbaumes, also nur die Laufzeit, die für die Berechnung und Ausführung der Methoden benötigt wurde. Die einzige Pre-Pruning-Methode, die in diesem Abschnitt betrachtet wird, ist das J-Pruning. Die anderen Pre-Pruning-Methoden werden aufgrund ihrer vernachlässigbaren Laufzeit in diesem Abschnitt nicht betrachtet. Es ist interessant zu betrachten, inwiefern Unterschiede der Laufzeit zwischen den Pre- und Post-Pruning-Methoden bestehen. So kann betrachtet werden, wie aufwändig die Berechnungen der Methoden sind. In diesem Abschnitt werden wieder die Testdatensätze der Regression betrachtet. Diese genügen, um die Ergebnisse darzustellen. Die Abbildungen für die restlichen Datensätze sind im Anhang zu finden.

Die Abbildung 6.7 wird im Folgenden betrachtet. Die Abbildungen in den Spalten sind dabei immer den Messungen eines Entscheidungsbaumalgorithmus zugehörig. In den Abbildungen zeigt sich, dass das REP in den meisten Fällen die niedrigste Laufzeit besitzt. Dies ist in sechs der neun Abbildungen der Fall. Lediglich bei dem Mushroom Datensatz in Abbildung 6.7b und dem Poker Datensatz in den Abbildungen 6.7h und 6.7e ist dies nicht der Fall. Für den Mushroom Datensatz kann dies damit erklärt werden, dass das J-Pruning den Baum früh genug abschneidet. Aufgrund dessen muss ein großer Teil des Baumes nicht erstellt werden. Dadurch müssen weniger Berechnungen für das J-Pruning ausgeführt werden, als wenn der Baum weiterwachsen würde. Die Post-Pruning-Methoden müssen hingegen den maximal ausgewachsenen Baum überprüfen. Ist dieser deutlich größer, als der durch das J-Pruning erzeugte, so ist die Laufzeit der Post-Pruning-Methoden entsprechend höher, da sie mehr Berechnungen durchführen müssen.

Weiterhin ist die Laufzeit des REP auf dem Poker Datensatz in den Abbildungen 6.7h und 6.7e interessant. Diese ist ein vielfaches höher, als die der anderen Methoden. Das ist ungewöhnlich, da das REP in den anderen Abbildungen die Pruning-Methode mit der geringsten Laufzeit ist. Die erhöhte Laufzeit ist auf die Anforderung des REP zurückzuführen, dass ein Testdatensatz für das Finden der Klassifikationsfehler benötigt wird. Der Testdatensatz des Poker-Datensatzes ist, im Gegensatz zu dem der anderen Datensätze, verhältnismäßig groß. Dies bedeutet, dass die benötigte Laufzeit für die Klassifikation des Testdatensatzes durch den erstellten Entscheidungsbaum für das Berechnen des Klassifikationsfehler, höher ist. Dies wirkt sich auf die Gesamtlaufzeit des REP aus. Das REP ist in den anderen Abbildungen die Pruning-Methode mit der geringsten Laufzeit. Dies ist der Einfachheit des REP geschuldet, da die Methode keine zusätzlichen Metriken berechnen muss sondern lediglich eine Klassifikation eines Testdatensatzes vornimmt und die dadurch entstandenen Klassifikationsfehler der Knoten vergleicht. EBP und J-Pruning müssen hingegen Metriken berechnen, um Pruning durchzuführen. EBP hat die höchste Laufzeit von den beiden Post-Pruning-Methoden in allen Messungen, ausgenommen den Poker Datensätzen. Dies liegt an den Berechnungen, die das EBP vornehmen muss. Anstatt die Klassifikationsfehler anhand eines Testdatensatzes zu bestimmen, müssen diese über eine Wahrscheinlichkeitsverteilung berechnet werden, was eine höhere Laufzeit in Anspruch nimmt.

6. Evaluation

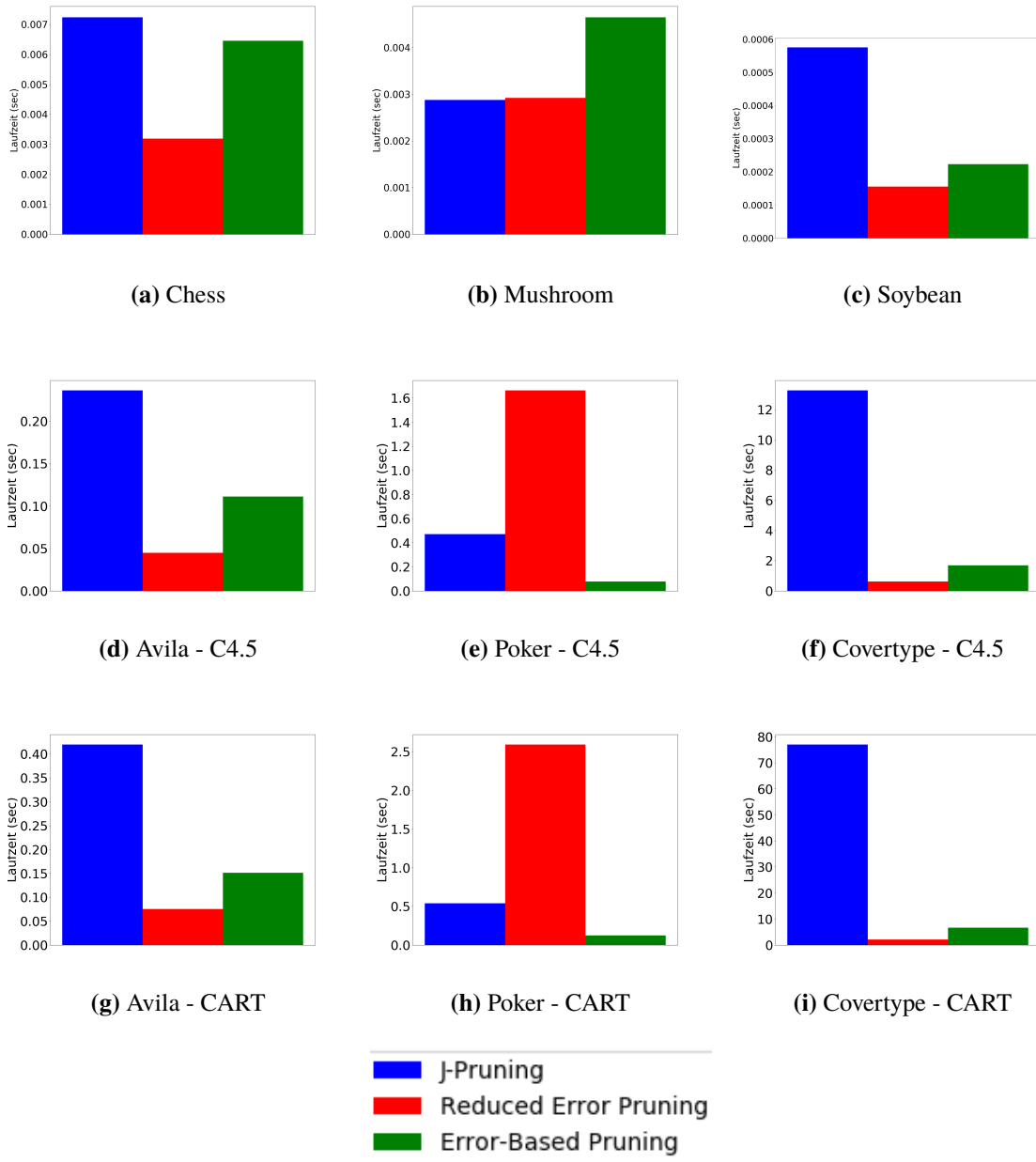


Abbildung 6.7.: Die Laufzeit des J-Pruning, EBP und REP.

Im Allgemeinen lässt sich sagen, dass das J-Pruning in den meisten Fällen die Pruning-Methode mit der höchsten Laufzeit und das REP die Pruning-Methode mit der geringsten Laufzeit ist. Die geringe Laufzeit des REP ist mit der geringen Komplexität der Pruning-Methode zu begründen. Für diese Pruning-Methode muss keine Berechnung angestellt werden, lediglich der Klassifikationsfehler von Knoten wird verglichen. Das EBP benötigt meist eine höhere Laufzeit. Dies ist der aufwändigsten

Berechnung der Klassifikationsfehler geschuldet, die von dem EBP vorgenommen werden muss. Jedoch bleibt die Laufzeit des EBP in den meisten Fällen geringer als die des J-Pruning. Des weiteren werden Post-Pruning-Methoden nur einmal für einen Entscheidungsbaumalgorithmus ausgeführt. Dies ist für das J-Pruning nicht der Fall. Dieses muss nach jeder Erstellung eines neuen Knotens die JMeasure (vgl. Kapitel 5.3) berechnen. Da dies während der Erstellung eines Entscheidungsbaumes oft der Fall ist, fällt die Laufzeit des J-Prunings entsprechend höher aus.

7. Zusammenfassung

Die Menge an Daten, die durch den Menschen und Maschinen erzeugt werden, wächst rasant an. Mit der Menge an Daten steigt die Laufzeit der Data-Mining- und Machine-Learning-Prozesse, die diese Daten analysieren. Eine Gruppe an Algorithmen, die zur Klassifikation von Daten eingesetzt wird, sind die Entscheidungsbaumalgorithmen. Diese grenzen sich von anderen Algorithmen durch die Darstellung des erlernten Modelles als Entscheidungsbaum ab. Durch die steigende Menge an Daten kann die Darstellung immer komplexer und schwieriger verständlich für menschliche Analysten werden. Um Entscheidungsbäume möglichst klein zu halten, existieren Pruning-Methoden die Entscheidungsbäume beschneiden. Bisherige Ansätze für das Pruning nutzen Post-Pruning-Methoden, die nach der Erstellung der Entscheidungsbäume ansetzen. Dies hat eine lange Gesamtlaufzeit zur Folge. Pre-Pruning-Methoden, die während der Laufzeit des Entscheidungsbaumes ansetzen, wurden bisher wenig betrachtet. Sie bieten jedoch den Vorteil einer verringerten Gesamtlaufzeit, da der Entscheidungsbaum nicht vollständig ausgewachsen muss. Das Wachstum des Entscheidungsbaumes wird bereits während der Erstellung begrenzt.

In dieser Arbeit wurde ein Verfahren vorgestellt, mit dem fünf verschiedene Pre-Pruning-Methoden auf allen Entscheidungsbäumen angewendet werden können, die der Familie der Top-Down Induction of Decision Trees (TDIDT) angehören. Insbesondere die drei Algorithmen ID3, C4.5 und CART der TDIDT-Familie wurden in dieser Arbeit betrachtet. Das Verfahren erlaubt es die Pre-Pruning-Methoden ausführlich zu evaluieren und mit existierenden Post-Pruning-Methoden zu vergleichen. Dazu wurde der allgemeine Ablauf der TDIDT-Entscheidungsbaumalgorithmen identifiziert und an zwei Schritten des Ablaufes Ansatzpunkte für Pre-Pruning-Methoden gefunden. Vier der fünf Pre-Pruning-Methoden benötigen Schwellenwerte, mit denen diese Methoden ausgeführt werden. Unterschiedliche Schwellenwerte folgern eine unterschiedliche Qualität der erstellten Entscheidungsbäume. Ein Zusammenhang zwischen Schwellenwerten, die von Pre-Pruning-Methoden benötigt werden und der Qualität der Entscheidungsbäume wurde untersucht. Die Qualität wurde dabei anhand der Accuracy untersucht. Der Zusammenhang wurde in Form einer Regression ausgedrückt, die es erlaubt für eine gewünschte Accuracy des Baumes einen entsprechenden Wert für die Pre-Pruning-Methoden zu finden.

Bei der Überprüfung des Zusammenhanges stellte sich heraus, dass zwei der Pre-Pruning-Methoden keinen Zusammenhang mit der Accuracy besitzen. Diese wurden daraufhin als Pruning-Methoden für die weitere Untersuchung verworfen. Bei den beiden anderen Methoden konnten Zusammenhänge erahnt werden. Diese Methoden sind das Baumtiefen-Pruning, welches das Wachstum ab einer angegebenen Tiefe beendet und das Instanzen-Pruning, welches das Wachstum von Knoten beendet, welche zu wenige Instanzen enthalten. Die fünfte Pre-Pruning Methode ist das J-Pruning, welche bei einem Informationsverlust in einem Pfad das Wachstum des Baumes beendet. Für das J-Pruning stellten sich die Accuracy-Werte als schwankend heraus. Für zwei der sechs Datensätze des ID3 und für je drei der sechs Datensätze des C4.5 und CART, lagen die Accuracy-Werte innerhalb eines 10% Accuracy-Bereich um den Wert des ursprünglichen Entscheidungsbaumes. Bei insgesamt drei

7. Zusammenfassung

Datensätzen lagen die erreichten Accuracy-Werte über denen des ursprünglichen Entscheidungsbaumes. Bei den restlichen Datensätze stellte sich die Methode jedoch als deutliche Verschlechterung in der Accuracy heraus.

Weiterhin wurden die Laufzeiteinsparungen, die durch die Pre-Pruning-Methoden entstanden sind, betrachtet. Hierbei stellte sich heraus, dass das J-Pruning in sieben von neun Fällen die höchsten Laufzeiteinsparungen erzeugte. Das Baumtiefen-Pruning stellte sich hingegen bei hohen Accuracy-Werten als die Pre-Pruning-Methode mit den geringsten Laufzeiteinsparungen heraus. Im Allgemeinen lässt sich bei dem Baumtiefen- und Instanzen-Pruning feststellen, dass je geringer die gewünschte Accuracy ist, die mit einem Schwellenwert für die Pre-Pruning-Methode erreicht wird, desto höher ist die Laufzeiteinsparung, die durch die jeweilige Methode erreicht wird. Durch die Pre-Pruning-Methoden wurden dabei bis an die 94% Laufzeit des ursprünglichen Entscheidungsbaumes eingespart. Bei der Anwendung des J-Prunings und des Baumtiefen- oder Instanzen-Prunings bei einer gewünschten Accuracy von 25% lag die Laufzeiteinsparung oft bei über 80%. Laufzeiteinsparungen für Baumtiefen- oder Instanzen-Prunings mit einer gewünschten Accuracy von 99% erreichten hingegen oft nur Laufzeiteinsparungen unter 20%.

Die Accuracy der Entscheidungsbäume mit Pre-Pruning-Methoden konnte nur unzuverlässig vorhergesagt werden. Es stellte sich heraus, dass die verwendete Regression ungenaue Ergebnisse zurückgab. Die Regression lieferte für eine Eingabe einer gewünschten Accuracy eines Baumes einen Schwellenwert für eine Pre-Pruning-Methode zurück. Die reale Accuracy, die mit diesem Schwellenwert erreicht wurde, war meist höher als die gewünschte Accuracy. Folglich ist die Vorhersage, die die Regression in den meisten Fällen getroffen hatte, gegenüber der realen Accuracy pessimistisch. Die Accuracy der Entscheidungsbäume mit Pre-Pruning-Methoden scheint also hoch auszufallen. Jedoch ist dies mit Vorsicht zu betrachten, da die Regressionskurven, mit der die Accuracy berechnet wurde, schlecht für diese Vorhersage geeignet sind.

Zuletzt wurden die Pre-Pruning-Methoden mit zwei existierenden Post-Pruning-Methoden verglichen. Die Entscheidungsbäume mit Post-Pruning-Methoden erreichten die höchste Laufzeit. Die Gesamtlaufzeit der Entscheidungsbäume mit Pre-Pruning-Methoden war in den meisten Fällen niedriger, als die der Bäume mit Post-Pruning-Methoden. Für das J-Pruning war dies in acht von neun Fällen der Fall und für die anderen Pre-Pruning-Methoden waren die Ergebnisse uneindeutig. Über die mit den Pre-Pruning-Methoden erreichte Qualität der Entscheidungsbäume lässt sich hinsichtlich der mit den Post-Pruning-Methoden erreichten Qualität, aufgrund der beschriebenen Regression, nur eine ungenaue Aussage treffen. Für das J-Pruning lag sie in acht der neun getesteten Fälle unter der Accuracy der Bäume mit Post-Pruning-Methoden. Für die restlichen beiden Pre-Pruning-Methoden lieferte die Regression ungenaue Ergebnisse, die nicht zuverlässig mit den Post-Pruning-Methoden verglichen werden konnten. Unter anderem wurden Accuracy-Werte zurückgegeben, die deutlich über den Werten der Accuracy der Post-Pruning-Methoden lagen und unrealistisch hoch waren. Abschließend wurde die Laufzeit des J-Pruning mit der Laufzeit der beiden Post-Pruning-Methoden verglichen. Dabei stellte sich heraus, dass die Post-Pruning-Methoden in sechs von neun Fällen eine geringere Laufzeit erreichten. Dies kann darauf zurückgeführt werden, dass die Berechnung der J-Pruning-Methode öfters während der Erstellung des Baumes ausgeführt werden muss, die Post-Pruning-Methoden jedoch nur eine einzige Ausführung benötigen.

Abschließend lässt sich sagen, dass durch das Pre-Pruning potentiell hohe Laufzeiteinsparungen auf den TDIDT-Entscheidungsbäumen möglich sind. Dies konnte anhand der Anwendung der Pre-Pruning-Methoden auf den in dieser Arbeit vorgestellten allgemeinen Ablauf der TDIDT-Entscheidungsbäume gezeigt werden. Um diese jedoch zuverlässig hinsichtlich der Qualität

bewerten zu können ist der hier getestete Ansatz nicht geeignet. Die durchgeführte Regression zeigt, dass mit dem Ansatz kein Zusammenhang zwischen den Pre-Pruning-Werten und der Genauigkeit der Entscheidungsbäume gezeigt werden konnte.

8. Ausblick

Es stellte sich in der Arbeit heraus, dass durch die Regression die Zusammenhänge zwischen Pre-Pruning-Werten und der Genauigkeit der Entscheidungsbäume nicht zuverlässig gezeigt werden konnten. Um in Zukunft mögliche andere oder bessere Verfahren zum Finden des Zusammenhanges zu erstellen wird vorgeschlagen, dass Datencharakteristika genauer untersucht werden. Da kein direkt erkennbarer Zusammenhang in dieser Arbeit gefunden wurde, jedoch einer vorhanden zu sein scheint, ist eine weitere Möglichkeit das Anwenden von Meta-Learning. Mithilfe dessen könnten diese Zusammenhänge erlernt werden.

Des Weiteren können weitere Qualitätsmetriken außer der Accuracy und Macro F1-Measure betrachtet werden, um die Qualität des Baumes zu messen. Andere Qualitätsmetriken könnten weiteren Aufschluss über den Zusammenhang zwischen der Qualität und den Pre-Pruning-Werten geben.

Bei den Pre-Pruning-Methoden, die in dieser Arbeit untersucht wurden, stellte sich eine Abhängigkeit von den verwendeten Datensätzen heraus. Folglich ist ein möglicher Ansatz, der verfolgt werden kann, das Durchführen der Experimente mit synthetischen Datensätzen. Die Pre-Pruning-Methoden können weiter untersucht werden, in dem synthetische Datensätze erzeugt werden, die gewünschte Eigenschaften enthalten. Mit diesen Datensätzen könnten gezielt Eigenschaften von Pre-Pruning-Methoden untersucht werden. So ist es denkbar, dass bei Datensätzen mit einer hohen Komplexität tiefere Bäume entstehen, die mehr Aufschluss über die untersuchten Pre-Pruning-Methoden geben könnten.

Weiterhin kann versucht werden mehr Einblicke in die Pre-Pruning-Methoden zu erhalten, in dem Kombinationen der Methoden betrachtet werden. So kann in Zukunft untersucht werden, ob ein Zusammenhang zwischen der relativen Anzahl der Instanzen in einem Knoten und der Baumtiefe hinsichtlich der Qualität eines Entscheidungsbaumes besteht.

Weitere Pre-Pruning-Methoden sind ebenso denkbar. So ist eine Methode denkbar, die während der Erstellung eines Entscheidungsbaumes einen Testdatensatz verwendet, um entsprechend auf Veränderungen nicht nur in der Qualität des Trainingsdatensatzes sondern auch der Qualität des Testdatensatzes reagieren zu können. Eine andere Möglichkeit ist, dass ab einer bestimmten Klassenverhältnis in einem Knoten dieser abgeschnitten wird. So kann etwa gesagt werden, dass das Wachstum eines Knotens, dessen Instanzen zu einem bestimmten Prozentsatz einer Klasse angehören, gestoppt wird.

Ein letzter Ansatz betrifft die Implementierung der Entscheidungsbäume. So können andere Diskretisierungsmethoden von kontinuierlichen Werten verwendet werden, um deren Auswirkung auf Pre-Pruning-Methoden zu untersuchen.

Literaturverzeichnis

- [Ber73] H. J. Berliner. „Some Necessary Conditions for a Master Chess Program“. In: *IJCAI*. 1973 (zitiert auf S. 26, 43).
- [BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN: 0-412-04841-8 (zitiert auf S. 18, 22, 23, 25, 31, 32, 39, 41, 44, 49, 50).
- [Bra02] M. Bramer. „An Information-Theoretic Approach to the Pre-Pruning of Classification Rules“. In: *Intelligent Information Processing*. Hrsg. von M. A. Musen, B. Neumann, R. Studer. Boston, MA: Springer US, 2002, S. 201–212. ISBN: 978-0-387-35602-0 (zitiert auf S. 33, 46).
- [DH00] P. Domingos, G. Hulten. „Mining High-speed Data Streams“. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '00. Boston, Massachusetts, USA: ACM, 2000, S. 71–80. ISBN: 1-58113-233-6. DOI: [10.1145/347090.347107](https://doi.org/10.1145/347090.347107) (zitiert auf S. 18).
- [EMS97] F. Esposito, D. Malerba, G. Semeraro. „A comparative analysis of methods for pruning decision trees“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.5 (1997), S. 476–491. ISSN: 01628828. DOI: [10.1109/34.589207](https://doi.org/10.1109/34.589207) (zitiert auf S. 26–28, 32, 35, 50).
- [FBS19] M. Fritz, M. Behringer, H. Schwarz. „Quality-driven early stopping for explorative cluster analysis for big data“. In: *SICS Software-Intensive Cyber-Physical Systems* (2019), S. 1–12 (zitiert auf S. 41).
- [GR12] J. Gantz, D. Reinsel. „The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east“. In: *IDC iView: IDC Analyze the future 2007.2012* (2012), S. 1–16 (zitiert auf S. 13).
- [GS91] R. M. Goodman, P. Smyth. „Rule induction using information theory“. In: *Piatetsky-Shapiro, G. and Frawley, W.J. (eds.), Knowledge Discovery in Databases* (1991) (zitiert auf S. 33, 46).
- [HFH+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. „The WEKA data mining software: an update“. In: *ACM SIGKDD explorations newsletter* 11.1 (2009), S. 10–18 (zitiert auf S. 33).
- [Min87] J. Mingers. „Expert systems—rule induction with statistical data“. In: *Journal of the operational research society* 38.1 (1987), S. 39–47 (zitiert auf S. 32).
- [NB87] T. Niblett, I. Bratko. „Learning decision rules in noisy domains“. In: *Proceedings of Expert Systems' 86, the 6th Annual Technical Conference on Research and development in expert systems III*. Cambridge University Press. 1987, S. 25–34 (zitiert auf S. 32).

- [Pea00] K. Pearson. „X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling“. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50.302 (1900), S. 157–175 (zitiert auf S. 45).
- [Qui86] J. R. Quinlan. „Induction of decision trees“. In: *Machine learning* 1.1 (1986), S. 81–106 (zitiert auf S. 13, 15–19, 32, 33, 44, 45, 49, 50).
- [Qui87] J. Quinlan. „Simplifying Decision Trees“. In: *International Journal of Man-Machine Studies* 27.August 1986 (1987), S. 221–234 (zitiert auf S. 27, 28, 32, 35).
- [Qui90] J. R. Quinlan. „Learning logical definitions from relations“. In: *Machine learning* 5.3 (1990), S. 239–266 (zitiert auf S. 17).
- [Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1-55860-238-0 (zitiert auf S. 18, 20–22, 24–26, 28, 29, 31–33, 41, 49, 50).
- [RM08] L. Rokach, O. Z. Maimon. *Data mining with decision trees: theory and applications*. Bd. 69. World scientific, 2008 (zitiert auf S. 25).
- [Sha48] C. E. Shannon. „A mathematical theory of communication“. In: *Bell system technical journal* 27.3 (1948), S. 379–423 (zitiert auf S. 19).
- [WKR+08] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, D. Steinberg. „Top 10 algorithms in data mining“. In: *Knowledge and Information Systems* 14.1 (Jan. 2008), S. 1–37. ISSN: 0219-3116. DOI: [10.1007/s10115-007-0114-2](https://doi.org/10.1007/s10115-007-0114-2). URL: <https://doi.org/10.1007/s10115-007-0114-2> (zitiert auf S. 13, 18, 37).

Alle URLs wurden zuletzt am 05. 05. 2019 geprüft.

A. Anhang

A.1. Zusammenhang zwischen Pre-Pruning-Methoden und der Macro F1-Measure

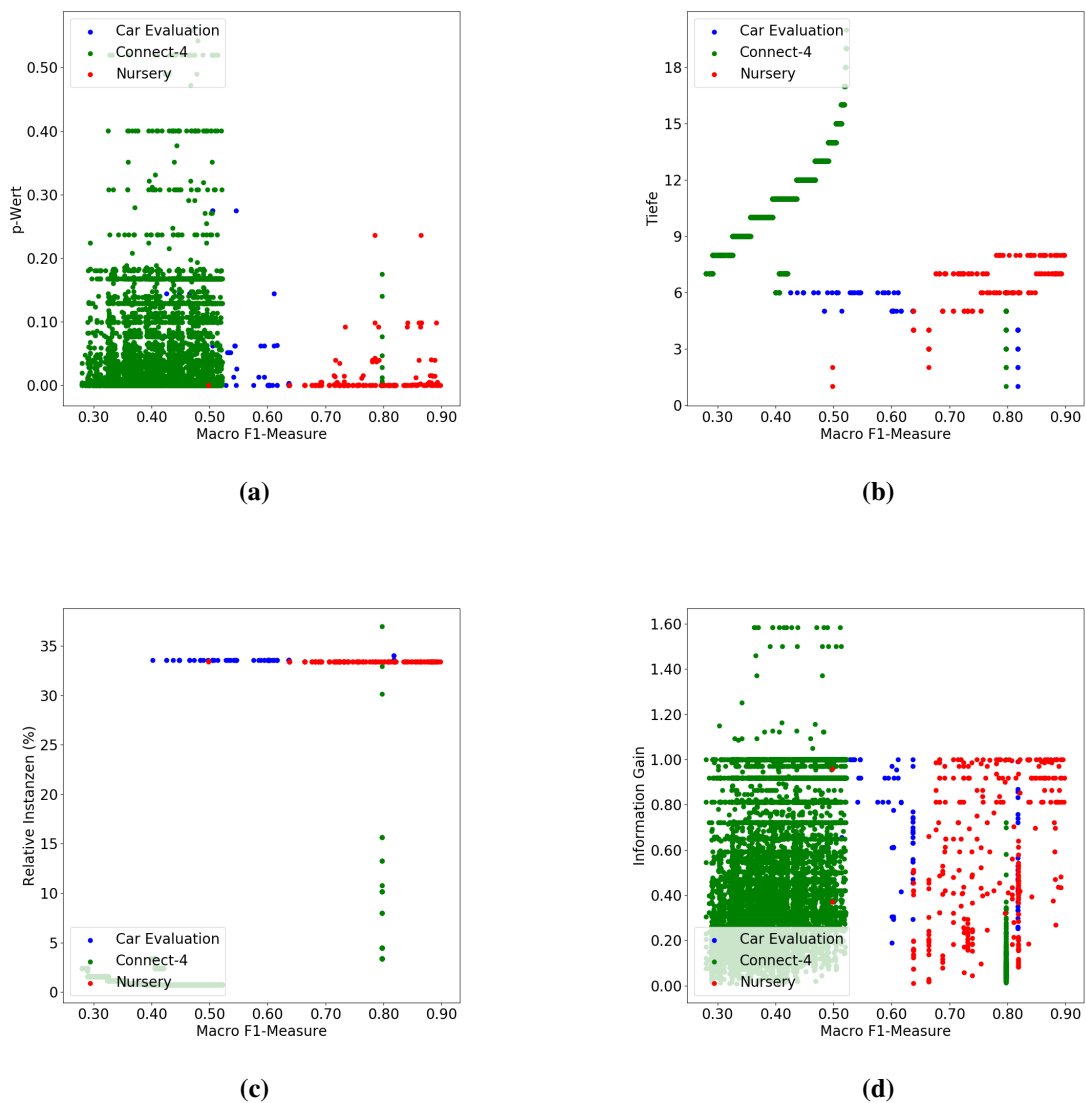
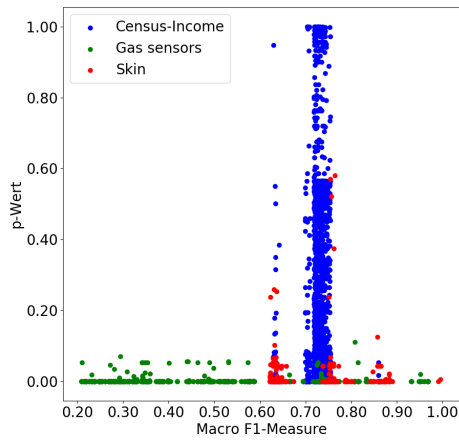
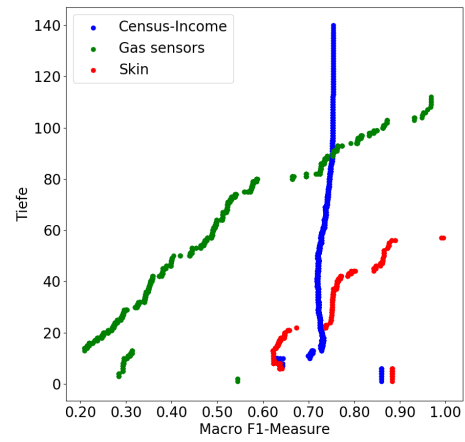


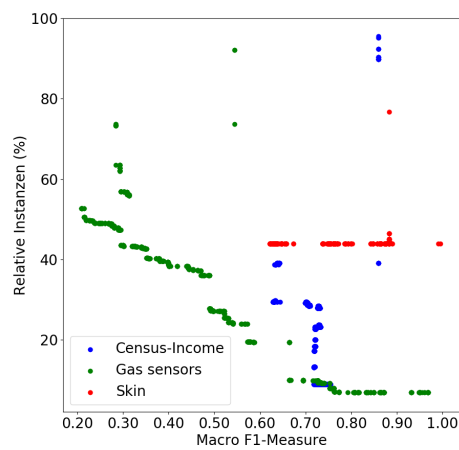
Abbildung A.1.: Scatterplots, die den Wert der Pre-Pruning-Methoden der Genauigkeit eines ID3-Entscheidungsbaumes gegenüberstellen.



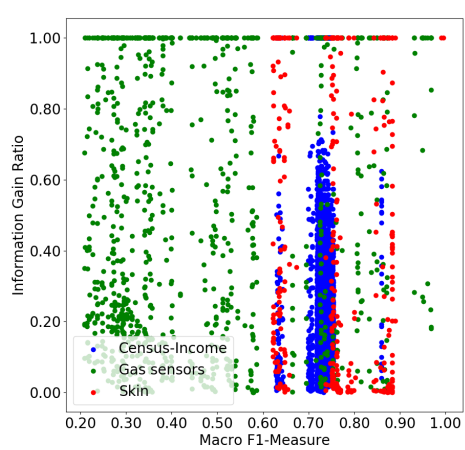
(a)



(b)



(c)



(d)

Abbildung A.2.: Scatterplots, die den Wert der Pre-Pruning-Methoden der Genauigkeit eines C4.5-Entscheidungsbaumes gegenüberstellen.

A.1. Zusammenhang zwischen Pre-Pruning-Methoden und der Macro F1-Measure

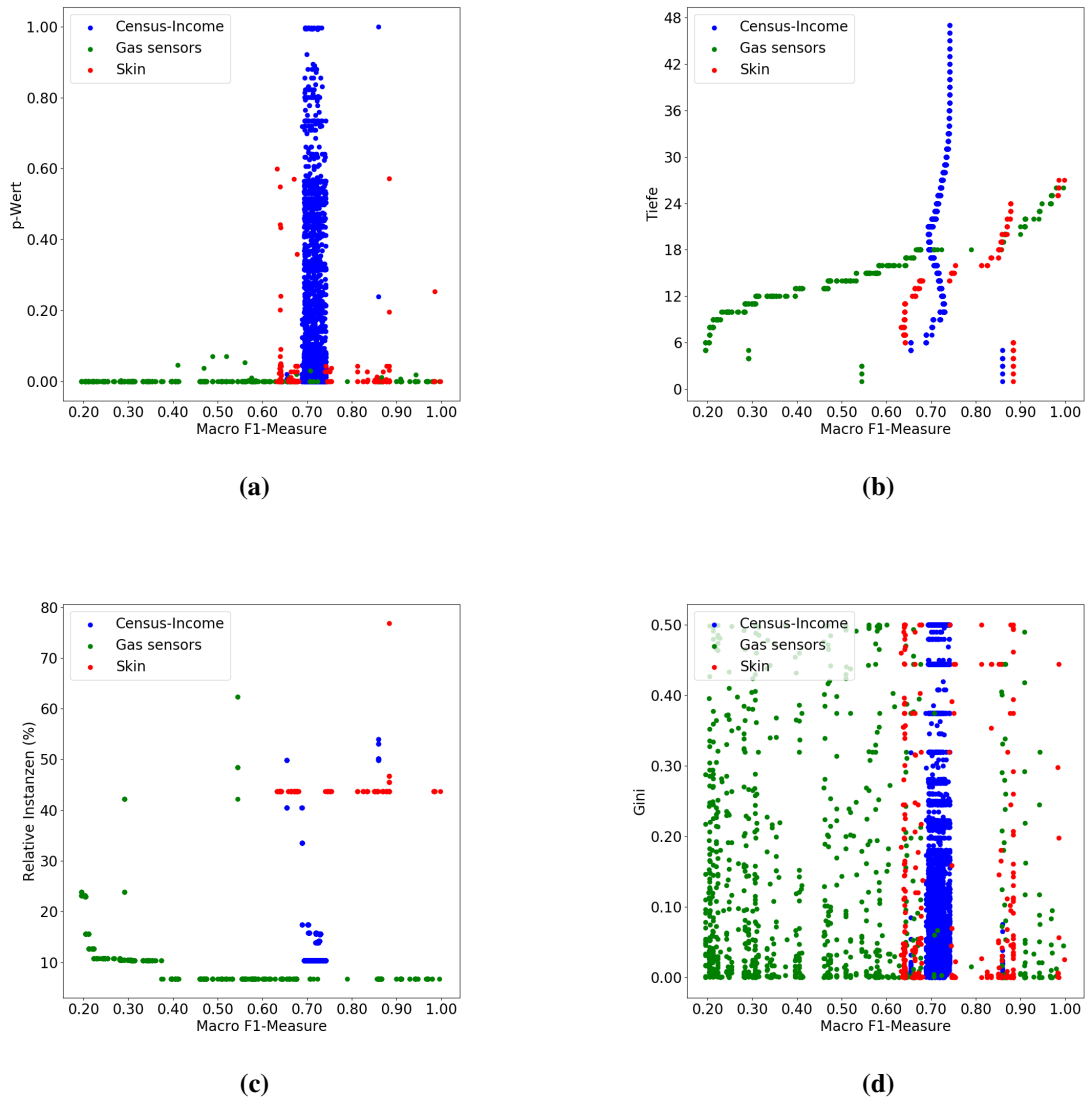


Abbildung A.3.: Scatterplots, die den Wert der Pre-Pruning-Methoden der Genauigkeit eines ID3-Entscheidungsbaumes gegenüberstellen.

A.2. Evaluation der qualitativen Zusammenhänge mit der Macro F1-Measure

Tabelle A.1. Die Regressionsmetriken der Regressionskurve der Macro F1-Measure und Pre-Pruning-Methoden. Die Werte sind auf drei Stellen nach dem Komma gerundet.

Metrik	ID3		C4.5		CART	
	Relative Instanzen	Baumtiefe	Relative Instanzen	Baumtiefe	Relative Instanzen	Baumtiefe
R^2	0.051	-0.208	-3.816	-0.317	-13.009	-2.105
Mean Squared Error	175.414	280.097	975.466	3066.139	163.218	102.934019

A.3. Qualität des J-Pruning mit der Macro F1-Measure

Tabelle A.2. Macro F1-Measure der ID3-Entscheidungsbäume mit und ohne J-Pruning

	Car	Chess	Connect	Mushroom	Nursery	Soybean
Macro F1-Measure	0.682	0.994	0.507	1.0	0.835	0.415
Macro F1-Measure J-Pruning	0.483	0.788	0.533	0.998	0.755	0.265

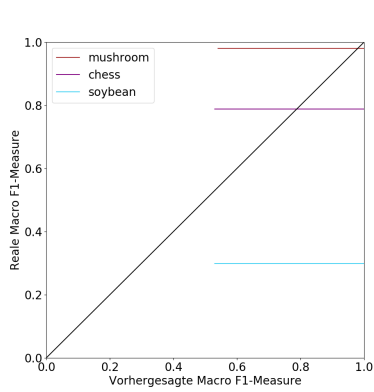
Tabelle A.3. Macro F1-Measure der C4.5-Entscheidungsbäume mit und ohne J-Pruning

	Avila	Census	Gas	Poker	Covertime	Skin
Macro F1-Measure Gesamt	0.682	0.800	0.998	0.174	0.573	0.999
Macro F1-Measure J-Pruning	0.149	0.739	0.652	0.316	0.188	0.904

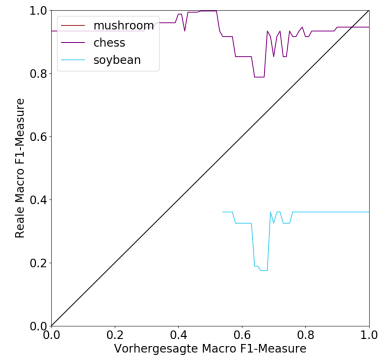
Tabelle A.4. Macro F1-Measure der CART-Entscheidungsbäume mit und ohne J-Pruning

	Avila	Census	Gas	Poker	Covertime	Skin
Macro F1-Measure Gesamt	0.659	0.772	0.999	0.134	0.603	0.999
Macro F1-Measure J-Pruning	0.370	0.740	0.542	0.413	0.452	0.900

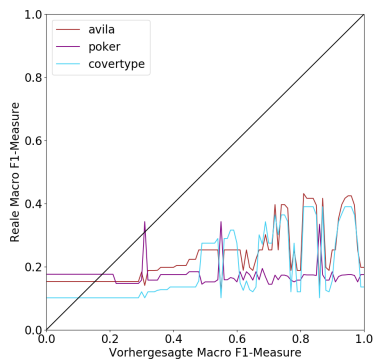
A.4. Vorhersage der Macro F1-Messure



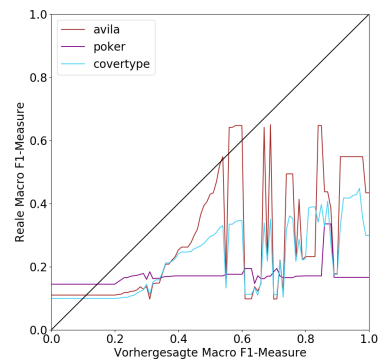
(a) Instanzen-Pruning für den ID3-Algorithmus



(b) Baumentiefen-Pruning für den ID3-Algorithmus

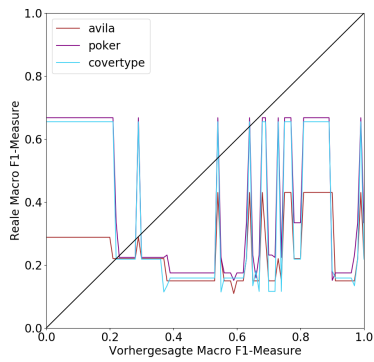


(c) Instanzen-Pruning für den C4.5-Algorithmus

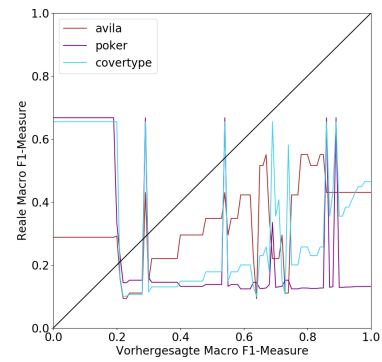


(d) Baumentiefen-Pruning für den C4.5-Algorithmus

Abbildung A.4.: Vorhersagen der Macro F1-Messure, die mithilfe der Regressionen aus Kapitel 6.2.2 getroffen wurden, für C4.5 und ID3



(a) Instanzen-Pruning



(b) Baumtiefen-Pruning

Abbildung A.5.: Vorhersagen der Macro F1-Measure, die mithilfe der Regressionen aus Kapitel 6.2.2 getroffen wurden, für den CART-Algorithmus

A.5. Vergleich von Pre-Pruning und Post-Pruning-Methoden anhand der Macro F1-Messure

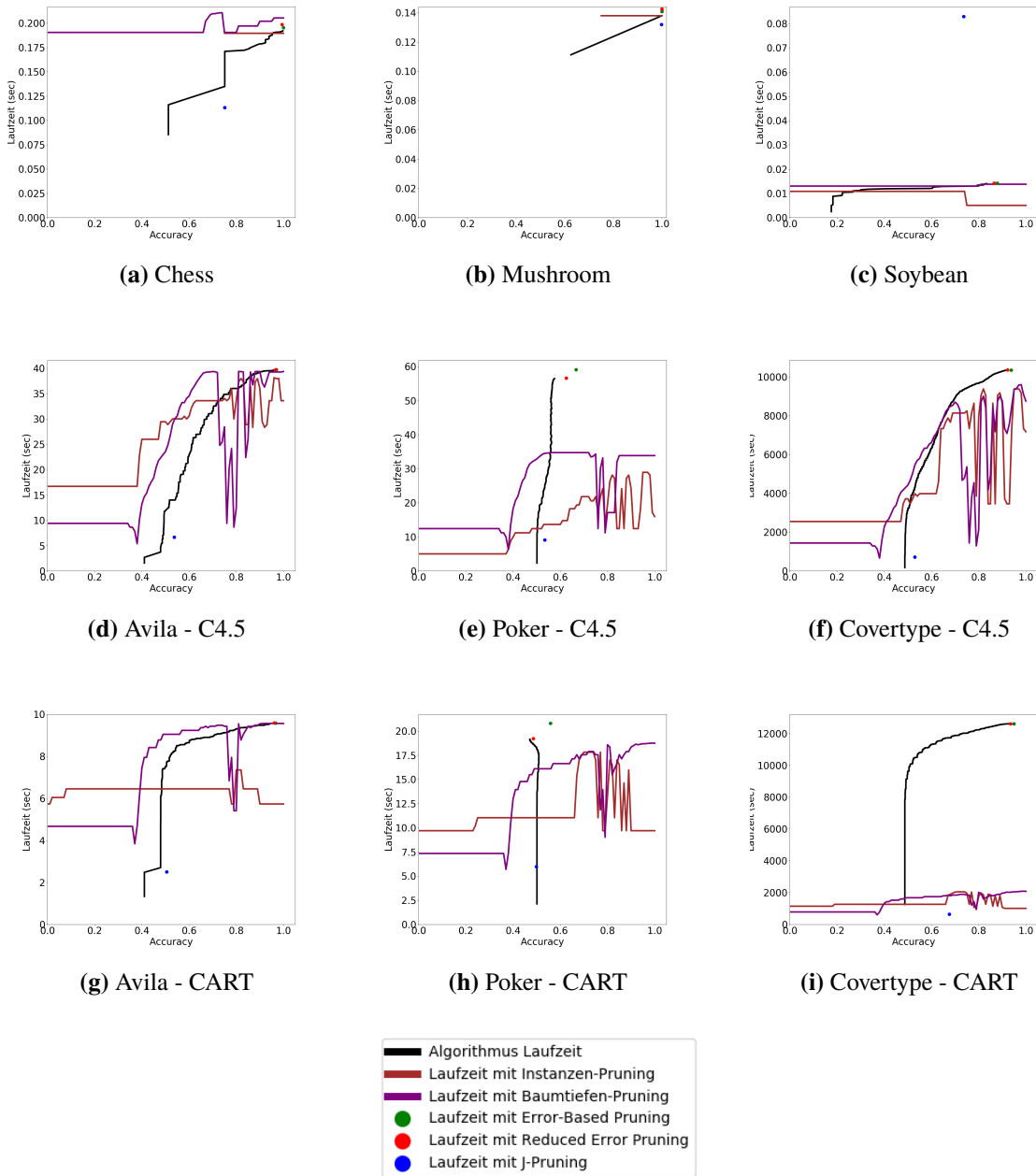
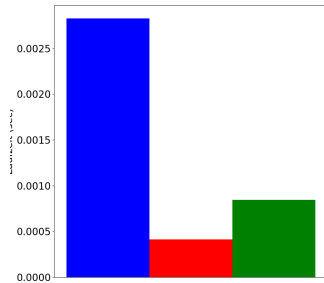
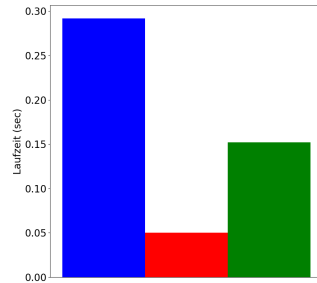


Abbildung A.6.: Die Gesamtlaufrzeiten der Entscheidungsbäume und der Pruning-Methoden.

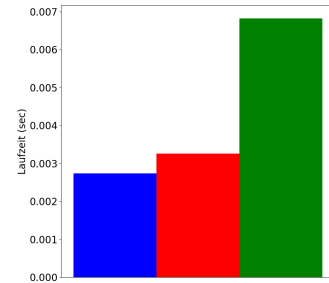
A.6. Laufzeit der Pruning-Methoden



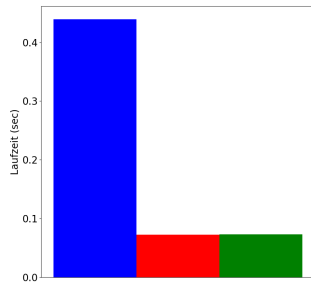
(a)



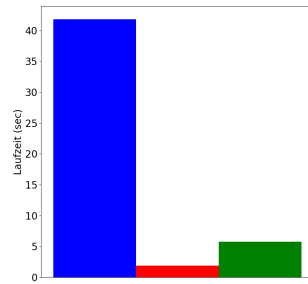
(b)



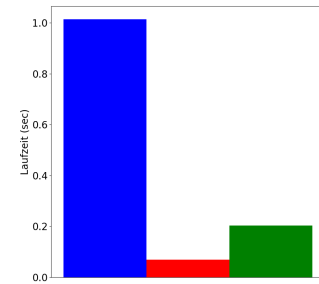
(c)



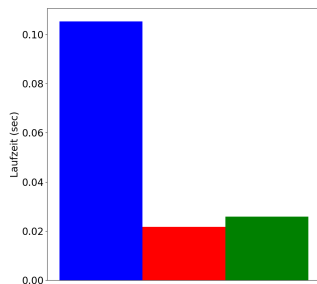
(d)



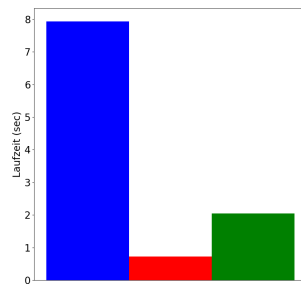
(e)



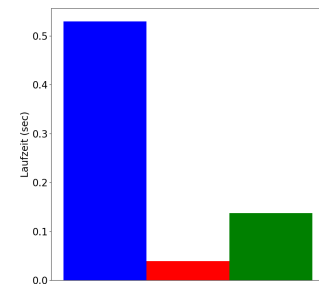
(f)



(g)



(h)



(i)

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift