Aus der Klinik für Strahlentherapie und Radioonkologie
der Medizinischen Fakultät Mannheim
(Direktor: PD Dr. Frank Giordano)

Multiple Retrieval Case-based Reasoning - Klinisches Entscheidungsunterstützungssystem auf unvollständigen Datenbanken in Anwendung für das Tumorboard

Inauguraldissertation
zur Erlangung des Doctor scientiarum humanarum
der
Medizinischen Fakultät Mannheim
der Ruprecht-Karls-Universität
zu
Heidelberg

vorgelegt von
Nikolas Immanuel Löw
aus Baden-Baden

2019

Dekan: Professor Dr. med. Goerdt

Referent: Professor Dr. rer. nat. Jürgen Hesser

Inhaltsverzeichnis

Abkürzungsverzeichnis

CBR	Cased-based Reasoning
MRCBR	Multiple Retrieval Case-based Reasoning
CBR-TDS	Entscheidungsunterstützungssystem für das Tumorboard
MOSAIQ	Datenbank der Klinik für Strahlentherapie und Radioonkologie
MCAR	Missing Completely At Random
MAR	Missing at Random
MNAR	Missing Not At Random
MI	Multiple Imputation
Int	Integer-Variable
Float	Gleitkommazahl-Variable
Kat	kategoriale Variable/ Kategoriale
MODE	häufigst vorkommender Wert
MAF	mittlerer absoluter Fehler
STD	Standardabweichung
Top N	ähnlichste N Fälle

kwFS	künstlich wahres Fall Scenario
List Delete	listenweiser Fallausschluss
Pair Delete	paarweiser Fallausschluss
Var Delete	Auslassen der Variable
N/A Sim 0	Substitution der fehlenden lokalen Ähnlichkeiten mit 0
N/A Sim 0.5	Substitution der fehlenden lokalen Ähnlichkeiten mit 1
Mean	Mittelwert Subsitution
CART	Classification and Regression Trees/ Entscheidungsbaum
RF	Random Forest
MRCBR CART	Multiple Retrieval Case-based Reasoning mit CART
MRCBR RF	Multiple Retrieval Case-based Reasoning mit RF

1 Einleitung

Cased-based Reasoning (CBR) ist ein Verfahren des maschinellen Lernens, das aufgrund des Wissens vergangener ähnlicher Fälle und ihren Lösungen Prognosen und Hilfestellungen zu einem neuen Fall liefert. Es besteht aus vier eigenständigen Phasen, die einen Kreislauf bilden. Der neue Fall und seine gewonnene Lösung werden im nächsten Durchlauf Teil dieses Kreislaufes und führen zu einer Erweiterung des bisherigen Wissens. Die Leistung des CBR hängt zum einen von der Größe und Informationsdichte der Datenbank ab und zum anderen maßgeblich von der korrekten Rangfolge der ähnlichen Fälle in der Retrieve-Phase, da alle folgenden Phasen von deren Ergebnissen betroffen sind. Mit den rasant wachsenden Datenbanken der heutigen Zeit steigert sich allerdings aus den verschiedensten Gründen auch die Menge an fehlenden Daten innerhalb dieser Datenbanken. Diese Tatsache hat zur Folge, dass das gesamte CBR auf solchen Datenbanken destabilisiert wird und besonders die Rangfolge der Retrieve-Phase darunter leidet, weil unvollständige Fälle weniger verlässlich gewertet werden können als vollständige Fälle. Überraschenderweise existieren für diese Problematik bisher kaum Arbeiten, welche den Einfluss der fehlenden Daten auf die Retrieve-Phase des CBR untersuchen und eine zuverlässige Lösung basierend auf modernen Verfahren bieten. Insbesondere fehlt es an einer umfassenden Lösung, welche unterschiedliche Arten von Variablen betroffen von unterschiedlichen Typen von fehlenden Daten verarbeiten kann.

Als Antwort auf diese Fragestellung wird in dieser Arbeit das Multiple Retrieval Case-based Reasoning (MRCBR) vorgestellt und evaluiert. MRCBR ist ein Framework für CBR auf unvollständigen Datenbanken, das eine möglichste korrekte Rang-

folge der ähnlichen Fälle mit Hilfe von modernen Methoden des maschinellen Lernens und der Statistik gewährleistet. Es bezieht die Verteilung der vollständigen Daten und die mögliche Verteilung der unvollständigen Daten in seine Berechnungen mit ein, indem es die Vorteile der Multiple Imputation und CBR in einem Verfahren effizient vereint. Das Verfahren wurde als Erweiterung des klinischen Entscheidungsunterstützungssystems für das Tumorboard (CBR-TDS) entworfen, damit dieses auf der unvollständigen Datenbank der Klinik für Strahlentherapie und Radioonkologie (MOSAIQ) fehlerfrei und vertrauenswürdig arbeiten kann. In diesem Hinblick wurde es optimiert und getestet. Des Weiteren ist es jedoch ein allgemeines Verfahren, das nicht auf diesen Anwendungsbereich allein beschränkt ist und für jedes CBR System angepasst werden kann. Es erlaubt die Verarbeitung der gängigen Arten von Variablen in medizinischen Datenbanken, numerische und kategoriale Variable, und aller Typen von fehlenden Daten.

Die Methodik des MRCBR wurde mit acht konkurrierenden Methoden des letzten Standes der Technik verglichen, welche in der Lage sind CBR im Kontext von fehlenden Daten auszuführen. Die Ergebnisse auf der wahren vollständigen Datenbank bildeten die Referenz für die Einstufung der Ergebnisse der unterschiedlichen Verfahren mit Hilfe zweier verschiedener Fehlermaße. In vier repräsentativen Experimenten bestehend aus mehreren eigenständigen Versuchen wurden verschiedene Umgebungen und Bedingungen der fehlenden Daten realistisch simuliert und untersucht. Auch der Einfluss der Größe der Datenbank und andere Parameter des CBR wurden in Betracht gezogen. Für eine korrekte statistische Auswertung entsteht das Ergebnis jeder Methode in jedem Versuch aus der Mittelung von 200 einzelnen Ergebnissen.

MRCBR hat in so gut wie allen Versuchen die bestehenden Methoden übertroffen und zeigte verlässliche stabile Ergebnisse in fast jedem der Experimente. Besonders in großen Datenbanken und einer großen Anzahl von unvollständigen Variablen konnte es seinen Abstand zu den anderen Methoden noch vergrößern. Die Analyse des Verhaltens der Verfahren zeigte, dass es keine Möglichkeit gibt fehlende Daten zu ignorieren ohne die Leistung des CBR drastisch zu reduzieren.

In den folgenden Abschnitten dieses Kapitels wird eine vertiefende Einführung in die Problematik von CBR im Kontext von fehlenden Daten gegeben, so wie die theoretischen Grundlagen der Verfahren erläutert auf die sich MRCBR gründet und welche die konkurrierenden Methoden verwenden.

1.1 Einführung

Die Möglichkeiten und Errungenschaften der modernen Verfahren des maschinellen Lernens halten immer mehr Einzug in so gut wie alle Anwendungsbereiche. Besonders in der Medizin und dem Gesundheitswesen allgemein ist diese Entwicklung deutlich spürbar [35]. Cased-based Reasoning (CBR) ist eines dieser Verfahren und wurde im Hinblick darauf entwickelt den menschlichen Entscheidungsprozess zu imitieren. Es nutzt das in den Fällen einer Datenbank gesammelte Wissen, um eine Lösung für einen Zielfall zu finden oder abzuleiten [51]. Sein intuitiver Ansatz und seine einfache Anwendung haben dazu beigetragen, dass es mittlerweile in mannigfaltigen Anwendungsfeldern zum Einsatz kommt, in denen eine Entscheidungsunterstützung gefragt ist. Die Weiterentwicklung von CBR an neu gestellte Herausforderungen nimmt in vielen Anwendungsbereichen zu, doch gerade im medizinischen Bereich ist dieser Fortschritt kaum merklich. Bedauerlicherweise werden nur wenige entwickelte CBR System tatsächlich in der Praxis eingesetzt und es ist dringend notwendig diesen Umstand zu ändern [96]. Einer der Gründe für die Stagnation in Entwicklung und Einsatz liegt nach der Ansicht und Erfahrung des Autors darin begründet, dass ein erheblicher Teil der klinischen Datenbanken von fehlenden Daten betroffen ist und dadurch nicht effizient genutzt werden kann. Auch wenn es verschiedene Ansätze gibt diese Umstände in der Zukunft zu ändern, wird es zum jetzigen Zeitpunkt keine schnelle Lösung für diese Problematik geben und eine Auseinandersetzung damit ist unentbehrlich [39]. Umso überraschender ist es, dass für CBR bisher nur wenige Forschungsarbeiten existieren, welche den Einfluss und die Auswirkungen von fehlenden Daten auf die Leistung des gesamten CBR untersuchten und Lösungsansätze dafür entwickelten.

Entscheidungsunterstützungssystem für das Tumorboard

Diese Arbeit liefert einen modernen und effizienten Ansatz, um eine Antwort auf die Fragestellung zu geben, wie CBR auf unvollständigen Datenbanken mit stabilen und verlässlichen Ergebnissen verwendet werden kann. Die Ursprünge für den Fokus dieser Arbeit liegen in der Strahlentherapie und Radioonkologie. Neben Herz-Kreislauf-Erkrankungen gehören Tumorerkrankungen zu den häufigsten Todesursachen in den Industrienationen. Die individuelle Therapie unterscheidet sich von Patient zu Patient und setzt sich aus den drei großen Bausteinen chirurgischer Eingriff, Chemotherapie und Strahlentherapie zusammen, welche sich ihrerseits unterteilen. Um eine optimale Behandlung des Patienten zu gewährleisten wurde das Tumorboard ins Leben gerufen. Das Tumorboard ist eine interdisziplinäre Konferenz in der Fachärzte der unterschiedlichen Abteilungen gemeinsam die Patientenfälle besprechen und über die optimale Behandlung entscheiden. Die endgültige Entscheidung ist das gebündelte Resultat aus dem spezifischen Wissen und der Erfahrung jedes einzelnen Arztes. Um die Ärzte des Tumorboards in ihrem Entscheidungsprozess zu unterstützen, wurde in einem Kooperationsprojekt mit der Firma celsius37 und der experimentellen Strahlentherapie des Universitätsklinikums Mannheim das Entscheidungsunterstützungssystem für das Tumorboard (CBR-TDS) entwickelt. Das CBR-TDS bietet den Ärzten die Möglichkeit auf den gesamten Wissensschatz einer Patientendatenbank zurückzugreifen und ähnliche Fälle zum aktuellen Patienten präsentiert zu bekommen. Diese früheren Fälle leisten bei komplizierten Befunden mit ihrer Geschichte, Medikation, Bildern und anderen Vermerken eine Hilfestellung für die Entscheidungsfindung.

Cased-based Reasoning

Die Stärke von CBR liegt darin, Informationen und Lösungen für einen Zielfall von früheren ähnlichen Fällen in einer Datenbank zu erhalten. Diese Informationen können weiterverarbeitet oder für eine Entscheidung direkt genutzt werden. Hierfür bildet CBR einen Kreislauf aus vier Phasen: die Retrieve-Phase zum Finden und Sortieren der ähnlichen Fälle, die Reuse-Phase zum Anpassen der Lösungen dieser Fälle an den Ziel-

fall, die Revise-Phase zum Prüfen dieser Lösung und die Retain-Phase zum Speichern des Zielfalles mit seiner angepassten Lösung in der Datenbank [92]. Aufgrund dieses Kreislaufes lernt CBR mit jedem neuen Zielfall dazu und verbessert seine Vorhersage in zukünftigen Anfragen. Diese Art der Lösungssuche ist der Entscheidungsfindung eines erfahrenen Arztes verwandt. In den letzten Jahrzehnten entstanden unzählige verschiedene CBR Systeme in der Medizin [28]. Die Hauptanwendungsfelder in der Medizin erstrecken sich von Klassifikation über Diagnose zu Planungsunterstützung und Kombinationen daraus [18]. Obwohl die Erzeugung und Repräsentation der Fälle in der Datenbank keine eigene Phase innerhalb des CBR Kreislaufes einnimmt, ist sie eine der kritischen Schritte für das weitere Verhalten von CBR [49]. Dadurch sind alle Anwendungen mit CBR nur so gut wie die Datenbank auf die sie sich berufen.

Datenbank der Klinik für Strahlentherapie und Radioonkologie

Exemplarisch dafür steht die Datenbank der Klinik für Strahlentherapie und Radioonkologie (MOSAIQ) des Universitätsklinikums Mannheims auf der sich CBR-TDS gründet. MOSAIQ ist ein Produkt von Elekta AB (Stockholm, Schweden) und beinhaltet die Daten von mehr 50000 Patienten mit einem Wachstum von einigen tausend Fällen pro Jahr. Es enthält Dutzende von spezifischen Variablen für jeden Patienten von einfachen Basisdaten bis zu kritischen onkologischen Befunden. Die Variablen unterscheiden sich in zwei Hauptarten, numerische (Integer und Gleitkommazahl) und kategoriale Variable, wobei die numerischen den größeren Teil in der Datenbank einnehmen. Aus verschiedensten Gründen sind die Fälle in MOSAIQ unvollständig, so dass annähernd jede Variable in unterschiedlichen Schwergraden davon betroffen ist. Fehlende Daten sind in den meisten medizinischen Datenbanken ein unerwünschtes Übel. Kränkere Patienten mit längeren Krankheitsverläufen haben weniger fehlende Daten [34]. Was dazu führt, dass die positiven Verläufe weniger detailliert dokumentiert werden. Dies macht den Einsatz von Verfahren des maschinellen Lernens, wie CBR-TDS, auf MOSAIQ äußerst schwierig und schränkt die Möglichkeiten passende Vorhersagen zu erstellen erheblich ein. Die bisher genutzten klassischen Methoden, die betroffenen Variablen und Fälle zu

ignorieren, führen weiterhin zu einem großen Informationsverlust unter der jede Form der Analyse leidet [50].

Fehlende Daten und ihre Typen

Ein adäquater Umgang und die sorgfältige Verarbeitung von unvollständigen Datenbanken sind nur mit dem Wissen um die Struktur der fehlenden Daten möglich. Fehlende Daten teilen sich allgemein in drei Typen auf, die sich auf der Bedingung unter der sie gelöscht wurden definieren und im späteren Verlauf der Arbeit genauer erläutert werden [115]. Für Missing Completely At Random sind die fehlenden Daten zufällig verteilt. Für Missing Not At Random hängt die Löschung nur von der Information der Variable selbst ab. Für Missing At Random hängt die Löschung von der Information einer anderen Variablen ab. Jeder dieser Typen verhält sich unterschiedlich in Bezug auf eine auf ihm angewendeten Methode. Vor allem MNAR muss mit Vorsicht behandelt werden, da der mögliche totale Informationsverlust innerhalb eine Variable zu kritischen Verzerrungen in den Ergebnissen führen kann. Allerdings ist die Kenntnis über den Typ und die Herkunft der fehlenden Daten schwierig mit algorithmischen Mitteln zu bewerkstelligen [57].

Verfahren für den Umgang mit fehlenden Daten

Über die letzten Jahrzehnte haben sich zwei Philosophien entwickelt, um mit den fehlenden Daten umzugehen. Zum einen die klassischen Eliminierungsverfahren (Complete-case Analysis), welche die fehlenden Daten ignorieren [7]. Dies bedeutet die betroffenen Fälle oder Werte nicht in die Verarbeitung der Analyse eines Verfahrens mit aufzunehmen und somit die Information dieser komplett zu löschen. Es ist die häufigste angewendete Methodik, sowohl bei Anwendern als auch kommerziellen Programmen. Zum anderen gibt es verschiedenste Ansätze die fehlenden Daten wiederherzustellen und zu ersetzen. Der einfachste Ansatz substituiert den fehlenden Wert mit einem anderen festen Wert, zum Beispiel dem Mittelwert der betroffenen Variable [32]. Tiefergehende Methoden des maschinellen Lernen imputierten die fehlenden Werte individuell anhand der

Verteilungen und Eigenschaften aller Variablen der Datenbank, wie der Klassifikations-Algorithmus Random Forest [23]. Die Ergebnisse dieser singulären Imputation können durch den modernen statistischen Ansatz der Multiple Imputation noch deutlich verbessert werden [115]. Die Multiple Imputation erzeugt eine gewisse Anzahl von imputierten Datenbanken mit Hilfe eines beliebigen Imputations-Algorithmus. Jede dieser imputierten Datenbanken unterscheidet sich leicht von der anderen, so dass die Unsicherheit betreffs der Imputation unterstrichen wird. Die Ergebnisse des auf den verschiedenen Datenbanken verwendeten Analyse-Verfahrens werden dann am Ende gemittelt.

CBR im Kontext von fehlenden Daten

Die Retrieve-Phase des CBR ist der kritischste Schritt innerhalb des CBR Kreislaufes, da alle anderen Phasen von ihr abhängen und sie äußert empfindlich auf Störungen in der Datenbank reagiert. Eine Zunahme von fehlenden Daten geht direkt mit einer Reduzierung der wahren ähnlichen Fälle zum Zielfall einher, was zu einer Verminderung der Genauigkeit und Verlässlichkeit der Ergebnisse in der Retrieve-Phase führt [73]. Eine Analyse mit CBR auf unvollständigen Datenbanken kann daher nicht ohne Vorbehalt durchgeführt werden. Insbesondere für das CBR-TDS als klinische Entscheidungsunterstützung ist eine korrekte Rangfolge der ähnlichen Fälle zum aktuellen Patienten für eine vertrauenswürdige Vorhersage unerlässlich.

Verfahren für CBR im Kontext von fehlenden Daten

Gleichwohl wurde diese wichtige Problematik bisher nur in einigen wenigen Forschungsarbeiten aufgegriffen. Methoden, welche für CBR im Kontext mit fehlenden Daten entwickelt und geprüft wurden, sind rar. Auf der einen Seite stehen die universell nutzbaren Methoden, die keine Beschränkung bezüglich der Datenbank haben und sowohl numerische als auch kategoriale Variablen verarbeiten können. Diese Methoden substituieren einfach einen festen Ähnlichkeitswert für die Ähnlichkeit der fehlenden Daten [91, 110, 74]. Auf der anderen Seite wurden spezifische Systeme entworfen, die Bedin-

gungen an die Beschaffenheit der Datenbank oder die Art der Variable stellen. Die Systeme decken auf unterschiedliche Weise verschiedene Bereiche ab: angepasste Ähnlichkeitswerte für kategoriale Variablen [4], Gewichtung der Ergebnisse eines Entscheidungsbaumes durch Nächste-Nachbarn Metrik auf nicht reduzierbaren Datenbanken [73], CBR als Imputierungs-Algorithmus für sich selbst auf einer Längsschnittstudie [119, 103], Bestrafung der Ergebnisse für numerische Variablen [55], Ermittlung eines Grades der fehlenden Daten [124] und Clustering der Datenbank mit einer Ontologie um die fehlenden Daten zu umgehen [122]. Viele dieser Forschungsarbeiten leiden unter dem Fehler einer soliden statistischen Evaluation ihrer Ergebnisse. Ein vergleichender Überblick der verschiedenen Methoden und eine Bewertung dieser stehen bis jetzt noch aus.

Es ist anzumerken, dass keine der existierenden Methoden die intrinsischen statistischen Eigenschaften der ganzen unvollständigen Datenbank mit in Betracht zieht, um eine zuverlässige Verarbeitung der fehlenden Daten zu bieten und eine möglichst korrekte Rangfolge der ähnlichen Fälle in der Retrieve-Phase zu gewährleisten. Auch die Möglichkeiten und Wirkungen der verfügbaren modernen Imputations-Verfahren auf die Leistung der Retrieve-Phase des CBR wurden zum jetzigen Zeitpunkt nicht untersucht. Des Weiteren fehlt es an einem allgemeinen verifizierten Lösungsansatz für die verschiedenen Typen von fehlenden Daten und Variablen. Eine Untersuchung anderer relevanter Einflüsse, wie die Parameter des CBR oder die Größe der Datenbank, ist nicht existent.

Multiple Retrieval Case-based Reasoning

Aufgrund der genannten Mängel der bisherigen Verfahren wurde das Multiple Retrieval Case-based Reasoning (MRCBR) für die Nutzung von CBR auf unvollständige Datenbanken entwickelt. Besonders im Hinblick auf das Entscheidungsunterstützungssystem für das Tumorboard (CBR-TDS) ist dies von Relevanz. Die Methodik und Evaluation des MRCBR wurden im Journal of Biomedical Informatics vom Autor dieser Arbeit veröffentlicht [69]. Der Gedanke dahinter ist es, die Multiple Imputation mit einem

Klassifikations-Algorithmus in den Kreislauf des CBR einzubinden und so die Stärken dieser Verfahren optimal zu vereinen. Dies geschieht, in dem ein individuelles Retrieval auf jeder der imputierten Datenbanken, welche durch eine Multiple Imputation erzeugt wurden, ausgeführt wird. Die Ergebnisse auf diesen Datenbanken, sprich die berechneten Ähnlichkeiten der Fälle zum Zielfall, werden in einem Poolingschritt zu einem finalen Ergebnis vereint. Dies stellt eine effiziente Nutzung der gesamten Information der Datenbank sicher und berücksichtigt die mögliche Verteilung der imputierten fehlenden Daten, welche ihrerseits die Ähnlichkeiten der Fälle positiv beeinflussen. Dadurch ist aus statistischer Sicht eine stabile Rangfolge der ähnlichen Fälle weitgehend garantiert, die sich auch weniger anfällig bezüglich des Typs der fehlenden Daten zeigt.

Ausblick der Arbeit

In den kommenden Abschnitten dieses Kapitels werden die Grundlagen für die Entwicklung des MRCBR präsentiert. Zu Beginn wird Case-based Reasoning und seine Anwendungsbereiche in der Medizin vorgestellt. Darauf folgt eine Einführung in die Welt der unvollständigen Datenbanken und der verfügbaren Verfahren zur Verarbeitung dieser. Das Kapitel endet mit einer kritischen Abhandlung der bisherigen Methoden für CBR im Kontext von fehlenden Daten.

Im Anschluss daran wird in Kapitel 2 die Methodik des Multiple Retrieval Case-based Reasoning explizit erläutert. Dies betrifft zum einen die Motivation für Forschung und Ausarbeitung von diesem Verfahren und zum anderen die detaillierte Beschreibung des Algorithmus. Außerdem werden der Aufbau und die verwendeten Mittel für die Evaluation dargelegt.

Kapitel 3 präsentiert die Ergebnisse der Evaluation, welche MRCBR in verschiedenen Umgebungen von unvollständigen Daten untersucht und in direkte Konkurrenz zu den Methoden aus Kapitel 1 setzt. Auch die Umgebungen untereinander und das Verhalten der Methoden in diesen wird untersucht. So wird ein umfassendes Bild von MRCBR und den konkurrierenden Methoden zur Analyse erschaffen.

Die anschließende Diskussion in Kapitel 4 greift die Ergebnisse der Analyse auf und erörtert die Beobachtungen unter verschiedenen Aspekten und ihren Auswirkungen. Am Ende gibt es einen Ausblick auf mögliche zukünftige Forschungsschwerpunkte.

Den Abschluss bildet eine Zusammenfassung der Arbeit in Kapitel 5.

1.2 Case-based Reasoning

Cased-based Reasoning (CBR) ist ein Verfahren des maschinellen Lernens, welches aufgrund von Wissen aus vergangenen Erfahrungen Antworten auf eine Fragestellung findet und mit jeder Anwendung dieses Wissen ausbaut [63, 51, 92]. Die grundlegende Idee entstand Ende der siebziger Jahre [101], wurde in den weiteren Jahrzehnten vertieft [13] und fand seine endgültige Form mit seinen vier Phasen in den neunziger Jahren [1]. Seit der ersten internationalen Konferenz 1995 zu diesem Thema ist CBR fest als Teilgebiet der künstlichen Intelligenz verankert [27]. Sein Anwendungsspektrum erstreckt sich mittlerweile über unzählige Einsatzbereiche, in denen eine Abbildung der menschlichen Problemlösung gefragt ist. Dies reicht, um nur einige wenige zu nennen, von Produktdesign [70], Abläufe im Ingenieurwesen [107], Vorhersagemodelle in der Wirtschaft [100], Fehlerdiagnose bei Geräten [76], automatische Notfalldienste [8] bis hin zu unterschiedlichsten Bereichen in der Medizin [96].

Nach dem grundlegenden Prinzip, dass ähnliche Probleme ähnliche Lösungen aufweisen, kann CBR als Form einer Schlussfolgerung aufgrund von Analogie betrachtet werden [10]. Der intuitive Ansatz von CBR entsteht aus der Tatsache, dass er der menschlichen Entscheidungsfindung zur Problemlösung sehr nahekommt. So wie Menschen aufgrund von vergangenen Erfahrungen eine schnelle Antwort auf eine Frage finden, erfordert auch CBR keine tiefe Analyse des Problembereichs und großen Aufwand das Wissen zu erschaffen, so wie andere Verfahren der künstlichen Intelligenz, wie das regelbasierte Schließen [80]. Das gesammelte Wissen einer Datenbank, bestehend aus Fällen und ihren möglichen bereits gelösten Problemen, dient CBR als Erfah-

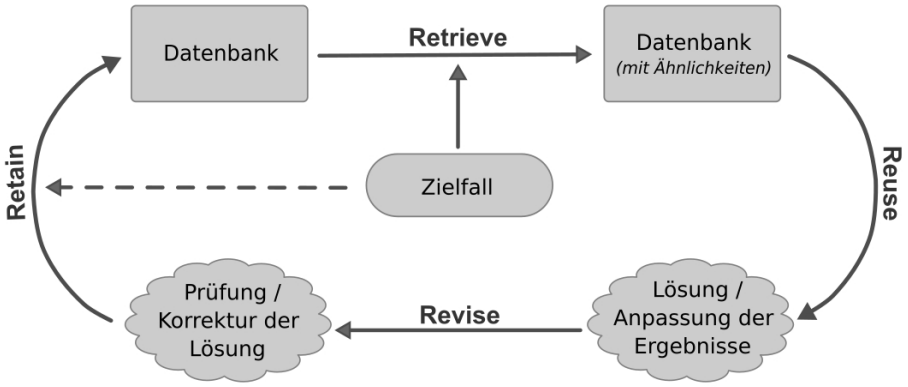


Abbildung 1.1: Kreislauf des Case-based Reasoning mit seinen vier Phasen.

rungsgrundlage, um eine Lösung für einen neuen Fall durch den Grad der Ähnlichkeit zwischen dem neuen Fall und einem oder mehreren vergangenen Fällen abzuleiten [92].

Der nächste Abschnitt wird eine formale Einführung in CBR und seine vier Phasen geben. Danach folgt ein Blick auf die verschiedenen Anwendungsbereiche in der Medizin, mit besonderem Fokus auf die Onkologie und Radiologie.

1.2.1 Kreislauf und Phasen

Cased-based Reasoning (CBR) nutzt die Informationen einer Datenbank, welche in den Fällen dieser gespeichert sind, um Prognosen und Lösungen zu einem neuen Zielfall zu liefern. Das Verfahren besteht aus vier aufeinanderfolgenden Hauptphasen: Retrieve, Reuse, Revise und Retain [1, 17, 92, 96]. Die vier Phasen bilden einen geschlossenen Kreislauf und können bei Bedarf durch weitere individuelle Phasen ergänzt werden. Eine schematische Darstellung des Kreislaufes findet sich Abbildung 1.1.

Der Ablauf ist kurzgefasst wie folgt. CBR erhält einen neuen Zielfall mit einer möglichen Fragestellung. In der Retrieve-Phase wird eine kleine Anzahl von Fälle aus der Datenbank gesucht, welche die größte Ähnlichkeit zum Zielfall haben. Diese ähnlichsten Fälle werden in der Reuse-Phase als Lösung verwendet und falls notwendig an die spezifische Problemstellung angepasst. Die Lösung dieser wird in der Revise-Phase geprüft und der Zielfall mitsamt den neugewonnenen Informationen in der Retain-Phase

der Datenbank hinzugefügt. Jeder neue Durchlauf wird somit durch die gewonnenen Informationen und Lösungen des letzten Durchlaufes ergänzt und verfeinert damit die Ergebnisse in zukünftigen Anfragen.

Retrieve-Phase

Eine Datenbank besteht aus eine Anzahl von Fällen (Reihe) und Variablen (Spalte). Aus einer Datenbank werden zu einem Zielfall die ähnlichsten Fälle gesucht und aufgelistet. Dies geschieht mit Hilfe einer Ähnlichkeitsfunktion, welche die Fälle aufgrund ihrer Ähnlichkeit zum Zielfall sortiert. Viele CBR Systeme beschränken sich auf die Retrieve-Phase und nutzen allein das Wissen um die Ähnlichkeiten der Fälle zum Zielfall.

Eine Ähnlichkeitsfunktion unterteilt sich in lokale und globale Ähnlichkeiten. Die globale Ähnlichkeit eines Falls zum Zielfall wird aus den lokalen Ähnlichkeiten der Elemente dieses Falls berechnet. Als Ähnlichkeitsfunktion bieten sich alle Klassifikations-Algorithmen an, welche eine Metrik bezüglich ihrer Elemente zulassen [66, 38, 83], wie zum Beispiel Nächste-Nachbarn-Klassifikation (kNN) [37] oder Entscheidungsbäume [12]. Die meisten Ähnlichkeitsfunktionen für CBR basieren wegen seiner einfachen Interpretierbarkeit und Handhabung auf kNN [83].

Je nach Beschaffenheit der Datenbank weisen die Variable unterschiedliche Arten auf, die von numerisch, kategorial, Fuzzy bis zu Intervall reichen [16]. Zur Berechnung der lokalen Ähnlichkeit werden für jede Art der Variablen passende Metriken verwendet [30, 87]. Bei numerischen Variablen sind der Euklidische oder Manhattan-Abstand gebräuchlich. Metriken für kategoriale Variablen sind meist starr definiert. Die Metriken können durch ihre individuellen Eigenschaften die unterschiedlichen Variablen anders gewichten und damit ihre Präsenz abschwächen oder verstärken. Es gibt Ansätze, welche die inverse Exponentialfunktion der Metrik verwenden. [45, 20]. Diese bewahren Symmetrie, Reflexivität und Transitivität und spiegelt die menschliche Vorstellung von Ähnlichkeit besser wider [37].

Sei ein neuer Zielfall $T = (t_1, t_2, ..., t_i, ..., t_n)$ und ein Fall $C = (c_1, c_2, ..., c_i, ..., c_n)$ aus der Datenbank D gegeben. Die Elemente t_i und c_i werden durch dieselbe Art der Variable repräsentiert. Die globale Ähnlichkeit (SIM) eines Falles C zum Zielfall T berechnet sich allgemein aus den lokalen Ähnlichkeiten der Elemente t_i und c_i mit Metriken sim_i wie folgt:

$$SIM(T, C) = \sum_{i=1}^{n} w_i \cdot sim_i(t_i, c_i),$$

dabei sind w_i die spezifischen Gewichte für jede Variable mit $\sum_{i=1}^{n} w_i = 1$ und $0 \leq w_i \leq 1$. Die w_i gewichten die Variablen je nach Anforderung der Anwendung gleich oder unterschiedlich. Die Anpassung der Gewichte kann manuell oder automatisch vollzogen werden [123]. Die lokalen Ähnlichkeiten können je nach Variable oder Anforderung durch verschiedene Metriken sim_i bestimmt werden.

Für jeden Fall der Datenbank wird die globale Ähnlichkeit zum Zielfall bestimmt. Die Fälle werden aufgrund ihrer individuellen globalen Ähnlichkeiten der Größe nach sortiert und erhalten einen Platz (1, 2, ..., m) entsprechend ihrer Rangfolge. Die globale Ähnlichkeit und der Platz werden bis zum Ende des Durchlaufs gespeichert.

In den weiteren Phasen wird eine bestimmte Anzahl der ähnlichsten Fälle zum Zielfall verarbeitet und ihrer Information zur Lösung der Fragestellung herangezogen. Je nach CBR System kann dies ein oder unzählige Fälle umfassen. Die medizinischen Anwendungen in den nächsten Abschnitten verwenden meist zwischen 3 und 20 Fällen.

Reuse-Phase

Die Information und Lösung der ähnlichsten Fälle aus der Retrieve-Phase werden verwendet um eine Lösung für den Zielfall abzuleiten und die gestellte Fragestellung zu beantworten. Dies kann im einfachsten Falle der Durchschnitt aus den Lösungen der ähnlichsten Fälle sein oder nur die relevante Information aus diesen. Unter gewissen Umständen ist aber auch eine Anpassung oder Bearbeitung der Lösung notwendig, wenn diese nicht den Erfordernissen entspricht.

Eine einfache Übernahme der Lösung ist typisch für Klassifikations-Anwendungen, wie Diagnose Systeme. Bei diesen ist nur eine beschränkte Anzahl von möglichen Lösungsklassen vorhanden, einhergehend mit einer großen Falldatenbank. Unter solchen Voraussetzungen ist jede potentielle Lösung in der Datenbank enthalten und daher keine Anpassung erforderlich. Für synthetische Anwendungen, wie Planungsunterstützung, ist eine Anpassung der Lösung immer notwendig, da der Lösungsraum die Anzahl der verfügbaren Fälle überschreitet oder sogar unendlich ist [17].

Es wurden mittlerweile unzählige Verfahren für die algorithmische Anpassung der Lösung in der Reuse-Phase entwickelt, die von einfachen Schätzfunktionen bis zu neuronalen Netzwerken reichen [16, 92]. Die Anpassung ist meist individuell an die Fragestellung und das System adaptiert.

Revise-Phase

Die vorgeschlagene Lösung aus der Reuse-Phase wird in einer simulierten oder realen Umgebung geprüft und getestet. Dies geschieht zumeist auf einem zurückgehaltenen Teil der Datenbank oder dem wahren Einsatzgebiet.

Bei automatischen Systemen wird der Lösung eine Bewertung für ihren Grad der Korrektheit gegeben [17]. Liegt der Grad unter einer festgelegten Schranke wird die Reuse-Phase erneut durchgeführt. Die Validierung kann auch von Experten unter aktuellen Gegebenheiten vorgenommen werden, welche bei Bedarf eine manuelle Korrektur der Lösung vornehmen.

Retain-Phase

Der Zielfall und falls vorhanden seine Lösung werden in die Datenbank mit aufgenommen. Dieses ist die Lern-Phase von CBR, da es nach jedem Kreislauf neues Wissen aufgrund des Zielfalles und seiner möglichen Lösung erhält und im nächsten Durchlauf für kommende Fragestellungen bereithält.

Das Anwachsen der Datenbank kann zu Problemen führen, wenn die Fälle nicht genug differenziert sind und dadurch die Ergebnisse in der Retrieve-Phase verwaschen

werden. Falls der Zielfall eine Redundanz mit einem anderen Fall der Datenbank aufweist, ist zu entscheiden, ob eine Speicherung erfolgen soll. Eine Reduktion der redundanten Variablen oder ein Clustering der Datenbank können für Abhilfe schaffen [125].

1.2.2 Einsatzbereiche in der Medizin

In der Medizin sind in den letzten Jahrzehnten unzählige Anwendungen mit Casedbased Reasoning (CBR) für die unterschiedlichsten Einsatzbereiche entstanden [16, 21, 96, 28]. Dies entspringt vor allem der Verwandtschaft von CBR zur menschlichen Entscheidungsfindung. Ein Facharzt nutzt zum Fällen einer Diagnose Regeln der Deduktion, welche sich auf einer Mischung aus erlerntem Wissen und gesammelter Erfahrung begründen. Die Erfahrung bezieht sich auf Patienten, die einen typischen Lehrbuchverlauf aufweisen, aber auch auf Situationen, die nicht der Norm entsprachen. Er erinnert sich bei der Aufnahme eines neuen Patienten, einen ähnlichen Fall wie diesen bereits gesehen zu haben und damit an die vergangenen Fälle, die am ähnlichsten zu diesem neuen Fall sind. All dies fließt in seine endgültige Entscheidung für die bestmögliche Behandlung des Patienten mit ein [102]. Aufgrund dessen, dass CBR mit demselben Mechanismus arbeitet, aus Wissen und Erfahrung Prognosen für eine neue Problemstellung zu finden, ist es für den Einsatz in der Medizin prädestiniert. Seine auf dem Wissen aus vergangenen Fällen abgeleitete Antwort simuliert den Prozess der Entscheidungsfindung eines Arztes, dass ähnliche Probleme ähnliche Lösungen aufweisen und es einfacher ist die Lösung eines ähnlichen Falls anzupassen, als die eines weniger ähnlichen [19]. Auch die Fähigkeit von CBR den neuen Fall Teil seiner Erfahrung werden zu lassen, um zukünftige Fragestellungen zu beantworten, entspricht dem menschlichen Lernverhalten [18]. Leider finden nur wenige der entwickelten CBR-Systeme einen Einsatz in der Praxis [96].

Die Art der Anwendung von CBR unterteilt sich für die verschiedenen Bereiche in der Medizin im groben in vier Hauptfelder: Klassifikation, Diagnose, Planung und Tutoring [41, 48, 18]. Dabei können sich diese Anwendungsfelder durchaus überschneiden

und neue Formen herausbilden [28]. Die Entscheidungsunterstützung als Mischung aus Diagnose, Klassifikation und Planung ist je nach Ausprägung ein Beispiel dafür [43, 106]. Im Folgenden wird ein Überblick über CBR-Systeme in den vier Anwendungsfelder und ihre Einsatzbereiche gegeben.

Diagnose. Diagnose-Systeme bieten eine Unterstützung im Prozess der Bestimmung einer Krankheit oder während des Verlaufs einer Behandlung. Dies geht oft mit einer Entscheidungsunterstützung einher und kann in unterschiedlicher Ausprägung von einer Palette von möglichen Szenarien bis zu einer klaren Bewertung gehen. Die Anwendungsbereiche reichen von Erkennen einer chronisch obstruktiven Lungenerkrankung [46], Feststellung von Diabetes Mellitus [97], Bewertung während minimalinvasiver Chirurgie [36] bis hin zur Prognose des weiteren Verlaufs des Krankheitsweges [88].

Klassifikation. Systeme zur Klassifikation teilen Antworten auf eine Fragestellung in vordefinierte Klassen oder Kategorien ein. Die Antwort auf eine neue Frage wird eindeutig einer dieser Klassen zugeordnet. Die Bestimmung einer Krankheit zum Beispiel. Im Unterschied zur Diagnose liefern sie aber keine Prognose über den weiteren Verlauf oder anderweitige Möglichkeiten. Anwendungen finden sich in der Psychophysiologie für Stress [14], Bestimmung einer nosokomialen Infektion [42] und der Analyse forensischer Beweise [123].

Tutoring. Das Unterrichten mit Hilfe von CBR Systemen bietet eine individuelle Unterstützung der Lernenden und Anleitung zu bestimmten Themen. Durch ein Dialogsystem kann es direkt durch Simulation von spezifischen Situationen auf die Erfahrung der Studenten eingehen. Beispiele hierfür sind e-Learning für Patienten mit Diabetes Mellitus in Einbeziehung des behandelnden Arztes [47] oder eine Empfehlung für Patienten, welcher Facharzt aufgrund der Symptome und Krankheit als bevorzugt erscheint [61].

Planungsunterstützung. Solche Systeme werden im Allgemeinen bei chronischen Krankheiten zur Erstellung eines Behandlungsverfahrens oder zur Verwaltung der Therapie angewendet. Der Großteil der Planungsunterstützungssysteme fällt auf den Bereich der

Radioonkologie. Diese werden im nächsten Abschnitt ausführlich besprochen. Andere Einsatzbereiche sind die Bestimmung der Insulin Dosierung für Diabetes Patienten [72] oder die Qualitätsbewertung von kontinuierlichen Blutzuckermessungen [60].

1.2.3 Anwendungen in der Onkologie und Radiologie

In der Onkologie wurden allgemein vor allem Systeme zur Diagnose und Klassifikation mit Hilfe von CBR entwickelt, welche als Entscheidungsunterstützung fungieren. Diese beiden Einsatzbereiche überschneiden sich in den Anwendungen oft, so dass ein Hybridsystem zur idealen Unterstützung entstanden ist. Für die Radiologie überwiegen aus plausiblen Gründen die Planungsunterstützungs-Systeme, da dieser Schritt eine gute Möglichkeit für eine computergestützte Therapie bietet.

Die Diagnoseunterstützung findet bei den unterschiedlichen Tumorarten auf verschiedene Weise eine Anwendung und verfügt auch manchmal über eine Einteilung der Tumore. Eine alleinige Klassifikation wurde vor allem für Mammakarzinome entwickelt, die auf echten Patientendatenbanken gelernt wurde [2, 5, 75]. Die Detektion von Mammakarzinomen basierte auf digitalen Mammographie-Archiven [25] oder auf den Symptomen und Werten der Patienten [44]. Bei der Diagnose von Hirntumoren lieferten Magnetresonanztomographie Aufnahmen früherer ähnlicher Patienten eine Vergleichsgrundlage [9]. Extraktion und Erkennung von bösartigen Zellen auf mikroskopischen Aufnahmen fanden bei Lymphdrüsenkrebs eine Verwendung [29]. Die Vorhersage von Rezidiven bei malignen Lebertumoren wurde mit Längsschnittstudien verschiedener Patienten bewerkstelligt [86]. Durch eine Erweiterung von CBR mit einem Rule-Based Reasoning wurde eine Frühdiagnose von Darmkrebs aufgrund von Patientendaten entworfen [98]. Für eine grundlegende Diagnose von verschiedenen Tumorarten flossen beschriebene Symptome, persönliche Informationen und klinische Daten der Patienten ein [99].

Planungsunterstützung mit CBR ist primär in der Radiologie zur Planung der Bestrahlung zu finden. Die Bestrahlungsplanung verlangt von den Fachärzten eine zeitintensive akribische Berechnung der optimalen Parameter. Eine Einschränkung der

Möglichkeiten und Empfehlung der Werte mit Hilfe von CBR bietet eine Erleichterung und Zeitersparnis. Besonders eine Arbeitsgruppe beschäftigt sich mit dieser Thematik für Gehirntumore, um die optimale Anzahl von Feldern und die dazu passenden Winkel automatisch zu bestimmen [85, 54, 59, 84]. Aus diesen Forschungen entstand auch eine Arbeit zu CBR auf fehlenden Daten [55], welche in Abschnitt 1.4.2 ausführlich besprochen wird. Weitere Arbeiten betreffen die Bestrahlungsplanung bei Schilddrüsenkarzinomen [112] und Prostatakarzinomen [71].

1.3 Fehlende Daten

Von den Pionieren der Erforschung der fehlenden Daten stammt der Ausspruch "Obviously the best way to treat missing data is not to have them" [79]. Diese humoristisch klingende Aussage fasst das ganze Ausmaß dieser Problematik gut zusammen. Sobald eine Datenbank unvollständig ist, steht man vor einem schier unlösbaren Problem.

Einerseits ist der Informationsverlust, selbst mit den besten Methoden der künstlichen Intelligenz, nicht völlig auszugleichen. Dies führt dazu, dass die Verarbeitung der Daten und die daraus resultierenden Ergebnisse unweigerlich einer Verzerrung unterlegen sind. Dabei ist es unerheblich, ob es sich nur um eine kurze statistische Auswertung oder eine tiefgehende Prognose mit Verfahren des maschinellen Lernens handelt, denn die Auswirkungen sind in jedem Fall folgenreich. Das optimale ursprüngliche Ergebnis kann nicht mehr erreicht werden, was sowohl zu einer Verminderung der Prognose eines Verfahrens führt, als auch, dass verschiedene Durchläufe verschiedene Ergebnisse liefern. Eine Interpretation und Analyse der Daten mit Hilfe einer Methode wird dadurch immens erschwert.

Andererseits gibt es keine ratenswerte Alternative zu diesem Problem, als sich der Herausforderung der fehlenden Daten zu stellen und eine Möglichkeit der akzeptablen Wiederherstellung zu finden. Auch in der medizinischen Forschung richtet sich der Fokus in den letzten Jahren verstärkt auf dieses Thema und das Bewusstsein für die Notwendigkeit dem entgegenzutreten wächst [53, 77, 120, 78, 82]. Denn, wie auch diese

Arbeit zeigen wird, führt das Ignorieren des Problems und das Löschen der betreffenden Bereiche zu fatalen Ergebnissen, die vermeidbar wären [7].

In den letzten Jahrzehnten wurden immer anspruchsvollere und komplexere Methoden entwickelt, um Lösungen für das Problem der fehlenden Daten zu finden [7, 32, 115]. Die Methoden für den Umgang mit fehlenden Daten reichen von einfachen Techniken der Löschung der betreffenden Fälle bis zu wirksamen modernen Methoden des maschinellen Lernens und der Statistik. Alle bergen verschiedene Vorteile und Nachteile und sollten bewusst für die vorliegende Fragestellung gewählt werden.

In den nächsten Abschnitten werden zuerst die Entstehung und die Gründe der fehlenden Daten betrachtet, sowie ihre Unterteilung in verschiedene Typen. Im Anschluss folgt eine Beschreibung der gängigsten Methoden mit ihren Stärken und Schwächen. Den Anfang machen die einfachen Eliminierungsverfahren, gefolgt von den Methoden der singulären Imputation, welche die fehlenden Werte substituieren, und am Ende die Multiple Imputation, als Vereinigung der Klassifikations-Algorithmen mit statistischen Verfahren.

1.3.1 Gründe und Typen von fehlenden Daten

Die Gründe für die fehlenden Daten sind vielfältig und haben die unterschiedlichsten Ursachen [115]. Eine Person möchte keine Auskunft geben und gibt keine Angaben preis. Die Datenübertragung verläuft fehlerhaft oder ein Programm erzeugt falsche Werte. Der Patient ist nicht mehr Teil einer Studie oder die Daten sind nur ein Ausschnitt aus der Gesamtdatenlage. Und zuletzt bewusste Zensur oder zufällige menschliche Fehler. Doch wie die Gründe auch sein mögen, die fehlenden Daten lassen sich immer in drei grundlegende Typen einteilen [93].

Drei Typen von fehlenden Daten

Die Unterteilung der fehlenden Daten basiert auf der existierenden oder fehlenden Abhängigkeit der Variable zu sich selbst oder zu einer anderen Variablen. Sie sind wie folgt definiert [7, 32, 115]:

- Missing Completely At Random (MCAR) bedeutet, dass die Wahrscheinlichkeit betreffs der Löschung der Daten innerhalb einer Variable zufällig ist. Sie hängt weder von der Variable selbst noch von einer anderen Variable ab. Die Verteilung der Werte der restlichen Daten in der Variable bleibt größtenteils erhalten.

- Bei Missing at Random (MAR) ist die Wahrscheinlichkeit der Löschung der Daten mit der Abhängigkeit der Variable von einer anderen Variable verbunden. Die fehlenden Daten in der Variable hängen somit direkt von der Information einer anderen Variable ab. Je nach Korrelation der Variablen untereinander übt dies einen schwachen oder starken Einfluss aus.

- Im Falle von Missing Not At Random (MNAR) hängt die Wahrscheinlichkeit der Löschung allein von der Information innerhalb der Variable selbst ab. Ganze Wertebereiche können dadurch verschwinden. Dies verursacht den größtmöglichen Informationsverlust aller Typen und eine deutliche Verzerrung der Verteilung innerhalb der Variable.

Beispiel der Auswirkungen der Typen

Die Auswirkung der drei Typen wird anhand eines künstlichen Beispiels in Tabelle 1.1 demonstriert. Die geschaffene Datenbank enthält zwölf Fälle mit den Variablen Alter, Gewicht und Augenfarbe. Die Spalten geben zum einen die Werte der Variablen an und zum anderen werden die Variablen je nach Typ der fehlenden Daten gruppiert. Zum Vergleich der Auswirkung der drei Typen auf die Verteilung der Variable werden der Mittelwert und die Standardabweichung der Variablen in Klammern aufgeführt. Das Gewicht ist eine Integer-Variable mit (78, 14.5) und die Augenfarbe eine kategoriale Variable mit den Leveln (braun, blau, grün).

Unter der zufälligen Löschung MCAR bewahrte die Augenfarbe alle ihre Level und die Verteilung des Gewichts weicht nur ein wenig ab (79.5, 15.2).

Bei MAR hängen die beiden Variablen vom Alter ab. Alle Menschen über 65 Jahren haben ihr Gewicht nicht angegeben, was auch nur zu einer leichten Abweichung der

Verteilung führte (78.3, 15.5). Die Augenfarbe haben alle Personen unter 50 Jahren nicht genannt, was die Anzahl der Level unangetastet ließ.

MNAR wirkt sich dagegen drastischer aus. Personen mit einem Gewicht über 80 verschwiegen ihr Gewicht, was die statistischen Parameter äußerst verzerrte (68, 8.6). Für die Augenfarbe ging die Information über zwei Level komplett verloren, da Träger der Farbe blau und grün diese nicht angaben.

Tabelle 1.1: Vergleich der Auswirkung der drei Typen von fehlenden Daten.

	Original		MCAR		MAR		MNAR	
Alter	Gewicht	Farbe	Gewicht	Farbe	Gewicht	Farbe	Gewicht	Farbe
40	56	blau	56	blau	56	NA	56	NA
70	60	braun	NA	NA	NA	braun	60	braun
45	64	braun	NA	braun	64	NA	64	braun
85	68	blau	68	NA	NA	blau	68	NA
50	72	braun	72	NA	72	NA	72	braun
65	76	braun	NA	NA	NA	braun	76	braun
30	80	braun	80	braun	80	NA	80	braun
35	84	braun	NA	braun	84	NA	NA	braun
80	88	braun	88	NA	NA	braun	NA	braun
60	92	grün	92	grün	92	grün	NA	NA
75	96	grün	NA	grün	NA	grün	NA	NA
55	100	blau	100	blau	100	blau	NA	NA

Feststellung des Typs

Die Kenntnis über den Typ der fehlenden Daten ist entscheidend für den richtigen Umgang mit ihnen und kann hilfreich bei der Wahl der weiteren passenden Verfahren sein. Die Bestimmung des Typs gestaltet sich jedoch als schwierig, wenn keine Informationen vorliegen, aus welchen Gründen sie gelöscht wurden. Nur für numerische Variablen mit Normalverteilung, welche unter MCAR gelöscht wurden, liegt ein Null-Hypothesentest vor, der Little's Test heißt [68, 57].

Little's Test vergleicht den Mittelwert und die Kovarianz aller Variablen mit dem Mittelwert und der Kovarianz kleiner Gruppen und Teilen der Variablen. Der Abstand aus den jeweiligen Ergebnissen wird gewichtet und zu einem Indikator der Hypothese

aufsummiert. Bei positivem Null-Hypothesentest folgt der Indikator einer χ^2-Verteilung und wird zumeist mit 0.05 als positiv anerkannt [64].

Wie viele Tests dieser Art ist auch Little's Text ein Omnibus-Test, so dass keine neuen Informationen über die Struktur der fehlenden Daten vorliegen, wenn die Null-Hypothese verworfen wurde [22]. Dies führt dazu, dass wenn schon eine der unvollständigen Variablen nicht den Typ MCAR aufweist, die Null-Hypothese verworfen wird. Besonders bei vielen unvollständigen Variablen führt dies zur häufigen Verwerfung der Null-Hypothese. Um dies zu umgehen, kann der Test auf variierenden Teilen der Datenbank durchgeführt werden, um die betreffenden unvollständigen Variablen zu bestimmen. Damit einhergeht die Gefahr wichtige Abhängigkeiten zu übersehen und eine falsche Aussage zu erhalten.

Der Hinweis, dass die Unvollständigkeit der Variable auf MCAR basiert, liefert die Möglichkeit, passende Verfahren zu Imputation hinzuzuziehen und die Ergebnisse aufgrund des geringen Informationsverlustes für verlässlicher zu erachten. Doch selbst ein negatives Ergebnis des Tests eröffnet die Chance verstärkt die Ursachen der Unvollständigkeit zu ergründen und dies für zukünftige Datenspeicherung zu vermeiden.

1.3.2 Eliminierungsverfahren

Die Eliminierung von unvollständigen Bereichen innerhalb der Datenbank, auch als Complete-Case Analysis bekannt, ist ein verbreitetes Vorgehen beim Umgang mit fehlenden Daten. Es umfasst entweder die Löschung der unvollständigen Variable oder der betroffenen Fälle (listenweiser Fallausschluss), kann aber auch nur das Auslassen der Werte mit fehlenden Daten in der Analyse (paarweiser Fallausschluss) betreffen.

Die Vorzüge dieser Methoden liegen darin, dass sie einfach und schnell in der Implementierung und Anwendung sind, da sie keine Kenntnisse über die Datenbank oder entsprechend komplexere Verfahren erfordern. Dies führt dazu, dass sie nicht nur bevorzugt von Anwender in der Analyse mit eigenen Mitteln verwendet werden, sondern auch in statistischen Toolboxen. Pakete, wie SPSS, SAS und SATA, stellen vor allem den listenweisen Fallausschluss standardmäßig zur Verarbeitung von fehlenden Daten

zur Verfügung [7]. Die potentielle Gefahr der Verzerrung der Ergebnisse und Löschung von relevanten Informationen wird dabei außer Acht gelassen.

Auslassen der Variable

Die schnellste Methode ist das Ignorieren und Auslassen der unvollständigen Variablen in der weiteren Analyse [95]. Vor allem wenn die Datenbank viele Variablen enthält oder die unvollständige Variable eine außerordentlich hohe Anzahl fehlender Daten beinhaltet, wird diese Methode zumeist angewendet. Die kritische Höhe beider Parameter unterliegt subjektiv dem Anwender.

Im Gegensatz zum listenweisen Fallausschluss hat der Typ der fehlenden Daten keinen Einfluss auf den Ansatz und das Resultat. Der Verlust der Information der ganzen Variable führt allerdings dazu, dass die Fälle der Datenbank weniger differenziert sind und das Ergebnis einer Analyse-Methode erheblich beeinträchtigt ist.

Listenweiser Fallausschluss

Der listenweise Fallausschluss überspringt die Fälle der Datenbank, welche fehlende Daten enthalten, und verwendet in der Analyse nur die vollständigen Fälle [7, 115]. Es ist ähnlich einfach in der Anwendung wie das Auslassen der Variable.

Das Verfahren erzeugt keine signifikante Verzerrung der Ergebnisse, wenn die Datenbank genug Fälle enthält, der Anteil der Fälle mit fehlenden Daten nicht zu groß und die Annahme der MCAR Hypothese (1.3.1) für die gesamte Datenbank erfüllt ist. Sind diese Bedingungen nicht gegeben, vor allem was die MCAR Hypothese anbelangt, sind selbst statistische Parameter wie der Mittelwert äußerst verfälscht und das Verfahren nicht zu empfehlen.

Paarweiser Fallausschluss

Der paarweise Fallausschluss betrachtet nur die vollständigen Werte der Fälle in der Datenbank für der Analyse und lässt die fehlenden Werte aus [7, 115]. Während der Analyse schrumpfen die Fälle sozusagen nur ihre vollständigen Werte zusammen.

Das Verfahren nutzt im Gegensatz zum listenweisen Fallausschluss alle Fälle der Datenbank und bewahrt die gesamte Information aller vollständigen Werte. Dadurch bleiben die statistischen Parameter der vorliegenden unvollständigen Datenbank erhalten, was auch die Kovarianz betrifft. Das Verfahren zeigt sich robuster gegenüber den drei Typen von fehlenden Daten. Die Analyse wird jedoch ineffizient, wenn der Anteil der fehlenden Daten innerhalb der Fälle hoch ist. Außerdem weisen die Fälle und Variablen unterschiedliche Größen auf, was zu Problemen bei der Anwendung und Interpretation einer Analyse führen kann.

1.3.3 Singuläre Imputation

Um die angesprochenen Nachteile der Eliminierungsverfahren zu überwinden liefert die Substitution oder Imputation der fehlenden Werte in der Datenbank die Möglichkeit auf die spezifische Struktur der Datenbank und auch die ihrer unvollständigen Werte einzugehen. Das Muster und die darin liegenden Eigenschaften der fehlenden Werte wurden bisher nicht bei den Eliminierungsverfahren in Betracht gezogen.

Substitution mit festgesetzten Werten

Zum einen kann diese Ersetzung durch eine einfache Substitution stattfinden, welche sich zumeist auf die statistischen Parameter der entsprechenden Variable gründet. Gewöhnlich findet die Ersetzung der fehlenden Werte einer Variable mit deren Mittelwert (Mean) statt [58]. Der Gedanke dahinter ist es für eine Variable, die einer Normalverteilung unterliegt, eine vernünftige Schätzung der Werte zu finden. Diese Annahme trifft bei vielen Datenbanken zu, die auf natürlichen Prozessen beruhen. Ein Test, ob die Variable der Normalverteilung folgt, kann hilfreich sein [57].

Die Mittelwertersetzung ermöglicht es alle Daten einer unvollständigen Datenbank zu nutzen. Allerdings fügt die Methode der Datenbank keine neuen Informationen hinzu und erhöht nur die Menge an verwertbaren Daten. Dies führt dazu, dass die Standardabweichung und damit der Standardfehler, unterschätzt werden. Wenn die

Verteilung der Variable nicht normal ist oder der Typ der fehlenden Daten nicht MCAR, werden alle statistischen Parameter verzerrt.

Es bieten sich für die Substitution der fehlenden Daten mit einem festen Wert auch andere Optionen an. Für kategorische Variablen wird der am häufigste vorkommende Wert (MODE) anstelle des Mittelwertes verwendet [6]. Es gibt auch andere Ansätze, die das Maximum oder Minimum der Variable verwenden, oder einfach einen selbst definierten festen Wert [32]. Diese bringen jedoch nicht die statistischen Vorteile einer Mittelwertersetzung mit sich.

Imputation mit Klassifikations-Algorithmen

Zum anderen bieten moderne Klassifikations-Algorithmen die Fähigkeit individuell angepasste Werte für jeden einzelnen fehlenden Wert zu erzeugen und dabei die gesamte Information der Datenbank möglichst auszuschöpfen. Bekannte Klassifikations-Algorithmen des maschinellen Lernens sind Nächste-Nachbarn-Klassifikation, Support-Vektor-Maschinen, Neuronale Netzwerke und baumbasierte Methoden [3]. Aufgrund der Größe und Beschaffenheit der medizinischen Datenbanken, mit denen diese Arbeit sich beschäftigt, eigenen sich für die Imputation baumbasierte Methoden [77, 78].

Classification and Regression Tree (CART), ist ein gerichteter, geordneter und kreisfreier Graph, ein sogenannter gewurzelter Entscheidungsbaum, welcher aus Knoten und Kanten besteht [24, 40]. Der Baum fällt die Entscheidung rekursiv in jedem Schritt (Knoten), wie die Daten auf Grundlage einer Metrik am besten zu teilen und damit zu klassifizieren sind. Als Metriken werden der Gini-Index oder ein Entropie-Maß verwendet. Diese berechnen, welche Wertebereiche in welcher Variable den größten momentanen Effekt auf die Gesamtheit der Daten haben, um eine optimale Klassifizierung zu gewährleisten. Auf den so entstandenen Teilmengen wird der Prozess wiederholt, bis der Baum so viele Knoten in jedem seiner Äste erzeugt hat, wie es für eine eindeutige Klassifizierung bedarf. Dies kann bedeuten, dass manche Äste bereits früher andere

später abgeschlossen werden. Die Größe eines Baumes und der Elemente in den Teilmengen werden durch Parameter am Anfang festgesetzt.

Zur Imputation von fehlenden Daten wird der Baum auf der unvollständigen Datenbank gelernt und erschafft eine Klassifikation dieser Daten ohne die fehlenden Werte selbst zu klassifizieren. Die fehlenden Werte werden aufgrund der Vorhersage des Baumes für ihre wahrscheinlichsten Werte mit diesen ersetzt.

Die diskrete Struktur der Entscheidungsbäume lässt eine einfache Interpretation ihrer Ergebnisse zu. Durch die Einbeziehung aller Variablen bewahren sie die statistische Struktur der Datenbank und können außerdem sowohl numerische als auch kategoriale Variablen verarbeiten.

Allerdings ist die Qualität der Ergebnisse empfindlich betreffs der Beschaffenheit der Datenbank. Eine kleine Anzahl von Fällen oder nicht entsprechend differenzierten Fällen beeinflusst die korrekte Vorhersage außerordentlich. Die nicht festgelegte Größe eines Baumes kann leicht zu einer Überanpassung an die vorliegenden Daten führen. Pruning-Methoden zum Beschneiden und Verkürzen der Bäume werden in diesem Falle angewandt. Das Setzten der richtigen Parameter sollte durch eine Fehlerrate auf einer Trainingsdatenbank bestimmt werden.

Random Forest ist ein Wald aus unkorrelierten Entscheidungsbäumen, welche mit Hilfe einer Randomisierung erzeugt wurden [23, 111]. Die finale Klassifizierung entsteht aus dem Mehrheitsergebnis der unterschiedlichen Bäume. Alle Bäume des Random Forest werden vor der Klassifizierung einer individuellen Variablenreduktion unterzogen, welche zu einer Randomisierung führt. Das heißt, jeder Baum nutzt nur $m < n$ Variablen zur Klassifizierung, wobei n die Anzahl aller Variablen der Datenbank ist und für jeden Baum die m Variablen permutieren. Da m meist eine kleine Zahl ist, werden die einzelnen Bäume nicht beschnitten. Die Anzahl der verwendeten Bäume und m müssen definiert werden.

Die Imputation der fehlenden Werte erfolgt wie bei CART aus der Vorhersage der finalen Klassifikation mit den bestimmten Werten für die fehlenden Daten.

Wie zuvor CART bewahrt der Random Forest die Verteilung der imputierten Variablen. Im Gegensatz zu diesem ist er jedoch durch die verkleinerten Bäume robuster gegen Überanpassung. Außerdem sind die Ergebnisse als Durchschnitt aus mehreren Bäumen präziser und stabiler, da sie durch die Randomisierung die Variablen mit der größten Wirkung auf die Klassifikation besser herausarbeiten. Seine volles Potential entfaltet der Random Forest allerdings auf großen Datenmengen und ist wie CART empfindlich gegenüber schlecht strukturierten Datenbanken. Der erhöhte Zeitfaktor durch die hohe Anzahl von Bäumen kann durch eine Parallelisierung umgangen werden.

1.3.4 Multiple Imputation

Die Multiple Imputation (MI) erweitert die singuläre Imputation um einem statistischem Ansatz, welcher die Unsicherheit bezüglich der imputierten Werte zu seinem Vorteil nutzt [32]. Auf Basis dieser Unsicherheit werden mehrere unterschiedliche imputierte Datenbanken erzeugt und die Ergebnisse der Analysen auf diesen vollständigen Datenbanken gemittelt. Die Multiple Imputation birgt die Möglichkeit die Stärken der Klassifikations-Algorithmen aus der singulären Imputation zu bewahren und ihre Schwächen auszubessern. Dies tut sie, in dem sie als eine Art Blackbox agiert und jeden Imputations-Algorithmus in sich aufnehmen kann [115]. Dabei ist zu beachten, dass MI im Gegensatz zu singulären Imputation-Methoden keine dauerhafte Imputation durchführt, sondern nur innerhalb der Analyse die Werte der verschiedenen Imputationen nutzt und danach verwirft. Der Vorteil für den Anwender ist, die Ergebnisse der Analyse zu verwenden und nicht die möglicherweise falsch imputierten Werte.

Die Methode wurde in den achtziger Jahren entworfen [94] und im letzten Jahrzehnt für Datenbanken mit mehreren unvollständigen Variablen weiterentwickelt [116]. Seit dieser Zeit hat MI sich als Verfahren für den Umgang mit fehlenden Daten etabliert [62, 65, 81, 52]. Auch in der Medizin hat sie mittlerweile Einzug gehalten und es wurden umfassende Empfehlungen für die passenden Einsatzbereiche erörtert [108, 82, 33].

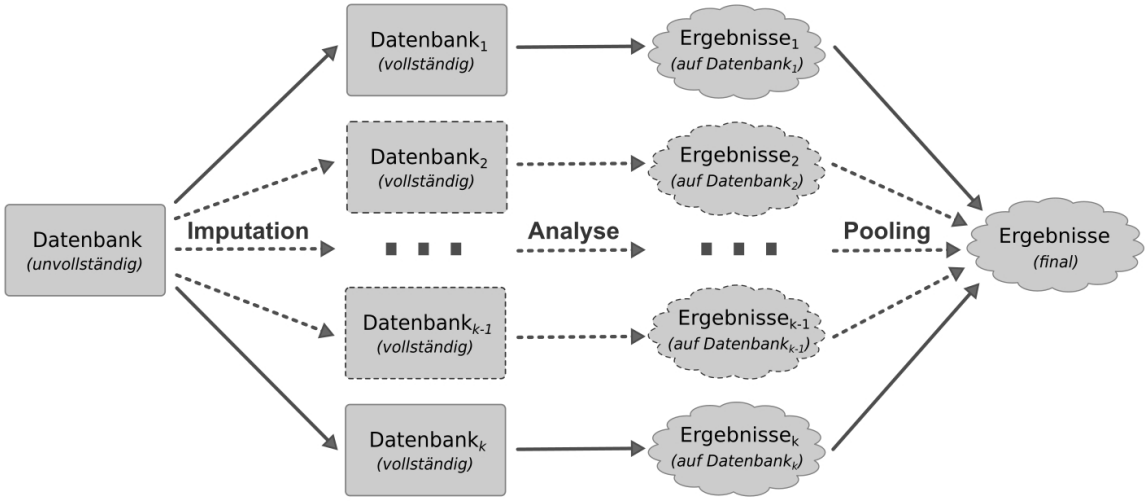


Abbildung 1.2: Die drei Schritte der Multiple Imputation.

Ablauf des Verfahrens

Die Multiple Imputation besteht aus drei Schritten, welche die Imputation der unvollständigen Datenbank, die Anwendung einer Analyse-Methode auf diesen und die Vereinigung der Ergebnisse beinhalten [115]. Eine schematische Darstellung dieser Schritte und ihrer Resultate findet sich in Abbildung 1.2.

Schritt 1 – Imputation: Aus der ursprünglichen unvollständigen Datenbank werden $k > 1$ vollständige Datenbanken mit imputierten Werten für die fehlenden Daten erzeugt. Die Imputation erfolgt mit einem Klassifikations-Algorithmus, wie CART oder Random Forest [105, 31]. Allerdings werden die entsprechenden analogen imputierten Werte jeder dieser Datenbanken mit Hilfe eines Verteilungsmodells generiert, welche individuell aufgrund der mehrdimensionalen Verteilung der Datenbank auf die fehlenden Werte angepasst ist. Aufgrund dessen unterscheiden sich die k imputierten vollständigen Datenbanken alle leicht in ihren imputierten Werten, aber sind identisch in den vollständigen Werten der ursprünglichen unvollständigen Datenbank. Die Größenordnung dieser feinen Unterschiede spiegelt die Unsicherheit betreffs der korrekten Imputation wieder. Für jede neue Datenlage ändert sich diese Größenordnung und passt sich an.

Schritt 2 – Analyse: Die gewünschte Analyse-Methode wird auf allen k imputierten vollständigen Datenbank separat ausgeführt. Dies steht im Kontrast zu der singulären Imputation oder den Eliminierungsverfahren, bei denen die Analyse-Methode nur auf einer Datenbank ausgeführt wird. Die Ergebnisse der Analyse-Methode auf den Datenbanken unterscheiden sich alle ein wenig wegen Schritt 1 und behalten somit die Unsicherheit betreffs der Imputation bei.

Schritt 3 – Pooling: Die verschiedenen Ergebnisse der Analyse-Methode auf den k imputierten vollständigen Datenbanken aus Schritt 2 werden gemittelt und als Durchschnitt in einem Ergebnis vereinigt. Unter den passenden Bedingungen sind die Ergebnisse unverzerrt und haben die richtigen statistischen Eigenschaften.

Herleitung aus dem Satz von Bayes

Der kritische Teil der Multiple Imputation ist die Erzeugung von unterschiedlich imputierten Werte der selben fehlenden Werte der Ursprungsdatenbank in den k imputierten vollständigen Datenbanken. Dies geschieht mit Hilfe einer an die fehlenden Daten angepassten Wahrscheinlichkeitsverteilung. Die Beschreibung des Vorgangs folgt dem neuesten Standardwerk über Multiple Imputation [115].

Die ursprüngliche unvollständige Datenbank D besteht aus den vollständigen Werten D_{voll} und den fehlenden Werten D_{fehl}. Aus dem Satz von Bayes ergibt sich aus den bedingten Wahrscheinlichkeit dieser beiden [56]:

$$P(D_{fehl}|D_{voll}) = \frac{P(D_{voll}|D_{fehl})P(D_{fehl})}{P(D_{voll})}$$

Gesucht ist eine Schätzung der relevanten statistischen Parameter E der hypothetischen vollständigen Datenbank, um diese ideal zu beschreiben und dadurch zu rekonstruieren. E ist unbekannt, da D_{fehl} unbekannt ist. Die Größe der Unsicherheit betreffs E hängt direkt von dem Wissen über D_{fehl} ab. Wenn D_{fehl} sich völlig wiederherstellen ließe, wäre auch die Korrektheit der geschätzten E gewährleistet. Da dies nicht erreich-

bar ist, kann die Verteilung von E nur durch plausible Variationen der möglichen D_{fehl} zusammengefasst werden.

Die a-posteriori-Verteilung $P(E|D_{voll})$ erfasst die möglichen Werte von E bei Kenntnis von D_{voll}. Da diese meist schwierig zu lösen ist, wird $P(E|D_{voll})$ in eine Gleichung mit zwei einfacher zu handhabenden Teilen aufgespalten:

$$P(E|D_{voll}) = \int P(E|D_{voll}, D_{fehl})P(D_{fehl}|D_{voll})dD_{fehl}$$

Dabei ist $P(E|D_{voll}, D_{fehl})$ die a-posteriori-Verteilung von E in der hypothetischen vollständigen Datenbank $D = (D_{voll}, D_{fehl})$ und $P(D_{fehl}|D_{voll})$ ist die a-posteriori-Verteilung der fehlenden Daten D_{fehl} aus den bekannten vollständigen Daten D_{voll}. Die Gleichung zeigt, dass die tatsächliche a-posteriori-Verteilung von E gleich dem Durchschnitt über die wiederholten Berechnungen von E ist.

Der Ablauf zur Lösung der Gleichung zu $P(E|D_{voll})$ ist wie folgend. Mit Hilfe des Satzes von Bayes wird aus $P(D_{fehl}|D_{voll})$ eine Imputation von D_{fehl} erzeugt, benannt mit $\tilde{D}_{fehl}$. Dann werden die statistischen Parameter E aus der hypothetischen vollständigen Datenbank $(D_{voll}, \tilde{D}_{fehl})$ mit $P(E|D_{voll}, \tilde{D}_{fehl})$ berechnet. Diese beiden Schritte werden iterativ bis zu einem festgelegten Ende wiederholt. Auf diese Weise kann $P(E|D_{voll})$ iterativ durch zwei a-posteriori-Verteilungen erzeugt werden, welche leicht berechnet werden können.

Mehrdimensionale fehlende Daten

Für den Imputations-Schritt der Multiple Imputation existiert noch eine Erweiterung für p unvollständige Variablen, die Fully Conditional Specification (FCS) [118]. Die Idee dahinter ist es, ein p-dimensionales Problem in p eindimensionale Probleme aufzuteilen.

Die FCS imputiert mehrdimensionale fehlende Daten iterativ mit Hilfe einer bedingten Wahrscheinlichkeitsverteilung. Dies geschieht am Anfang durch eine Schätzung der Verteilung der imputierten Werten, welche im letzten Abschnitt beschrieben wurde. Danach werden die geschätzten vorherigen Werte in den folgenden Iterationen als

Grundlage für die weiteren Schätzungen verwendet. Die Verarbeitung erfolgt variablenweise, so dass die Reihenfolge der zu verarbeitenden unvollständigen Variablen von Bedeutung sein kann.

Vor- und Nachteile

Die Multiple Imputation bietet zum einen alle Vorteile des Klassifikations-Algorithmus, welchen sie für die Imputation der fehlenden Daten verwendet. Darüber hinaus bezieht sie die Unsicherheit betreffs der imputierten Werte mit ein und bedenkt, welchen Wertebereich die wahren Daten hätten einnehmen können. Dies führt zu wesentlich verbesserten Ergebnissen als alle vorherigen Verfahren, da MI die mehrdimensionale Verteilung der Datenbank möglichst optimal nutzt und letztendlich bewahrt. MI ist somit nicht nur für MCAR und MAR geeignet, sondern auch aufgrund ihrer statistischen Eigenschaften robuster gegenüber MNAR als die anderen Methoden.

Allerdings bedarf MI als relativ komplexes Verfahren eines Verständnisses des Ansatzes und eine Einbettung in die spezifische Fragestellung und Analyse, um die gewünschten Erwartungen betreffs der Ergebnisse zu erfüllen. Auch ist sie durch ihren verwendeten Klassifikations-Algorithmus und der Wahrscheinlichkeitsverteilung der imputierten Werte abhängig von der Beschaffenheit der Datenbank. Des Weiteren können falsch gesetzte Parameter, wie die Anzahl der imputierten Datenbanken oder Iterationen, das Ergebnis schmälern. Die k Datenbanken sorgen sowohl im Imputations-Schritt, als auch in der Analyse für eine erhöhte Zeitkomplexität. Dieser kann wieder teilweise durch Parallelisierung entgegengewirkt werden.

1.4 CBR im Kontext von fehlenden Daten

Case-based Reasoning hängt als Entscheidungsunterstützungssystem maßgeblich von der Datenbank ab, welches es als Grundlage für seine Berechnungen nimmt und aufgrund dessen es seine algorithmische Entscheidung trifft. Die Retrieve-Phase ist der kritischste Punkt innerhalb des CBR Zyklus, da ihre Ähnlichkeitsfunktionen direkt

den Inhalt der Datenbank verwenden und alle weiteren Phasen ihr Ergebnis weiterverarbeiten. Daher ist die Retrieve-Phase sehr anfällig gegenüber Fehlern in der Datenbank und muss mit äußerstem Bedacht ausgeführt werden, da sonst die finale Lösung des gesamten CBR in Zweifel gezogen werden muss. Allerdings enthalten die meisten Datenbanken, so wie auch MOSAIQ selbst, von Natur aus fehlende Daten in unterschiedlichsten Arten und Anteilen. Es wurde vor einigen Jahren gezeigt, dass sich in der Retrieve-Phase die Anzahl der tatsächlich relevanten Fälle in Bezug auf den Zielfall durch ein Anwachsen fehlender Daten verschlechtert und darunter die Qualität der enthaltenen Information leidet [73].

In den letzten Jahrzehnten haben sich jedoch nur wenige Arbeiten mit dem Verhalten von CBR unter dem Einfluss von fehlenden Daten beschäftigt und sich einer Lösung dieser Problematik angenommen. Die meisten Arbeiten weisen nur einen speziellen Fokus auf und fordern Bedingungen an das System oder die Datenbank, was ihre Anwendungsfelder erheblich einschränkt. Nur einer dieser Ansätze führt zu universellen Methoden, die in jedes beliebige CBR System eingesetzt werden können, ohne den Algorithmus tiefgreifend zu verändern, und unabhängig von der Art der Variablen sind. In den wenigsten Arbeiten ist der Aspekt der fehlenden Daten der Hauptfokus, sondern nur ein Teilstück von vielen. Aus diesem Grunde fehlt eine vergleichende Analyse zu den untereinander konkurrierenden Methoden in den meisten Fällen. Eine systematische Übersicht der bisherigen Ansätze und eine vergleichende Analyse der verschiedenen Methoden ist nicht vorhanden. Nur die universellen Methoden eignen sich zu einem Vergleich ohne Einschränkungen und wurden dafür in der Evaluation dieser Arbeit herangezogen.

In den folgenden Abschnitten werden die unterschiedlichen Ansätze vorgestellt und in universelle und spezielle Methoden aufgeteilt.

In diesem Kapitel und allen weiteren ist eine lokale Ähnlichkeitsfunktion mit $sim(x, y)$ zwischen den Werten x und y zweier Fälle C_x und C_y im Intervall $[0, 1]$ definiert. Wenn x und y identisch sind gilt $sim(x, y) = 1$, wenn sie völlig verschieden sind $sim(x, y) = 0$.

Für kategoriale Variablen existieren nur diese beiden Zustände, für numerische Variablen werden auch alle Werte dazwischen eingenommen.

1.4.1 Universelle Methoden

Der einzige bisherige Ansatz für Case-based Reasoning der universell anwendbar ist, besteht darin die fehlenden Daten zu tolerieren und die Ähnlichkeitswerte der fehlenden Daten zu substituieren. Ein festgelegter Wert wird für die entsprechende lokale Ähnlichkeit eingesetzt und ist damit unabhängig von der Art der Variable oder der Struktur der Datenbank. Es gibt zwei verschiedene Auffassungen für die Wahl des Wertes, die beide zeitnah nach der Einführung des klassischen CBR untersucht wurden [1]. Allerdings lag der Fokus der Arbeiten nicht auf der Lösung des Problems der fehlenden Daten, sondern der Entwicklung eines CBR Systems, so dass keine Analyse der Wirksamkeit der Ansätze durchgeführt wurde.

Der eine Ansatz definiert die lokale Ähnlichkeit mit $sim(x, y) = 0.5$, wenn x oder y unbekannt sind, und wurde für ein allgemeines CBR mit euklidischer Metrik erstellt [91]. Er spiegelt die Unsicherheit aufgrund des Nichtwissens wieder und versucht keinen Ausschlag in eine Richtung zu geben, indem er den mittleren Wert der Ähnlichkeitsfunktion benutzt.

Der andere Ansatz nutzt eine lokale Ähnlichkeit von $sim(x, y) = 0$ für das Fehlen von x oder y und war Teil eines CBR Systems mit einer Erweiterung durch regelbasiertes Schließen [110]. Später wurde die Methode nochmals für ein dialog-basiertes CBR aufgegriffen [74], welches aufgrund der Fragen des Anwenders angepasst wird. Nach der Definition von 1.4 bedeutet $sim(x, y) = 0$, dass die Werte x und y völlig unterschiedlich sind, was somit einen Fall mit fehlenden Daten hart bestraft, da er in der Liste der ähnlichsten Fälle in Bezug auf einen Zielfall weit zurückfällt.

Beide Ansätze sind mit den singulären Substitutionsmethoden aus Abschnitt 1.3.3 verwandt, nur, dass sie die Ähnlichkeitswerte der fehlenden Daten ersetzen und nicht die fehlenden Daten an sich. Außerdem sind die Werte und gehen somit nicht auf die statistischen Eigenschaften der betreffenden Variable ein.

1.4.2 Spezielle Methoden

Alle weiteren Arbeiten, welche sich mit Case-based Reasoning im Kontext von fehlenden Daten beschäftigen, haben ihre Methoden für ein spezielles Problem formuliert und fordern Vorbedingungen, was sich als Beschränkungen auf das System oder die Datenbank auswirkt. Die Ideen und die Herangehensweise sind dabei sehr unterschiedlich und werden nach ihrer zeitlichen Entwicklung aufgeführt.

Kurz nach der Etablierung von CBR wurde eine Ähnlichkeitsfunktion mit veränderlicher lokaler Ähnlichkeit eingeführt, das sich auf (nominalskalierte) kategoriale Variablen beschränkte und sich an die Eigenschaften jeder Variable anpasste. Für ein fehlendes x oder y wurde die Ähnlichkeitsfunktion mit $sim(x, y) = 1 - (1/L * (1 - 1/L))$ definiert, wobei L die Anzahl der Level der entsprechenden kategorialen Variablen ist [4]. Die Methode wurde für einen Hybrid aus CBR und regelbasiertes Schließen entwickelt und mit den beiden alleinstehenden Verfahren auf Tumordatenbanken verglichen.

In nicht reduzierbaren Datenbanken gibt es jeden Fall nur genau einmal als Repräsentanten einer Klasse. Für diese Art von Datenbank wurde ein Algorithmus entwickelt, welcher auf einer Verbindung aus Klassifikation mit Entscheidungsbaum und zusätzlicher Gewichtung durch Nächste-Nachbarn Metrik basierte [73]. Aufgrund der eindeutigen Klassifikation des Entscheidungsbaumes addiert das System Punkte bei Ähnlichkeit, subtrahiert beim Gegenteil und vergibt keine Punkte für fehlende Werte. Zur Evaluation wurde ein selbst eingeführtes Maß für die Genauigkeit verwendet und mit den beiden einzelnen Methoden des Algorithmus verglichen.

Auf Basis des Wissens einer Datenbank einer Längsschnittstudie wurde CBR selbst verwendet, um die fehlenden Daten in der Adaptions-Phase wiederherzustellen. Aus den verschiedenen Zeitpunkten der Datenbank und dem Fachwissen eines Experten wurden die Daten entweder exakt oder nur geschätzt rekonstruiert [119, 103]. Das System wurde als Anwendung genutzt, um die Auswirkungen von Fitness auf Dialysepatienten zu überprüfen. Ein Vergleich mit anderen Verfahren fand nicht statt.

Ein ähnlicher Ansatz wurde mit der Nächsten-Nachbar-Klassifikation verwendet, so dass die fehlenden Daten aus der Gesamtheit der restlichen Fälle herausgefiltert und

ersetzt wurden [55]. Jedoch nur für numerische Variablen und mit Hilfe der enthaltenen Fälle ohne fehlende Daten. Um der Unsicherheit der richtigen Werte für die fehlenden Daten gerecht zu werden, wurden die globalen Ähnlichkeiten mit dem Fehler ihrer möglichen korrekten Imputation bestraft. Der Fehler wurde offline durch zufällige Löschung der vorhandenen Fälle berechnet. Die Methode bestrafte direkt den Rang der ähnlichsten Fälle zum Zielfall, wenn sie fehlende Daten enthielten. Das System war Teil eines bereits existierenden CBR, dass als Entscheidungsunterstützung der Bestrahlungsplanung für Prostatakarzinome diente. Eine Evaluation der Methode fand nur zum Nachweis des Herabstufens der Fälle statt und für den Typ der fehlenden Daten MCAR.

Der Begriff des Grades der fehlenden Daten wurde für ein CBR im Bereich der Taifun Analyse und Notfallwarnsysteme eingeführt [124]. Das System benutzte Fuzzylogik, um sprachliche Begriffe in Intervallen darzustellen. Allerdings waren die Ausführungen äußerst spärlich und der Begriff wurde nicht vertieft, so dass eine Beurteilung und ein Vergleich nicht möglich sind.

Der neueste Ansatz wurde für die Entwicklung von Wassergeneratoren entworfen und umgeht das Problem der Berechnung der fehlenden Daten [122]. Die Datenbank beruht auf einer Ontologie und somit auf Verknüpfung der sprachlichen Begriffe mit ihren Eigenschaften. Das System verwendet ein auf einem Entscheidungsbaum basierendes Retrieval und nutzt eine abgewandelte Levenshtein-Distanz für Graphen um den Abstand zwischen den verschiedenen Ecken über ihre Kanten zu bestimmen. Somit überbrückt es die fehlenden Daten und kann auch Fälle mit unterschiedlicher Variablen Anzahl zueinander in Verbindung setzt. Das System hat im Vergleich zu den universellen Methoden (1.4.1) keine Verbesserung gezeigt.

2 Material und Methoden

In diesem Kapitel wird in Abschnitt 2.1 die Idee und der Algorithmus des Multiple Retrieval Case-based Reasoning vorgestellt, welches im Rahmen dieser Arbeit entwickelt wurde, um eine effiziente und zuverlässige Lösung für die Ergebnisse von Case-based Reasoning im Umfeld von unvollständigen Datenbanken zu ermöglichen.

Für den Nachweis dieser Methodik erläutert der Abschnitt 2.2 detailliert den praktischen Aufbau und die theoretischen Grundlagen der Evaluation, deren Ergebnisse in Kapitel 3 unter verschiedenen Voraussetzungen präsentiert werden.

2.1 Multiple Retrieval Case-based Reasoning

Nach den Erläuterungen des letzten Kapitels über den Umgang mit fehlenden Daten und die Lösungen für Case-based Reasoning im Kontext dieser, wird nun der Kern dieser Arbeit, das Multiple Retrieval Case-based Reasoning (MRCBR), vorgestellt.

Begonnen wird mit einer Motivation für die Entwicklung dieses Verfahren, welche sich auf dem Wissen und den Erfahrungen des letzten Kapitels gründet. Danach folgt eine theoretische Darstellung der Methodik, welche ihrer Abrundung in der detaillierten Erklärung des Algorithmus von MRCBR findet.

2.1.1 Motivation

Nach den Erfahrungen in der klinischen Routine sind die Eliminierungsverfahren (1.3.2) für die Verwendung von Case-based Reasoning mit zugrundeliegenden unvollständigen Datenbanken das Mittel der Wahl. Dies bedeutet letztendlich die Löschung oder Aus-

lassung der betreffenden Daten, Fälle oder sogar Variablen und damit einen enormen Verlust an Information. Wie bereits erwähnt wurde, sind die Eliminierungsverfahren selbst bei kommerziellen statistischen Programmen als Standard gegeben.

Im letzten Abschnitt 1.4 wurde ersichtlich, dass nur wenige Arbeiten versucht haben diesen Umstand zu ändern und sich mit der Thematik von CBR im Kontext von fehlenden Daten auseinandergesetzt haben. Bisher wurden die Vorteile und Leistungsfähigkeit der in Abschnitt 1.3 vorgestellten modernen Imputations-Verfahren in keiner dieser Arbeiten für CBR genutzt. Im Weiteren fordern alle Ansätze bis auf die universellen Methoden Bedingungen an die Anwendung. Dies reicht von Einschränkungen auf den Typ der Datenbank über die Art der zu verarbeitenden Variable bis zu einem festgelegten Retrieval-Algorithmus. Vor allem Letzteres nimmt damit dem Anwender die Möglichkeit einer eigenen Ähnlichkeitsfunktion in der Retrieve-Phase, die an das vorliegende Problem optimal angepasst ist. Die Evaluationen der Methoden waren meist mangelhaft, da der Fokus nicht auf den fehlenden Daten lag, und eine Studie auf großen Datenbanken fehlte. Auch die Auswirkungen der verschiedenen Typen fehlenden Daten MCAR, MAR und MNAR auf das Ergebnis des CBR und die Rangfolge der ähnlichsten Fälle nach dem Retrieval wurden nicht erforscht.

Dies führt dazu, dass es für CBR im Kontext von fehlenden Daten bis heute keine verlässliche Lösung und einen Nachweis dieser gibt. Eine korrekte Rangfolge der Fälle nach dem Retrieval ist dadurch nicht gesichert und damit basiert auch die Weiterverarbeitung dieser Fälle in den darauffolgenden Phasen für Prognosen und Resultate auf keiner gesicherten Grundlage, da sie direkt von der Retrieve-Phase abhängen. Somit vermeiden daher fast alle entwickelten CBR-Systeme die Konfrontation mit fehlenden Daten oder wenden nur die eben genannten klassischen Methoden der Löschung an. Es entsteht ein Teufelskreis an Verlust von Information und unsicheren kaum verifizierten Ergebnissen.

Gerade im Hinblick auf die Möglichkeiten und Herausforderungen von Big Data bedarf es einer effektiven Lösung dieses Problems, um das volle Potential der Daten zu nutzen. Dies schließt nicht nur die Information der vollständigen Daten, sondern auch

die Ableitung des Wissens aus der eingebetteten Information der fehlenden Daten mit ein. Keine der bisherigen Arbeiten hat jedoch die intrinsischen statistischen Eigenschaften der unvollständigen Datenbanken mit Hilfe von modernen Imputations-Methoden verwendet, um eine stabile und verlässliche Lösung in der Retrieve-Phase und aller davon abhängenden nachfolgenden Phasen zu gewährleisten.

Die Fragestellung hat eine große Bedeutung für das Entscheidungsunterstützungssystem für das Tumorboard (CBR-TDS), da es seine Vorhersagen direkt aus MOSAIQ bezieht, welches, wie bereits erwähnt, in vielen Variablen äußerst unvollständig ist. Dies bringt eine große Unsicherheit in der Verlässlichkeit der Präsentation der ähnlichsten Fälle mit sich. Eine möglichst korrekte Auswahl der ähnlichsten Fälle ist wiederum ausschlaggebend für das Vertrauen in eine Systemvorhersage, welche die Fachärzte in ihre Entscheidung einbeziehen können, um die betreffenden Patienten bestmöglich zu therapieren. Es ist somit unabdingbar dies zu garantieren.

Um die aufgeführten Anforderungen zu erfüllen und die offenen Fragestellungen zu beantworten wurde das Multiple Retrieval Case-based Reasoning (MRCBR) entwickelt und vom Autor dieser Arbeit im Journal of Biomedical Informatics veröffentlicht [69]. Das MRCBR ist universell anwendbar, erfordert keine Vorbedingungen an die Art der Variable oder Datenbank, und ist für jeden Typ von fehlenden Daten passend. Obwohl das MRCBR für das CBR-TDS entworfen wurde, ist es unabhängig von der Wahl des CBR Systems und kann ohne Einschränkungen als Grundlage für jedes CBR System verwendet werden.

2.1.2 Idee und Aufbau

Diese Arbeit kombiniert die Stärken von maschinellem Lernen und Statistik um die angesprochene Problematik von Case-based Reasoning im Kontext von fehlenden Daten zu lösen. Der Ansatz des Multiple Retrieval Case-based Reasoning (MRCBR) basiert auf der Verschmelzung von Multiple Imputation, Klassifikations-Algorithmen und der Retrieve-Phase, welche zusammen einen neuen erweiterten Teil des lernenden Kreislaufs von CBR bilden.

Der Gedanke dahinter ist die intrinsischen statistischen Eigenschaften der Datenbank möglichst tiefgreifend zu nutzen. Sowohl die vollständigen Fälle, als auch das Wissen um die Verteilung der fehlenden Daten innerhalb der Datenbank, bieten wertvolle Informationen, die ansonsten für das CBR verloren gehen. Im Weiteren verbessert sich die Lösung bei jeder Anwendung aufgrund der neuen Informationen des aufgenommen Zielfalls. Im Vergleich zu allen bisherigen Methoden ist dadurch das Retrieval weniger anfällig gegenüber der Verzerrung von fehlenden Daten jeden Typs und bietet eine statistisch fundierte Lösung für die korrekte Rangfolge der gefundenen ähnlichen Fälle.

MRCBR Aufbau

Das MRCBR ist ein automatisches System, nur das Einlesen der Datenbank und des Zielfalls sind vom Nutzer abhängig. Es unterteilt sich in einen Offline-Teil, der die fehlenden Daten aufbereitet und verarbeitet, und einen Online-Teil, der das eigentliche CBR mit seinen vier Phasen darstellt. Beide können getrennt trainiert und genutzt werden, damit sich das System im Vornherein auf die Fragestellung einstellen und dann direkt vom Anwender ohne Wartezeit bedient werden kann.

Das Verfahren hat keine Einschränkungen oder Vorbedingungen an die Art der Anwendbarkeit oder Datenbank. Sowohl numerische und als auch kategoriale Variablen können simultan verarbeitet werden. Die Methode ist aufgrund seiner statistischen Methodik robust gegenüber jedem Typ von fehlenden Daten. Der Kreislauf des MRCBR ist in Abbildung 2.1 dargestellt. Der detaillierte Ablauf des Algorithmus des Verfahrens wird im nächsten Abschnitt 2.1.3 genauer betrachtet.

MRCBR - Offline

Im Offline-Teil wird eine vorher definierte Anzahl von Kopien der zugrundeliegenden unvollständigen Datenbank gemacht und jede dieser Kopien wird einer Imputation unterzogen, so dass sie vollständig sind. Die Imputation erfolgt mit Hilfe eines modernen Klassifikations-Algorithmus, wie sie in Abschnitt 1.3.3 beschrieben wurden.

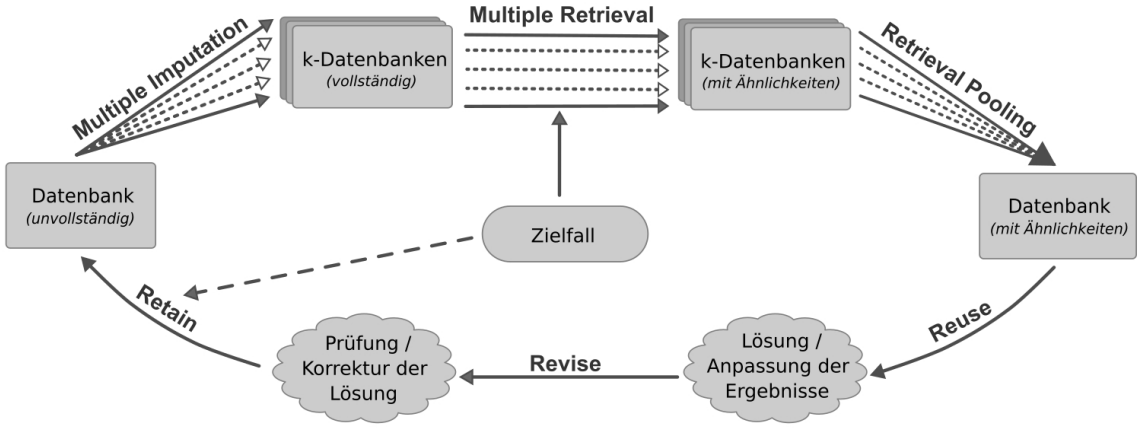


Abbildung 2.1: Kreislauf des Multiple Retrieval Case-based Reasoning mit seinen Phasen.

Gemäß dem Verfahren der Multiple Imputation unterscheiden sich die imputierten vollständigen Datenbanken alle ein wenig in ihren imputierten Werten, um die Unsicherheit bezüglich der wahren Imputation widerzuspiegeln. Diese Variation der imputierten Werte wird aufgrund einer Wahrscheinlichkeitsverteilung ihrer möglichen Werte in Bezug auf die vollständigen Werte der Variable und der ganzen Datenbank berechnet, wie in Abschnitt 1.3.4 erläutert. Somit existiert jeder ursprüngliche Fall in jeder imputierten vollständigen Datenbanken einmal und falls der Fall fehlende Daten aufwies, unterscheidet sich der Fall in den imputierten vollständigen Datenbanken nun jeweils in diesen Werten.

MRCBR - Online

Im Online-Teil wird nach Eingabe des Zielfalls auf jeder dieser imputierten vollständigen Datenbanken für alle Fälle die Retrieve-Phase durchgeführt. Das bedeutet, die lokalen und globalen Ähnlichkeiten zum Zielfall werden für alle Fall in jeder imputierten vollständigen Datenbanken in der Multiple Retrieval-Phase berechnet und in entsprechenden Ähnlichkeitsdatenbanken gespeichert. Wie zuvor in den imputierten vollständigen Datenbanken unterscheiden sich die lokalen Ähnlichkeiten der entsprechenden imputierten Daten und daraus folgend die globalen Ähnlichkeiten der jeweiligen Fälle in allen Ähnlichkeitsdatenbanken.

Im Anschluss werden in der Retrieval-Pooling-Phase die globalen Ähnlichkeiten der entsprechenden selben Fälle in den Ähnlichkeitsdatenbanken gemittelt, so dass nur noch eine Datenbank mit den globalen Ähnlichkeiten jedes Falles zurückbleibt. Die Vereinigung der globalen Ähnlichkeiten gewährleistet, dass die volle Bandbreite der möglichen Ähnlichkeiten der fehlenden Daten in einem Ergebnis zusammenfließt und ihr individueller Einfluss auf die finale Rangfolge der Fälle bewahrt bleibt.

Die globalen Ähnlichkeiten werden - nun um eine Rangfolge zu bilden - absteigend sortiert und die weiteren Phasen des CBR verlaufen wie gewohnt.

Unvollständiger Zielfall

Der Zielfall selbst kann gewollt oder ungewollt auch unvollständig sein. In diesem Falle werden nur die vollständigen Werte des Falles verwendet und die Datenbank während dieses Durchlaufs nur auf die Variablen der vollständigen Werte des Zielfalls verkürzt. Es geschieht keine Imputation, um die Ergebnisse der Retrieve-Phase nicht zu verzerren, da man keine sichere Aussage über die fehlenden Werte mit einer singulären fixen Imputation des Zielfalles machen kann. Da der Zielfall noch nicht in der Revise- und Retain-Phase verifiziert und somit nicht in CBR Kreislauf aufgenommen wurde, hat er keinen Anteil an der Multiple Imputation des MRCBR.

Die vollständigen Variablenwerte des Zielfalls werden allein in der Retrieve-Phase genutzt, um in der Lösung nicht die Vielfältigkeit der ähnlichsten Fälle einzuschränken. Diese können daher nur in den unvollständigen Variablenwerten des Zielfalls variieren und ansonsten identisch sein. In der Retain-Phase kann der unvollständige Zielfall nach Verifizierung in die Datenbank aufgenommen werden. Damit wird er im nächsten Durchlauf Teil des MRCBR, so dass seine vollständigen Variablenwerte das Verfahren in der Multiple Imputation-Phase als auch im Multiple Retrieval-Phase verbessern.

MRCBR als universelles Verfahren

Das MRCBR wurde für das Entscheidungsunterstützungssystem für das Tumorboard (CBR-TDS) entwickelt, um eine Lösung für die fehlenden Daten in MOSAIQ zu bieten.

Doch das aufgeführte Verfahren ist kein statisches CBR System und kann für jede beliebige Anwendung mit verschiedenen Schwerpunkten angepasst werden, wenn die Fragestellung auftritt, wie man mit den fehlenden Daten umzugehen hat.

Außer den universellen CBR Methoden aus Abschnitt 1.4.1 ist es die einzige Methode, welche die eingebettete Ähnlichkeitsfunktion in der Retrieve-Phase bewahrt und somit kein Anwendungsfeld vorschreibt. Das MRCBR agiert als eine Blackbox und integriert die Erfordernisse jeder CBR Anwendung in seinen Kreislauf, ohne sie zu manipulieren, und ist ein universelles Tool für jede Art von CBR und Einsatzfeld. Die jeweiligen Ähnlichkeitsfunktionen werden ganz natürlich in das Multiple Retrieval integriert um das erwünschte Ziel der Anwendung zu erfüllen. Die weiteren Phasen Reuse, Revise und Retain agieren unabhängig von der Retrieve-Phase, auch wenn ihre Lösungen direkt von dieser abhängen, und können je nach Anforderung beliebig erweitert oder verändert werden.

Das Verfahren ist nicht limitiert auf die reine Nutzung der Multiple Imputation und dem darauffolgenden Multiple Retrieval. Auf Wunsch kann es jede Methode für den Umgang mit fehlenden Daten aus Abschnitt 1.3 und die universellen Methoden aus Abschnitt 1.4.1 verwenden. Des Weiteren kann es unvollständige Datenbanken als auch vollständige Datenbanken verarbeiten. In diesen Fällen werden die Teile der Multiple Imputation, des Multiple Retrieval und Multiple Pooling für die entsprechenden Anforderungen automatisch angepasst oder übersprungen. Allerdings wird in den Ergebnissen von Kapitel 3 gezeigt und in der Diskussion in Kapitel 4 erörtert werden, dass dies nur in Ausnahmefällen in Betracht gezogen werden sollte. Es bleibt damit dem Nutzer überlassen, welche Schwerpunkte er setzen möchte.

2.1.3 Ablauf und Algorithmus

Der schematische Ablauf des Algorithmus des Multiple Retrieval Case-based Reasoning (MRCBR) ist in Abbildung 2.2 dargestellt und richtet sich nach den Schritten und Phasen des in diesem Abschnitt erläuterten Verfahren.

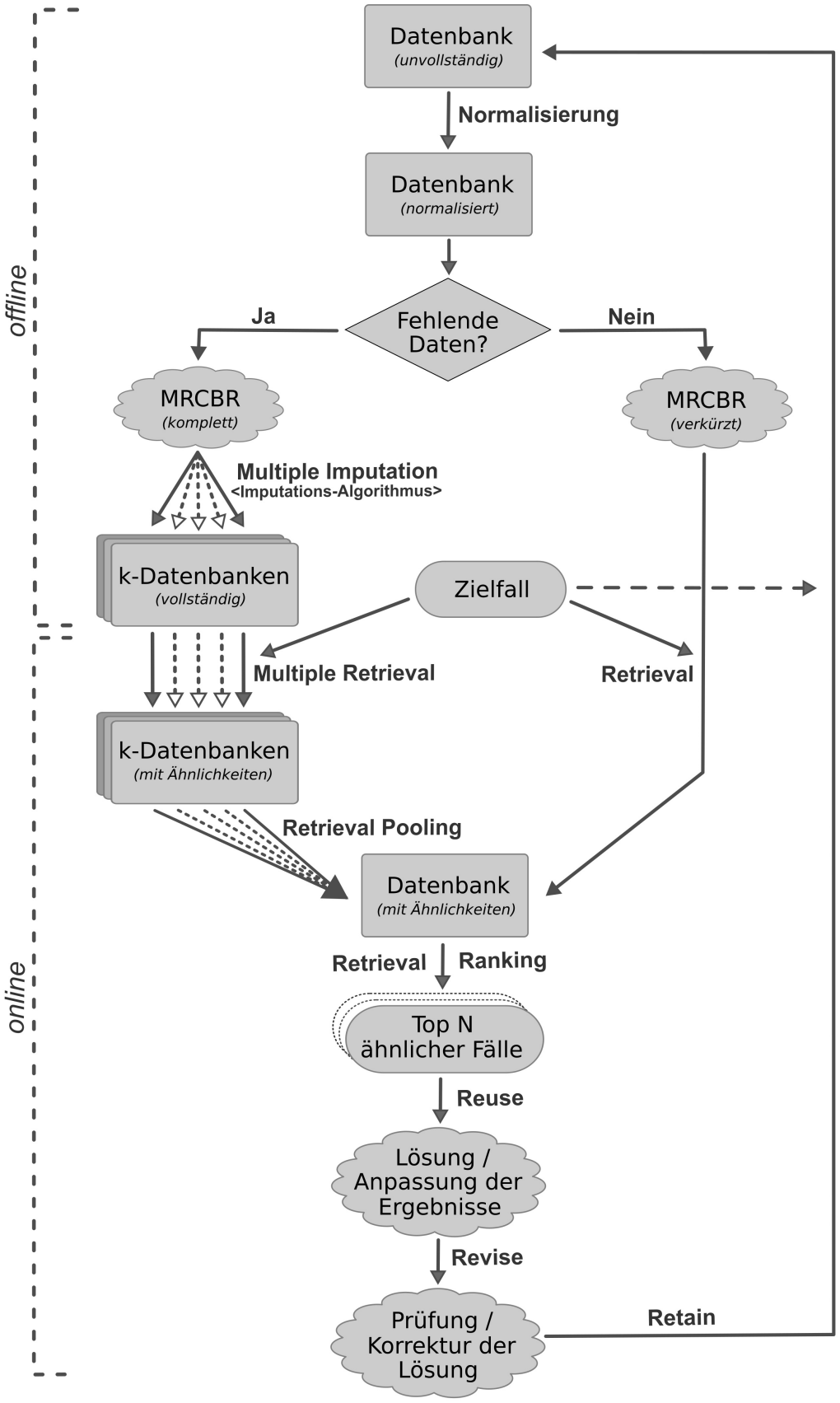


 Abbildung 2.2: Algorithmus des Multiple Retrieval Case-based Reasoning.

Definitionen und Erläuterungen

Die folgenden Begriffe und Definitionen werden für die theoretischen Ausführungen des MRCBR-Algorithmus verwendet. D bezeichnet die ursprüngliche (unvollständige) Datenbank. D enthält n Variable V_i $(1 \leq i \leq n)$ und x_{ij} $(1 \leq j \leq m)$ sind die Elemente der Variable V_i. Ein Fall C_j der Datenbank D wird repräsentiert durch $C_j = (x_{1j}, x_{2j}, ..., x_{ij}, ..., x_{nj})$. $mean$ steht für den Mittelwert und std für die Standardabweichung [56].

Die Teile der Multiple Imputation folgen dem in Abschnitt 1.3.4 erläuterten Verfahren [115]. Das Multiple Retrieval beinhaltet bis auf kleine Änderungen in der Normierung die in Abschnitt 1.2.1 beschriebene Hybrid-Ähnlichkeitsfunktion [37]. Diese Hybrid-Ähnlichkeitsfunktion ist für die Erfordernisse des CBR-TDS gewählt worden. Andere Ähnlichkeitsfunktionen sind ohne Einschränkungen auf das Verfahren und das Verhalten des MRCBR möglich. Die Reuse-, Revise- und Retain-Phase sind allgemein gehalten und an die Erfordernisse des CBR-TDS angelehnt.

Das MRCBR teilt sich in einen Offline-Teil, der die Analyse und Verarbeitung der fehlenden Daten durchführt, und einen Online-Teil, der die vier Phasen des Case-based Reasoning mit dem erweiterten Multiple Retrieval und Retrieval Pooling beinhaltet.

MRCBR - Offline (Fehlende Daten und Imputation)

1. **Normalisierung**: Jede numerische Variable V_i aus D wird mit einer z-Transformation $(x_{ij} - mean(V_i))/std(V_i))$ normalisiert, so dass $mean(V_i) = 0$ und $std(V_i) = 1$ gilt.

2. **Fehlende Daten**: Es wird geprüft, welche Verteilung die fehlenden Daten in D aufweisen und welche Variablen V_i betroffen sind. Wenn keine fehlenden Daten vorhanden sind, wird ein gewöhnliches Retrieval ohne die Schritte Multiple Imputation, Multiple Retrieval und Retrieval Pooling durchgeführt.

3. **MCAR Test**: Little's Test wird auf der Datenbank D ausgeführt um zu prüfen, ob die MCAR Hypothese beibehalten wird. Bei Ablehnung wird darauf hingewiesen

zusätzliche Informationen über die Gründe der fehlenden Daten in Erfahrung zu bringen, da es sich um kein zufälliges Fehlen handelt.

4. **Imputations-Algorithmus**: Wahl eines passenden Klassifikations-Algorithmus für die Imputation der fehlenden Daten. Verschiedene Algorithmen für verschiedene Variablen V_i sind möglich.

5. **Multiple Imputation**: Erzeugung von k Kopien der ursprünglichen Datenbank D. Imputation der k unvollständigen Datenbanken ID^l $(1 \leq l \leq k)$ mit Hilfe des entsprechenden Imputations-Algorithmus. Jede Datenbank ID^l mit Fällen C_j^l ist nun vollständig und hat leicht unterschiedliche imputierte Werte, um die Unsicherheit betreffs der Imputation widerzuspiegeln.

MRCBR - Online (Case-based Reasoning)

1. **Zielfall**: Wahl eines Zielfalles $T = (t_1, t_2, ..., t_i, ..., t_n)$. Normierung aller numerischen Werte t_i von T mit den entsprechenden Werten $mean(V_i)$ und $std(V_i)$ aus dem offline Schritt "1. Normalisierung". Bei fehlenden Daten innerhalb des Zielfalles T werden für die weiteren Berechnungen für alle Fälle C_j^l nur die Variable V_i der entsprechenden vollständigen Variablenwerte von T betrachtet, wie in Abschnitt 2.1.2 erläutert wurde.

2. **Multiple Retrieval**: Die globale Ähnlichkeit des Zielfall T mit allen Fällen C_j^l in jeder Datenbank ID^l besteht aus einer Hybrid-Ähnlichkeitsfunktion, welche sich aus den lokalen Ähnlichkeiten der numerischen und kategorialen Variablen V_i zusammensetzt. Die lokale Ähnlichkeit zwischen dem Element x_{ij}^l des Falles C_j^l und dem Element t_i des Zielfalles T wird mit $sim_{ij}^l(T, C_j^l)$ bezeichnet.

   Kategoriale lokale Ähnlichkeit: Für kategoriale Variablen kann nur eine *wahr* oder *falsch* Aussage bezüglich der Ähnlichkeit zweier Elemente x_{ij}^l und t_i getroffen werden. Die kategoriale lokale Ähnlichkeit wird definiert mit $sim_{ij}^l = 1$, wenn zwei

Elemente x_{ij}^l und t_i identisch sind, und mit $sim_{ij}^l = 0$, wenn sie sich unterscheiden:

$$sim_{ij}^l(T, C_j^l) = \begin{cases} 1, & \text{if } x_{ij}^l = t_i \\ 0, & \text{if } x_{ij}^l \neq t_i \end{cases}$$

Numerische lokale Ähnlichkeit: Für numerische Variablen wird die lokale Ähnlichkeit mit einem festgelegten Abstandsmaß (hier Manhattan-Distanz) definiert. Je näher die Elemente x_{ij}^l und t_i auf Basis dieses Abstandsmaßes sind, desto ähnlicher sind sie sich. Die Manhattan-Distanz $\delta_{ij}^l = |x_{ij}^l - t_i|$ wird zwischen allen C_j^l und T berechnet. Dann wird δ_{ij}^l einer Min/Max Normalisierung unterzogen zu $\Delta_{ij}^l = \frac{\delta_{ij}^l - min(\delta_i^l)}{max(\delta_i^l) - min(\delta_i^l)}$ mit $\delta_i^l = (\delta_{i1}^l, \delta_{i2}^l, ..., \delta_{ij}^l, ..., \delta_{im}^l)$. Das normierte Abstandsmaß Δ_{ij}^l wird für eine bessere intuitive (natürliche) Darstellung mit der inversen Exponentialfunktion transformiert und erstreckt sich auf das Intervall $[0, 1]$. Die numerische lokale Ähnlichkeit $sim_{ij}^l(T, C_j^l)$ ist als Transformation von Δ_{ij}^l mit der inversen Exponentialfunktion definiert:

$$sim_{ij}^l(T, C_j^l) = exp^{-\Delta_{ij}^l}$$

Globale Ähnlichkeit: Die Summe aller lokalen Ähnlichkeiten $sim_{ij}^l(T, C_j^l)$ multipliziert mit ihren entsprechenden Gewichten w_i definiert die globale Ähnlichkeit $SIM_j^l(T, C_j^l)$ zwischen dem Fall C_j^l und dem Zielfall T:

$$SIM_j^l(T, C_j^l) = \sum_{i=1}^{n} w_i \cdot sim_{ij}^l(T, C_j^l),$$

mit $\sum_{i=1}^{n} w_i = 1$ und $0 \leq w_i \leq 1$.

Jede Datenbank SIM^l enthält die lokalen und globalen Ähnlichkeiten aller Fälle C_j^l. Alle SIM^l unterscheiden sich ein bisschen, da die unterschiedlichen imputierten Werte der verschiedenen ID^l die Ergebnisse der lokalen und globalen Ähnlichkeiten in jedem SIM^l beeinflussen.

3. **Retrieval Pooling**: Erhalt der alleinigen globalen Ähnlichkeit SIM_j^* zwischen jedem Fall C_j und dem Zielfall T durch die Berechnung des Durchschnitts der entsprechenden globalen Ähnlichkeiten in allen SIM^l:

$$SIM_j^*(T, C_j) = \sum_{l=1}^{k} SIM_j^l(T, C_j^l) \Big/ k$$

4. **Retrieval Ranking**: Speicherung der globalen Ähnlichkeit zu jedem Fall bis zum Ende des Durchlaufs. Absteigende Sortierung der Fälle C_j gemäß ihren globalen Ähnlichkeiten in Bezug auf den Zielfall T. Die sortierten Fälle werden auf die geforderte Anzahl N der ähnlichsten Fälle reduziert, genannt *Top N* Fälle.

5. **Reuse**: Erhalt der Informationen und Lösungen der *Top N* Fälle. Bei Bedarf mögliche Anpassung der Lösungen an die Erfordernisse der Anwendung. Im Falle von CBR-TDS wird die individuelle Information jedes *Top N* Falles für die weitere Analyse und Verarbeitung angezeigt.

6. **Revise**: Prüfung der Lösungen. Falls erforderlich Korrektur der Lösungen. Im Falle von CBR-TDS benennt der Nutzer die für seine Fragestellung passendsten Fälle aus den *Top N* Fällen und wendet ihre Lösung auf den Zielfall T an.

7. **Retain**: Der Zielfall T und optional die angepasste Lösung werden zur ursprünglichen Datenbank D hinzugefügt. Jeder neue Zielfall T sorgt mit seiner Information für eine Verbesserung des nächsten Kreislaufes, sowohl für die Multiple Imputation als auch für das Multiple Retrieval.

2.2 Aufbau und Grundlagen der Evaluation

Die Evaluation des Multiple Retrieval Case-based Reasoning (MRCBR) wurde für verschiedene Szenarien durchgeführt um möglichst unterschiedliche Situationen und Bedingungen zu untersuchen, die in realen Anwendungsfeldern und ihren Datenbanken zum Tragen kommen. Dies betrifft verschiedene Aspekte.

Zum einen die Anzahl der fehlenden Daten innerhalb der Datenbank, die in unterschiedlichen Anteilen in den Variablen vorkommen können. Zum anderen den Typ der fehlenden Daten, die alle auch gemischt in einer unvollständigen Datenbank vorkommen können. Des Weiteren die Auswahl der betroffenen Variablen, welche die häufigsten Arten von Variablen in Datenbanken abdecken soll. Auch der Einfluss der Größe der Datenbank auf die Ergebnisse muss in Betracht gezogen werden. Und zuletzt die Selektion der Zielfälle für das Case-based Reasoning, um auch seltene Fälle nicht zu vernachlässigen und eine ausgewogene Mischung zu garantieren. Für den Nachweis der Stärken und Schwächen wurde MRCBR mit Methoden verglichen, die dem Stand der Technik entsprechen, um CBR auf Grundlage von fehlenden Daten verwenden zu können.

In vielfältigen Experimenten wurden die Szenarien, welche soeben umrissen wurden, erfüllt und geprüft. Der Kern der Experimente beruht darauf, die Genauigkeit der Verfahren in Bezug auf ihre korrekte Rangfolge nach dem Retrieval zu prüfen. Dafür werden die Ergebnisse des Retrievals auf der wahren vollständigen Datenbank mit den Ergebnissen des MRCBR und der konkurrierenden Methoden auf derselben Datenbank, welche mit künstlich erzeugten unvollständigen Daten belegt wurde, mit Hilfe von Fehlermaßen verglichen.

In den folgenden Abschnitten werden die Voraussetzungen für die Experimente in den unterschiedlichen Bereichen detailliert beschrieben. Die Ergebnisse der Experimente werden in Kapitel 3 vorgestellt.

2.2.1 Datenbank und Variablen

Für die Evaluation bedarf es einer vollständigen Datenbank, die keine Ausreißer oder Fehleinträge aufweist. Wie in der Einführung bereits erwähnt, ist MOSAIQ selbst in Weiten teilen unvollständig und nicht immer gut dokumentiert, so dass es sich aufgrund dessen nicht für eine gesicherte Evaluation eignet. Deshalb wurde die Evaluation auf einer frei verfügbaren vollständigen realen medizinischen Datenbank durchgeführt.

Herkunft und Aufbau der Datenbank

Die verwendete Datenbank gründet auf einer klinischen Studie für Östrogen Behandlung von Prostatakarzinomen aus dem Jahr 1980 [26] und wurde von der Vanderbilt University School of Medicine in Nashville zur Verfügung gestellt.

Die Datenbank enthält 17 Variablen, die sich in drei Arten aufteilen. Neun Variablen enthalten Integer, abgekürzt *Int*, zwei Variablen bestehen aus Gleitkommazahlen, welche von nun an *Float* genannt werden, und sechs Variablen sind kategorial, genannt *Kat*. Aus den ursprünglich 502 Fällen wurden 28 unvollständige und fehlerhafte Fälle entfernt, so dass 474 vollständige Fälle in der neu gewonnen Datenbank D_t erhalten blieben.

Unterschiedliche Größen der Datenbank

Um den Einfluss der Größe einer Datenbank auf die verschiedenen Verfahren zu prüfen, wurde nicht nur die Datenbank D_t mit voller Größe untersucht, sondern sie wurde durch zufällige Löschung zum einen auf 50% ihrer Größe mit 237 Fällen und zum anderen auf 25% ihrer Größe mit 118 Fällen verkleinert. Der für die Löschung nach dem Zufallsprinzip benutzte Seed unterscheidet sich von dem Verwendeten für das MCAR Szenario, welches im übernächsten Abschnitt erläutert wird.

Selektion der Variablen

Es wurden sieben Variablen aus der Datenbank D_t für die Versuche ausgewählt, welche einzeln als auch in Kombinationen in den verschiedenen Szenarien für fehlende Daten verwendet wurden. Die Variablen spiegeln die Struktur der Datenbank wieder und weisen unterschiedliche Verteilungen auf.

In den folgenden Tabellen und Abbildungen werden für eine schnellere Übersicht die medizinischen Bezeichnungen der selektierten Variablen wie folgt abgekürzt:

- *Weight Index* zu *W.I.*
- *Systolic Blood Pressure* zu *S.B.P.*

- *Dystolic Blood Pressure* zu *D.B.P.*
- *Hemoglobin* zu *Hg.*
- *Prostatic Acid Phosphatase* zu *P.A.P.*
- *Status* zu *St.*
- *Cardiovascular Disease* zu *Cv.D.*

Eine Auflistung der ausgewählten Variablen und ihrer statistischen Parametern findet sich in Tabelle 2.1. Die Spalte Missing wird im späteren Verlauf noch erläutert und gibt den Typ der fehlenden Daten an, welche die betreffende Variable in einem künstlich wahren Fall Szenario einnimmt. Eine grafische Darstellung der Histogramme der Variablen wird in Abbildung 2.3 dargestellt.

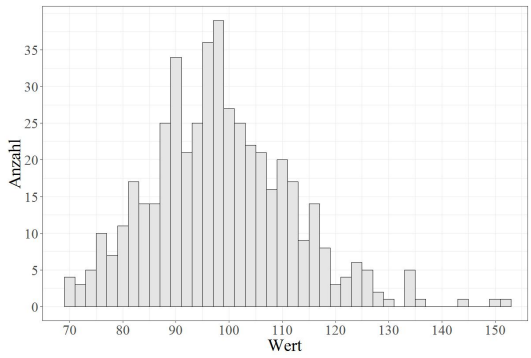
Tabelle 2.1: Ausgewählte Variablen mit ihren statistischen Parametern und dem zugehörigen Typ der fehlenden Daten im künstlich wahren Fall Szenario.

Variable	Typ	Median	Mean	Std	Min	Max	Level	Missing
W.I.	Int	98.0	99.0	13.4	69.0	152.0	–	MCAR
S.B.P.	Int	14.0	14.4	2.4	8.0	30.0	–	MAR
D.B.P.	Int	8.0	8.2	1.5	4.0	18.0	–	MNAR
Hg.	Float	13.7	13.4	1.9	5.9	18.2	–	MNAR
P.A.P.	Float	0.7	10.5	44.9	0.1	596.0	–	MCAR
St.	Kat	–	–	–	–	–	10	MNAR
Cv.D.	Kat	–	–	–	–	–	2	MAR

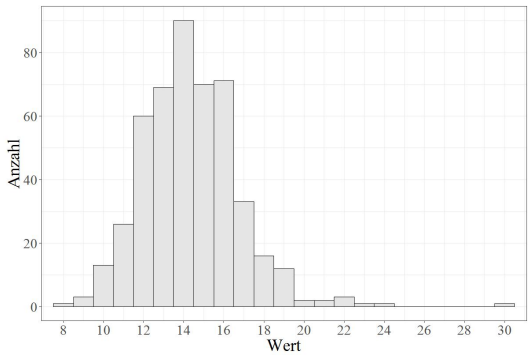
Verfahren der Selektion

Die kategorialen Variablen decken mit *Status*, welches 10 Level hat, und mit *Cardiovascular Disease*, welches nur 2 Level hat, die beiden Enden der möglichen Erscheinungsformen der Kategorialen der Datenbank D_t ab.

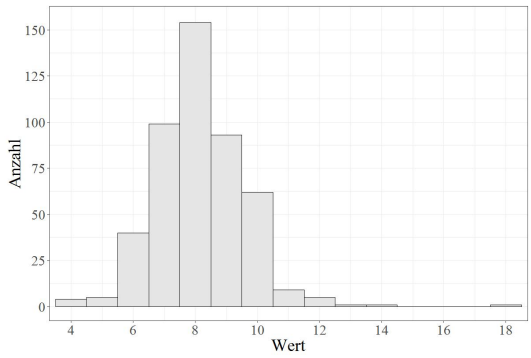
Die numerischen Variablen wurden mit Hilfe des Korrelationskoeffizienten nach Pearson ausgewählt, so dass die Variablen unterschiedliche Abhängigkeiten von einander aufweisen. Für zwei Variablen X und Y wird der Korrelationskoeffizienten nach Pearson mit $\frac{cov(X,Y)}{std(X)*std(X)}$ berechnet.[104]. Die Korrelationskoeffizienten nehmen das Intervall $[-1, 1]$ ein, dabei bedeutet 0 keine Korrelation, -1 eine negative Korrelation (entgegengesetztes Verhalten der Steigung der Werte) und 1 eine positive Korrelation.



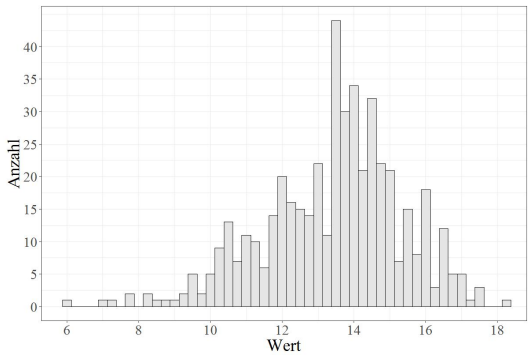
(a) W.I.



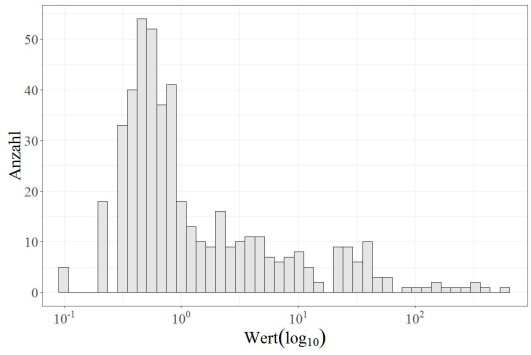
(b) S.B.P.



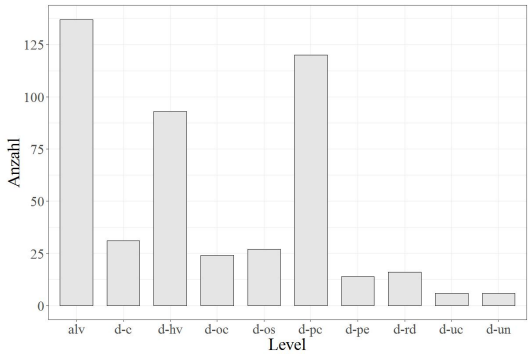
(c) D.B.P.



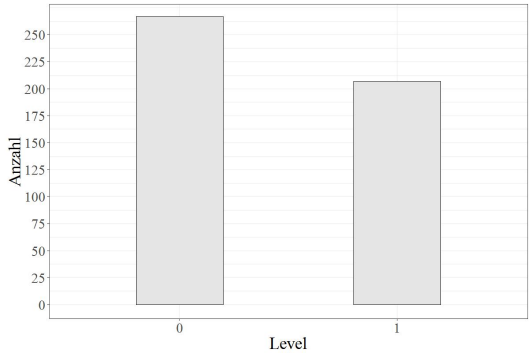
(d) Hg.



(e) P.A.P.



(f) St.



(g) Cv.D.

Abbildung 2.3: Histogramme der ausgewählte Variablen geordnet nach ihrer Art. Integer in 2.3a, 2.3b und 2.3c. Float in 2.3d und 2.3e. Kategoriale in 2.3f und 2.3g.

Tabelle 2.2 zeigt die verschiedenen Korrelationen zwischen den ausgewählten numerischen Variablen.

Tabelle 2.2: Korrelation der ausgewählten numerischen Variablen.

Variable	W.I.	S.B.P.	D.B.P.	Hg.	P.A.P.
W.I.	–	0.19	0.23	0.26	-0.09
S.B.P.	0.19	–	0.63	0.06	-0.03
D.B.P.	0.23	0.63	–	0.14	-0.07
Hg.	0.26	0.06	0.14	–	-0.14
P.A.P.	-0.09	-0.03	-0.07	-0.14	–

Verwendung der Variablen in den Experimenten

In den Szenarios von Kapitel 3 fungieren drei der ausgewählten Variablen repräsentativ für ihre Art der Variable und werden in den Ergebnissen zum leichteren Verständnis nur mit der betreffenden Abkürzung benannt. *Weight Index* repräsentiert die Integer und wird in den Versuchen nur mit *Int* bezeichnet. *Hemoglobin* steht für die Float und lautet dort *Float*. *Status* vertritt die kategoriale Variablen, abgekürzt mit *Kat*.

Des Weiteren wurden alle Kombinationen aus diesen verwendet und mit den Anfangsbuchstaben ihrer Art bezeichnet: Integer und Float mit *I.F.*, Integer und Kategoriale mit *I.K.*, Float und Kategoriale mit *F.K.* und aller drei zusammen mit *I.F.K.*. All diese Variablen und ihre Kombinationen stehen für einen der sieben Versuche jedes Experiments.

Das künstlich wahre Fall Szenario, welches im übernächsten Abschnitt eingeführt wird, und ein eigenes Experiment darstellt, verwendet alle sieben selektierten Variablen für die Erzeugung von fehlenden Daten mit unterschiedlichen Typen.

2.2.2 Zielfälle des CBR

Die Ergebnisse der Verfahren werden von der Wahl des Zielfalles beeinflusst, auf welchen sich die Berechnungen des CBR beziehen, das auf der von den Verfahren aufbereiteten Datenbank ausgeführt wird. Ein einzelner Zielfall, der häufig vorkommende

Variablenwerte aufweist, würde bei den meisten verglichenen Verfahren gute Ergebnisse erzielen, da seine Werte auch im Rest der Datenbank häufig vorkommen und falls sie fehlen einfach zu schätzen sind. Daher wurden für die Experimente 20 unterschiedliche Zielfälle verwendet und die Ergebnisse über diese Zielfälle gemittelt.

Selektion der Zielfälle

Damit die 20 Zielfälle möglichst verschiedene Werte in den ausgewählten Variablen aufweisen, wurden sie durch eine Clusteranalyse als Repräsentanten der Datenbank D_t erzeugt [11]. Die Gruppe der erhaltenen Zielfälle deckt mit nur einer kleinen Abweichung die statistischen Parameter der Datenbank D_t ab.

Die Zielfälle sind in allen Versuchen ohne Einschränkung der Allgemeinheit vollständig. Wie in Abschnitt 2.1.2 erläutert wurde, schrumpft die Datenbank bei Eingabe eines unvollständigen Zielfalles während des Durchlaufs auf die Variablen der vollständigen Werte des Zielfalles zusammen. Alle angewendeten Verfahren haben dadurch wieder die dieselben Grundvoraussetzungen, so dass die Unvollständigkeit des Zielfalls keinen Einfluss auf die Evaluation dieser Arbeit hat.

Tabelle 2.3 zeigt die selektierten Zielfälle mit den dazugehörigen Werten in den ausgewählten Variablen aus Abschnitt 2.2.1.

Verfahren der Selektion

Da in der Datenbank numerische und kategoriale Variablen vorkommen, ist ein simpler k-Means-Algorithmus zur Clusteranalyse nicht ausreichend, da kein Mittelwert bestimmt werden kann. Der k-Medoids-Algorithmus verwendet die Aufteilung durch Medoide (PAM) [90], welche immer selbst ein Teil der Daten sind. Dadurch entfällt die Notwendigkeit einer Mittelung. Er verhält sich stabil gegenüber Rauschen und Ausreißern.

Als Metrik innerhalb des k-Medoids-Algorithmus wird die Gower Distanz genutzt. Die Distanz für numerische Variablen wird hierbei mit der Manhattan-Metrik berech-

Tabelle 2.3: Ausgewählte Zielfälle mit den Werten der selektierten Variablen.

Patient	W.I.	S.B.P.	D.B.P.	Hg.	P.A.C.	St.	Cv.D.
465	114	30	18	11.6	0.5	alv	1
216	83	12	6	11.4	596.0	alv	0
149	96	14	10	10.5	29.9	d-c	0
94	102	16	9	18.2	0.6	d-c	1
418	85	15	7	5.9	0.3	d-hv	0
219	106	16	9	14.7	36.9	d-hv	1
322	79	13	9	10.9	0.4	d-oc	1
28	126	17	10	14.0	24.2	d-oc	0
293	78	14	7	14.9	0.2	d-os	0
143	98	15	8	10.2	127.8	d-os	1
442	134	16	8	14.1	1.2	d-pc	1
64	71	8	4	8.7	175.1	d-pc	0
360	89	18	8	14.3	25.3	d-pe	1
256	82	13	7	9.4	0.6	d-pe	0
305	92	13	6	13.6	0.7	d-rd	1
138	124	18	11	15.6	7.0	d-rd	0
443	107	14	9	14.1	0.2	d-uc	0
152	118	18	8	14.5	0.7	d-uc	1
472	127	16	10	15.8	0.5	d-un	1
245	86	14	8	10.2	0.40	d-un	0

net. Die kategorialen werden in mehrere binäre Variablen aufgeteilt und der Dice-Koeffizient verwendet [114].

Es wurden 10 Cluster erzeugt, in denen jeweils die zwei weitest entfernten Fälle ausgewählt wurden. Dadurch ist gewährleistet, dass die Zielfälle einerseits als Basis für die Datenbank dienen können, da sie das ganze Spektrum der Werte abbilden, andererseits auch Werte innehaben, die selten vorkommen, was besonders für manche Level der kategorialen Variable von Wichtigkeit ist.

2.2.3 Erzeugung der fehlenden Daten und Szenarien

Die künstliche Erzeugung der fehlenden Daten berücksichtigt zum einen den Anteil fehlender Daten und zum anderen den Typ der fehlenden Daten. Beides wird auf einer oder mehreren Variablen bewerkstelligt. Diese beiden Parameter haben einen signifikanten Einfluss auf das Verhalten der Methoden.

Anteil fehlender Daten

Jeder einzelne Versuch in den Abschnitten von Kapitel 3 wurde mit einer ansteigenden Rate fehlender Daten durchgeführt. Die fehlenden Daten innerhalb einer oder mehrerer Variablen reichen von 10% bis 90% mit einer Schrittlänge von 10%. Gemischte Prozentsätze innerhalb eines Szenario wurden aufgrund ihrer Vielzahl von Möglichkeiten nicht verfolgt.

Typen von fehlenden Daten

Die Implementierung der verschiedenen Typen von fehlenden Daten MCAR, MAR und MNAR folgt der Beschreibung in Abschnitt 1.3.1. Für den Typ MCAR wurden Werte der betreffenden Variable nach dem Zufallsprinzip aufgrund eines festgelegten Seeds und Prozentwertes der fehlenden Daten gelöscht. Für den Typ MNAR wurden die Werte der Variable der Größe nach geordnet und ein Intervall innerhalb dieser geordneten Werte gelöscht. Bei kategorialen Variablen wurden die Level alphabetisch geordnet. Die Länge des Intervalls beruht auf dem festgelegten Prozentwert der fehlenden Daten. Für den Typ MAR wird ähnlich verfahren wie für MNAR, nur, dass die Werte der verbundenen Variable geordnet werden, das Intervall aber in der abhängigen Variable selbst gelöscht wird.

Das künstlich wahre Fall Szenario

Die Szenarien wurden jeweils auf einzelnen Variablen als auch auf Kombinationen von unterschiedlichen Arten von Variablen durchgeführt, um die Auswirkungen auf das Ergebnis zu ergründen. In echten Datenbanken allerdings sind die fehlenden Daten als Folge einer Vermischung der drei Typen von fehlenden Daten in unterschiedlichen Variablen der Datenbank entstanden. Um dieser Anforderung gerecht zu werden, wurde ein künstlich wahres Fall Szenario (kwFS) erschaffen, dass alle sieben ausgewählten Variablen aus Abschnitt 2.2.1 mit verschiedenen Typen von fehlenden Daten belegt.

Die Zuordnung der Variablen mit welchem Typ von fehlenden Daten sie gelöscht wurden, ist in Tabelle 2.1 unter der Spalte Missing vermerkt. Alle Variablen mit MAR hängen von der selben Variable *D.B.P.* ab und die zugehörigen Fälle haben somit mindestens drei fehlende Werte. Es besteht die Möglichkeit, dass alle sieben Variablenwerte gelöscht wurden. Das kwFS dient als eine Art Worstcase-Szenario, um auch extreme aber realistische Bedingungen abzubilden.

Aufbau der Versuche und Szenarien

Die Szenarien in Kapitel 3 spiegeln die verschiedenen Möglichkeiten der fehlenden Daten wider. Daher wurden nicht nur die verschiedenen Typen von fehlenden Daten einzeln für die selektierten Variablen und Kombination daraus getestet, sondern auch das Verhalten der Methoden unter dem kwFS. Außerdem wurden die Auswirkungen der Größe der Datenbank und die verwendete Anzahl der ähnlichsten Fälle nach dem Retrieval in einem eigenen Szenario erforscht.

Die jeweiligen sieben Versuche der verschiedenen Szenarien, bestehend aus den drei selektierten Variablen und den entsprechenden Kombinationen, wurden 10-mal auf verschiedenen unvollständigen Datenbanken wiederholt, die jeweils durch eine immer neue Zufälligkeit erzeugt wurden. Im Falle von MCAR wurden verschiedene Seeds benutzt um die fehlenden Daten in der betreffenden Variable zu löschen. Damit weist selbst bei einer 10% Rate fehlender Daten so gut wie jeder Fall einmal einen fehlenden Wert in dieser Variable auf. Bei MNAR und MAR wanderten die Intervalle vom Minimum zum Maximum der geordneten Variablenwerte, so dass der ganze Wertebereich der Variable abgedeckt wurde. So wurde sichergestellt, dass jeder Wert einmal innerhalb der betroffenen Variable gelöscht wurde. Im Falle, dass mehrere Variablen mit fehlenden Daten des Typs MCAR erzeugt wurden, wurde für jede dieser Variable ein anderer Seed verwendet. Die Intervalle bei MNAR und MAR sind aufgrund der eigenen Statistik jeder Variable automatisch unterschiedlich. Aufgrund dieses Aufbaus konnten Fälle mehrere fehlende Daten aufweisen, wenn mehrere Variablen von fehlenden Daten betroffen waren. Vor allem beim künstlich wahren Fall Szenario war dies der Fall.

Das Ergebnis jedes Versuchs in einem Szenario ist der Durchschnitt aus 200 einzelnen kleineren Versuchen. Die Anzahl dieser kleineren Versuche ergibt sich daraus, dass die verglichenen Verfahren für alle 20 selektierten Zielfälle auf den 10 unvollständigen Datenbanken durchgeführt wurden. Die Zielfälle weisen für die Variablen nahezu die statistischen Eigenschaften der Datenbank auf, so dass im Hinblick auf numerische Variablen das ganze mögliche Werteintervall abgedeckt wurde und bei Kategorialen jedes Level. Da in den 10 generierten unvollständigen Datenbanken die Werte der betroffenen Variablen in jedem Fall mindestens einmal gelöscht wurden, fand auch immer eine Anfrage eines der Zielfälle nach diesem Wert statt. So ist gewährleistet, dass alle Werte gleichermaßen und gleichberechtigt gelöscht und angefragt wurden, damit jede Methode die Chance hat diese korrekt wiederherzustellen. Die 200 einzelnen kleineren Versuchen stehen damit repräsentativ für alle Möglichkeiten, die sich bei Anfragen des Zielfalls und fehlenden Daten ergeben könnten. Im zusammenfassenden Vergleich der Szenarien entsteht das Ergebnis als Durchschnitt aus den sieben jeweiligen Versuchen und besteht aufgrund dessen aus 1400 einzelnen Resultaten.

2.2.4 Methoden und Implementierung

Für die Evaluation von MRCBR wurden die vorgestellten Methoden aus Abschnitt 1.3 und Abschnitt 1.4, welche CBR die Fähigkeit verleihen unter der Voraussetzung von fehlenden Daten arbeiten zu können, zum Vergleich herangezogen und implementiert.

Selektion der Methoden

Die ausgewählten Methoden stehen für die verschiedenen Klassen von Methoden von simpler Löschung der Fälle über einfache Imputation von festen Werten bis zu modernen Klassifikations-Algorithmen. Um die am Anfang des Kapitels erläuterten Bedingungen einer umfassenden Evaluation zu erfüllen, wurden nur die Methoden aufgenommen, die in der Lage sind sowohl numerische als auch kategoriale Variablen zu verarbeiten und auch sonst keine Einschränkungen fordern.

Von den CBR Methoden aus Abschnitt 1.4 erfüllen diese Bedingung nur die universellen Methoden, welche einen festgelegten Wert, 0 oder 0.5, für die lokale Ähnlichkeit des fehlenden Wertes einsetzen und von nun an *N/A Sim 0* und *N/A Sim 0.5* genannt werden. Diese beiden sind außerdem neben MRCBR die einzigen CBR Methoden, welche die Ähnlichkeitsfunktion im Retrieval unverändert lassen. Die anderen CBR Methoden für fehlende Daten wurden aus unterschiedlichen Gründen für die Evaluation verworfen, die von der Limitierung der Art der Variable [4, 55] über Anforderungen an die Struktur der Datenbank [73, 119, 103, 122] bis hin zu schlechter Dokumentation des Verfahrens [124] reichten.

Implementierung der Methoden

Die Programmiersprache R [89] dient als Grundlage für alle Implementierungen. Variablenlöschung, Listenweiser und paarweiser Fallausschluss (1.3.2), Mittelwert beziehungsweise MODE Substitution (1.3.3), *N/A Sim 0* und *N/A Sim 0.5* (1.4.1) wurden nach den Beschreibungen der Verfahren in den betreffenden Referenzen eigenhändig implementiert.

Für die Implementierung des Entscheidungsbaumes CART (1.3.3) wurde das Paket *rpart* [113] mit den Parametern minsplit = 5 und minbucket = 40 verwendet. Der Random Forest (1.3.3) wurde auf Basis des Paketes *randomForest* [67] implementiert. Die Anzahl der Bäume wurde auf 100 und die Anzahl der in den Bäumen genutzten Variablen auf 4 gesetzt.

Die Multiple Imputation (1.3.4) gründet sich auf dem Paket *MICE* [117] mit CART und Random Forest als Imputations-Algorithmen. Um einen fairen Vergleich zu gewährleisten wurden dieselben Implementierungen und Parameter für CART und Random Forest in der singulären Imputation als auch Multiple Imputation verwendet. Die theoretischen Referenzen zu den Toolboxen finden sich in den entsprechenden referenzierten Abschnitten.

Die Implementierung des MRCBR folgt der Anleitung aus Abschnitt 2.1.3. Alle verglichenen Methoden verwenden dasselbe dort erläuterte CBR Retrieval mit seinen

Ähnlichkeitsfunktionen für numerische und kategoriale Variablen. Die Reduzierung des MRCBR auf die verschiedenen genannten Methoden wurde in Abschnitt 2.1.2 explizit erläutert. Für die Löschungs- und Imputations-Methoden wird das CBR Retrieval auf den reduzierten oder imputierten Datenbanken ausgeführt, für die universellen CBR Methoden wird nur die Ähnlichkeit des fehlenden Wertes in der Ähnlichkeitsdatenbank mit dem entsprechenden Wert substituiert. Die n Gewichte w für die Berechnung der globalen Ähnlichkeit aus den lokalen Ähnlichkeiten wurden ohne Einschränkung der Allgemeinheit gleichverteilt definiert mit $\sum_{i=1}^{n} w_i = 1$ und $0 \leq w_i \leq 1$.

Das Anliegen dieser Arbeit ist es, die Korrektheit der aufgeführten Methoden in Bezug auf das Retrieval und die daraus folgende Rangfolge zu untersuchen. Die Reuse-, Revise- und Retain-Phase haben keinen Einfluss darauf, sind von Anwendung zu Anwendung verschieden und wurden nicht in die Evaluation aufgenommen. Selbstverständlich arbeitet das in Abschnitt 1.1 und Abschnitt 2.1.2 beschriebene CBR-TDS innerhalb des MRCBR als kompletter Kreislauf und kann auch so genutzt werden.

Abkürzungen der Methoden

In den nächsten Kapiteln werden die Methoden wie folgt abgekürzt:

- Variablenlöschung mit *Var Delete*
- Listenweiser Fallausschluss mit *List Delete*
- Paarweiser Fallausschluss mit *Pair Delete*
- Universelle CBR Methode mit Ähnlichkeit 0 mit *N/A Sim 0*
- Universelle CBR Methode mit Ähnlichkeit 0.5 mit *N/A Sim 0.5*
- Mittelwert und MODE Imputation mit *Mean*
- Entscheidungsbaum CART mit *CART*
- Random Forest mit *RF*
- MRCBR mit CART mit *MRCBR CART*
- MRCBR mit Random Forest mit *MRCBR RF*

2.2.5 Fehlerabschätzung

Zur Überprüfung der Korrektheit der Rangfolge nach dem Retrieval bedarf es eines Maßes der Genauigkeit um die unterschiedlichen Methoden zu vergleichen. Dies geschieht in dieser Arbeit mit Hilfe von verschiedenen Fehlern, welche die Qualität und Verlässlichkeit der Methoden bestimmen und die Möglichkeit bieten diese in Vergleich zueinander zu setzen. Außerdem wird auch ein Blick auf die Anzahl der zu verwendeten ähnlichsten Fälle geworfen, welche in verschiedenen CBR Systemen verwendet werden.

Beste Fälle *Top N*

Zur Berechnung der Fehler wurden die Fälle nach dem Retrieval des CBR auf der vollständige Datenbank D_t (2.2.1) der Größe nach betreffs ihrer globalen Ähnlichkeiten zum Zielfall sortiert und mit ihrem jeweiligen Platz in der Rangfolge gekennzeichnet. Der ähnlichste Fall zum Zielfall hat somit den Platz 1, der zweit-ähnlichste den Platz 2, etc. Die N ähnlichsten Fälle werden mitsamt ihren Ähnlichkeitswerten und ihren Rangfolgen von nun an mit *Top N* bezeichnet. Die Rangfolge der *Top N* und die entsprechenden Plätze ihrer Fälle wurden als Grundwahrheit für die weitere Fehlerberechnung genommen und in den folgenden Fehlerdefinitionen $true$ genannt.

Zu der wahren Rangfolge $true$ wurden die entsprechenden Plätze der *Top N* Fälle in Bezug gesetzt, welche sie in der Rangfolge bei den unterschiedlichen Methoden aus Abschnitt 2.2.4 belegt haben. Benannt sind die Plätze der *Top N* innerhalb der Rangfolge der Methoden allgemein mit $approx$.

In den meisten CBR Systemen aus Abschnitt 1.2 wird eine begrenzte Anzahl von Fällen, normalerweise zwischen 3 und 20 Fällen, für die Verarbeitung nach dem Retrieval herangezogen. Aufgrund dessen wurde die Evaluation in den Experimenten von Kapitel 3 für die *Top* 20 durchgeführt. In Abschnitt 3.4 wurden auch die *Top* 5 auf unterschiedlich großen Datenbank getestet und in Vergleich zu den *Top* 20 Ergebnissen gesetzt. Es zeigte sich, dass die Wahl der N der *Top N* keinen signifikanten Einfluss auf die Ergebnisse hat.

Fehlermaße - mittlerer absoluter Fehler und Standardabweichung

Für die Fehlerabschätzung wurde die Betragsdifferenz der *Top N* der beiden Ergebnisse *true* und *approx* verglichen. Dafür wurde zum einen der mittlere absolute Fehler (MAF) und zum anderen die Standardabweichung (STD) verwendet [109]. Der MAF wurde bereits in Evaluationen zur Leistung und Vergleich von CBR Systemen verwendet [15]. Des Weiteren bringt er einige Vorteile gegenüber dem Root-Mean-Square Error (RMSE), vor allem die einfache Interpretierbarkeit der Fehlerwerte in Bezug auf die Ergebnisse [121]. Die Standardabweichung ist ein klassisches Maß um die Abweichung der Ergebnisse von ihrem Mittelwert, in unserem Fall dem MAF, zu bestimmen. Ein kleiner MAF bedeutet nicht zwangsläufig eine kleine STD und umgekehrt. Die beiden Fehlermaße werden für jede Methode bei jeder Rate von fehlenden Daten berechnet und sind wie folgt definiert.

MAF ist durch den Mittelwert der Betragsdifferenz von *true* und *approx* gegeben mit $MAF = mean(|true - approx|)$ und spiegelt die Korrektheit der genauen Rangfolge wider. Je kleiner der MAF, desto besser ist die Methode. MAF gibt den Wert an, um den die Position eines Falles bei der Anwendung einer Methode von seiner wahren Position im Durchschnitt abweicht. z.B. Fall C ist als Ergebnis von *true* auf Postion 3, aber an Position 7 bei *approx*, dann ist seine absolute Differenz 4. Diese absolute Differenz wird für alle Fälle in den *Top N* berechnet und die Ergebnisse gemittelt.

STD ist als Standardabweichung der Betragsdifferenz von *true* und *approx* formuliert mit $STD = std(|true - approx|)$. Diese ist ein Maß für die Stabilität des entsprechenden MAF und stellt Abweichungen in den Ergebnissen aller für die Berechnung beinhaltenden Werte fest. Denn eine Methode mit einem kleinen MAF Wert kann durchaus eine große Spannweite falscher Werte aufweisen. Nur im Zusammenspiel beider Fehlermaße ist eine verlässliche Beurteilung der Methoden in Bezug auf die Rangfolge der ähnlichsten Fälle nach dem Retrieval möglich.

Mittlere Fehler über alle Raten

Für eine Vergleichsübersicht des Verhaltens aller Methoden innerhalb jedes Versuchs wurde der mittlere Fehler als Durchschnitt jeweils des MAF und der STD über die steigende Rate von fehlenden Daten berechnet. Um sinnvolle Werte zu erhalten, wurden nur die Raten von 10% bis 70% zur Mittelung der Fehler verwendet. Ab 70% sind für die meisten Methoden keine stabilen Vorhersagen anhand der Datenlage mehr möglich und auch in der Praxis wird dies meist vermieden. Der Umgang mit einem höheren Anteil fehlender Daten als 70% wird in der Diskussion Kapitel 4 besprochen. Es ist anzumerken, dass die Abbildungen trotzdem für eine informative Übersicht bis zu einer Rate von 90% gehen.

Der mittlere Fehler wird wie folgt berechnet. Sei mit V_i $(1 \leq i \leq 7)$ jeweils ein Versuch benannt z.B. V_1 ist der Versuch für Integer. Eine Methode sei M_j $(1 \leq j \leq 10)$, z.B. M_1 ist *List Delete*. Die Rate der fehlenden Daten wird mit R_k $(1 \leq j \leq 7)$ bezeichnet und geht von 10% bis 70% mit 10% Schritten. Für eine Rate R_k bezeichnet dann e_{ijk} den Fehler, entweder MAF oder STD, für einen Versuch V_i und eine Methode M_j. Dann ist der mittlere Fehler definiert als $E_{ij} = \sum_{k=1}^{7} e_{ijk}/7$.

Mittlere gewichtete Fehler für eine Rate über alle Versuche

Um die Leistung der Methoden über alle Versuche hinweg zusammenfassend vergleichen zu können, wurde in Abschnitt 3.2 und Abschnitt 3.3 der mittlere gewichtete Fehler von MAF oder STD, welcher über alle Versuche geht, für jede Rate von fehlenden Daten einzeln als Vergleichsmaß betrachtet.

Die Notationen und Bedingungen der Variablen für die Definition des Fehlers folgen dem letzten Abschnitt über den mittleren Fehler. Dann ist der mittlere gewichtete Fehler über aller Versuche V_i einer Rate R_k und einer Methode M_j wie folgt definiert. Zuerst wird der Fehler e_{ijk}, entweder MAF oder STD, mit der Summe der Fehler aller Methoden eines Versuchs gewichtet mit $e^*_{ijk} = e_{ijk} / \sum_{j=1}^{10} e_{ijk}$. Dann ist der mittlere gewichtete Fehler der Durchschnitt der gewichteten Fehler e^*_{ijk} aller Versuche gegeben

mit $E_{jk} = \sum_{i=1}^{7} e^*_{ijk}/7$. Er ist somit der Durchschnitt aus der Summe aller Versuche, die jeweils durch die Summe der Fehler aller Methoden eines Versuchs gewichtet wurden.

Eine Gewichtung der Fehler MAF und STD auf diese Art bringt einige Vorteile gegenüber dem einfachen Durchschnitt der Werte oder einer Normalisierung mit z.B. Min/Max. Sie vermeidet, dass verschiedene Versuche unterschiedlich gewichtet werden. Die Summe der mittleren gewichteten Fehler einer Rate ist immer 1 und die Werte sind relativ zueinander. Des Weiteren bewahrt die Gewichtung das fiktive Optimum einer perfekten Methode, da der Fehler einer perfekten Methode Null wäre. Eine einfache Normalisierung zu $[0, 1]$ würde dies nicht erfüllen, da hier die beste Methode der Versuche immer einen Fehler von Null haben würde, selbst wenn die geschätzten Werte nicht den wahren Werten der vollständigen Datenbank entsprächen. Auch wären die Werte dann nicht mehr relativ zueinander vergleichbar.

Mittlere gewichtete Fehler für alle Raten über alle Versuche

Die Vergleichsanalyse der verschiedenen Experimente aus den Abschnitten 3.2, 3.3 und 3.4, welche in Abschnitt 3.5 erfolgt, beruht auf dem mittleren gewichteten Fehler für alle Raten über alle Versuche. Dies ist der mittlere gewichtete Fehler des mittleren Fehlers über alle Raten, welche beide in den letzten beiden Abschnitten erläutert wurden. Frei gesprochen verdichtet er die Ergebnisse aller Raten von fehlenden Daten aus allen Versuchen für jede Methode.

Die Notationen und Bedingungen der Variablen für die Definition des Fehlers folgen dem vorletzten Abschnitt über den mittleren Fehler. Dann ist der mittlere gewichtete Fehler jeder Methode M_j für alle Raten R_k über aller Versuche V_i wie folgt definiert. Zuerst wird der mittlere Fehler $E_{ij} = \sum_{k=1}^{7} e_{ijk}/7$ für die Fehler e_{ijk}, entweder MAF oder STD, aller Raten R_k berechnet. Danach wird der mittlere Fehler E_{ij} mit der Gesamtsumme aller Methoden gewichtet zu $E^*_{ij} = E_{ij} / \sum_{j=1}^{10} E_{ij}$. Zum Abschluss wird der gewichtete mittlere Fehler E^*_{ij} über alle Versuche gemittelt zu $E^*_j = \sum_{i=1}^{7} E^*_{ij}/7$. Das Ergebnis ist der Fehler einer Methode über alle Raten und Versuche.

3 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Evaluation anhand von verschiedenen Experimenten vorgestellt, um das Potential und die Leistung des Multiple Retrieval Case-based Reasoning im Vergleich zu den existierenden Methoden nachzuweisen. Jedes Experiment teilt sich eine Gruppe von Versuchen auf, welche die verschiedenen Arten von Variablen und ihre Kombinationen untersuchen.

Der erste Abschnitt untersucht den Einfluss des Multiple Imputation Teils des MRCBR auf die Ergebnisse des Multiple Retrieval und Retrieval Pooling. Die nächsten Abschnitte betrachten das Verhalten des MRCBR und der konkurrierenden Methoden für verschiedene Typen von fehlenden Daten in einer oder mehreren Variablen. Danach werden unterschiedliche Größen der Datenbank und die Anzahl der verarbeiteten Fälle in einer komplexeren realistischen Umgebung in Betracht bezogen. Der vorletzte Abschnitt setzt die Ergebnisse der unterschiedlichen Experimente in Bezug zueinander und liefert einen Vergleich ihrer Auswirkungen auf die Methoden. Den Abschluss bildet ein experimenteller Nachweis, dass das Verhalten der Methoden in der MAR Umgebung zwischen MCAR und MNAR liegt.

3.1 Parameter der Multiple Imputation für das MRCBR

Der Offline-Teil des Multiple Retrieval Case-based Reasoning wird von der Multiple Imputation (MI) bestimmt und der Online-Teil mit dem Multiple Retrieval hängt direkt davon ab, was wiederum die Korrektheit der Rangfolge der ähnlichsten Fälle bestimmt.

Dies bedeutet das die Wahl der richtigen Parameter für die MI großen Einfluss auf das weitere Verhalten des MRCBR hat.

Daher wird in diesem Abschnitt der Einfluss der zwei wichtigsten Parameter, die Anzahl der imputierten Datenbanken und Iterationen, der MI auf die Ergebnisse des MRCBR betrachtet. Die Details und Eigenschaften dieser Parameter wurden in Abschnitt 1.3.4 erläutert. Nach der Auswertung der Ergebnisse werden die am geeignetsten befunden Parameter in den nächsten Abschnitten dieses Kapitels festgesetzt und für den MI-Teil des MRCBR verwendet.

3.1.1 Parametereinstellungen

Der eine Parameter bestimmt die Anzahl der imputierten Datenbanken, welche von der MI generiert werden und dann vom CBR-Teil des MRCBR im Multiple Retrieval weiterverarbeitet werden. In den folgenden Experimenten wurde eine steigende Anzahl (5, 10, 20, 40, 60, 80) von imputierten Datenbanken untersucht, um nachzuweisen, ob eine größere Anzahl von imputierten Datenbanken auf die finalen Ergebnisse des Retrieval eine Wirkung ausüben.

Der zweite Parameter betrifft die Anzahl der Iterationen innerhalb der MI. Die Iterationen sind eine mögliche Erweiterung der MI Schritte für eine bessere Verarbeitung von unvollständigen Datenbanken mit gelöschten Daten in mehreren Variablen. Iterativ werden die Ergebnisse der Imputation einer Variablen für die Imputation der nächsten Variable innerhalb eines Iterationsschrittes genutzt. Die erzeugten Werte aller imputierten Variablen innerhalb eines Iterationsschrittes werden daraufhin als Grundlage für die Berechnungen im nächsten Iterationsschritt verwendet, um eine Verfeinerung der Ergebnisse zu erzeugen. Für die Bestimmung des Parameters wurde eine aufsteigende Anzahl (1, 2, 4, 8, 16) von Iterationen verwendet.

3.1.2 Verhalten der Parameter

Die Experimente wurden für das *MRCBR CART* mit einem Anteil fehlender Daten mit einer Rate von 0.1, 0.4 und 0.7 auf dem künstlich wahren Fall Szenario aus Abschnitt 2.2.3 durchgeführt, da dieses mehrere unvollständige Variablen mit verschiedenen Typen von fehlenden Daten aufweist und damit die Wirkung beider Parameter möglichst ausreizt.

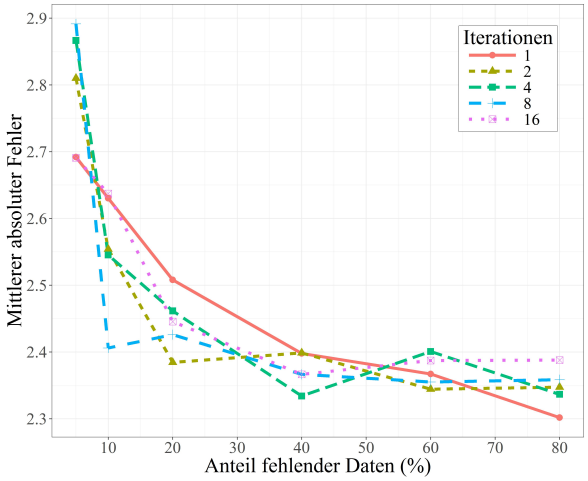
Die Ergebnisse für die verschiedenen Parameter des MI Teil des MRCBR und die drei Raten fehlender Daten sind in Abbildung 3.1 dargestellt.

3.1.3 Analyse der Ergebnisse

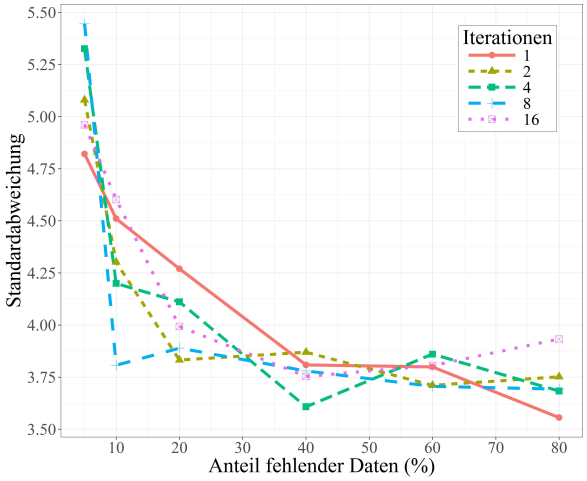
Der Verlauf der Kurven in den sechs einzelnen Abbildungen aus Abbildung 3.1 verhält sich sehr ähnlich. Die Kurven korrelieren sowohl für den mittleren absoluten Fehler und die Standardabweichung als auch für die drei unterschiedlichen Raten von fehlender Daten. Bei einer steigenden Anzahl von imputierten Datenbanken sinkt das Fehlermaß und die Kurve konvergiert hin zu einem kleineren Fehlerwert, was bedeutet, dass die Ergebnisse besser werden. Die steigende Anzahl von Iterationen hat keinen signifikanten positiven Effekt.

Mit einem Ansteigen der Rate von fehlenden Daten sieht man, dass die Kurven, welche für die verschiedenen Iterationen stehen, innerhalb jeder Abbildung sich immer mehr glätten und angleichen. Außerdem wird bei einer kleinen Rate von fehlenden Daten bereits mit einer kleinen Anzahl von imputierten Datenbanken eine große Verbesserung erzeugt, die auch mit einer wachsenden Anzahl von imputierten Datenbanken nicht tiefgreifend verbessert werden kann. Dieser Effekt verschiebt sich mit Ansteigen der Rate.

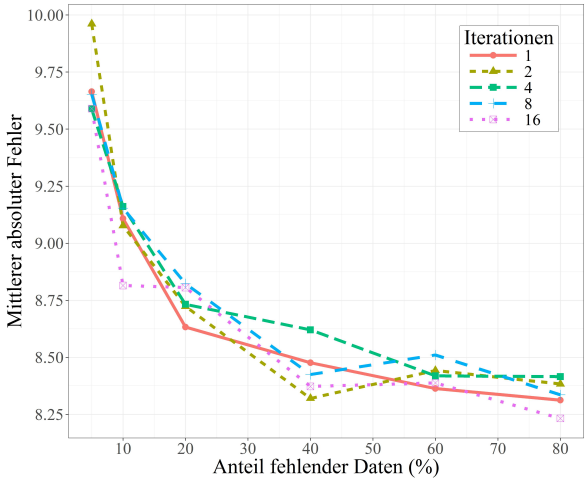
Von einer Anzahl von 5 auf 20 imputierten Datenbanken gibt es einen großen Sprung in der Verbesserung der Ergebnisse von bis zu 20% für alle Iterationen. Daraufhin flachen die Kurven der Fehler langsam ab und die Verbesserung der Ergebnisse zwischen 40 und 80 imputierten Datenbanken ist nur geringfügig. Somit zeigt sich, dass eine



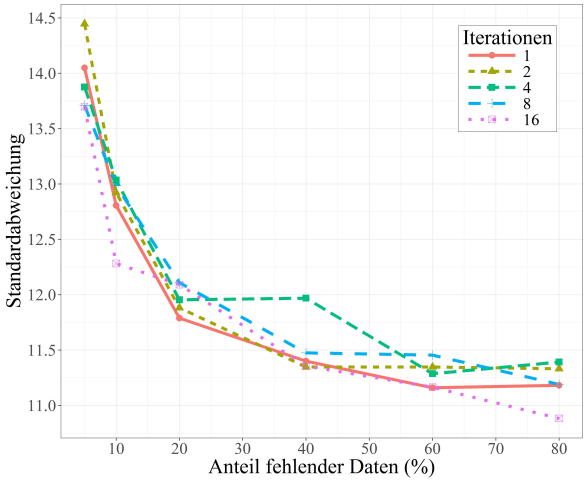
(a) Rate = 0.1/ mittlerer absoluter Fehler.



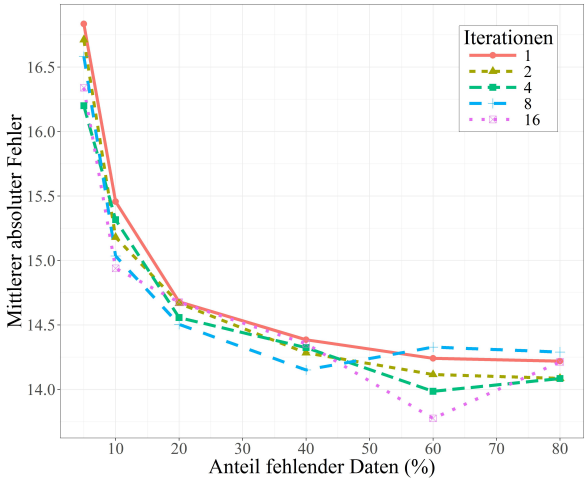
(b) Rate = 0.1/ Standardabweichung.



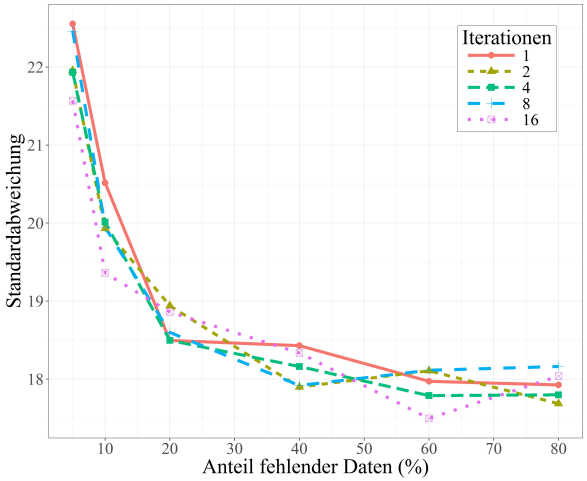
(c) Rate = 0.4/ mittlerer absoluter Fehler.



(d) Rate = 0.4/ Standardabweichung.



(e) Rate = 0.7/ mittlerer absoluter Fehler.



(f) Rate = 0.7/ Standardabweichung.

Abbildung 3.1: Verhalten des *MRCBR CART* für eine steigende Zahl von Iterationen und imputierten Datenbanken als Parameter des Multiple Imputation Teils für das künstlich wahre Fall Szenario mit einer Rate fehlender Daten von 0.1 in 3.1a und 3.1b, von 0.4 in 3.1c und 3.1d und von 0.7 in 3.1e und 3.1f.

höhere Anzahl von imputierten Datenbanken wie erwartet einen positiven Effekt auf die Leistung des MRCBR hat, da innerhalb der Multiple Retrieval Phase eine größere Bandbreite von möglichen Werten verarbeitet werden kann. Allerdings verschlechtern sich teilweise minimal die Ergebnisse ab einer Anzahl von 40 imputierten Datenbanken, um sich dann auf diesem Wert zu stabilisieren. Folglich hat eine zu hohe Anzahl imputierter Datenbanken nicht automatisch eine Verbesserung der Ergebnisse zufolge und sollte mit Vorsicht gewählt werden. Dies ist darauf zurückzuführen, dass eine äußerst hohe Anzahl von imputierten Datenbanken und die daraus resultierenden Ergebnisse des Multiple Retrieval das gemittelte Ergebnis im Retrieval Pooling zu sehr verwässern.

Die Anzahl der Iterationen hat dagegen keinen positiven Effekt auf die Leistung und ist sowohl in den verschiedenen Szenarien als auch für verschiedene Anzahlen von imputierten Datenbanken chaotisch. Dies deutet darauf hin, dass die Wahl der Iterationen auf die gestellte Fragestellung vorher abgestimmt sein sollte und je nach Datenbank anders ausfallen kann. Die Auswirkungen sind aber zu gering, in seltenen Fällen maximal 5%, als dass dieser Parameter einen entscheidenden Einfluss auf die finalen Ergebnisse ausübt.

3.1.4 Zusammenfassung

Die Anzahl der Iterationen hat keinen signifikanten Effekt auf die Leistung des MRCBR, wogegen die Anzahl der imputierten Datenbanken eine große Verbesserung bringt. Mit steigender Rate von fehlenden Daten lohnt sich auch eine höhere Anzahl von imputierten Datenbanken zu wählen.

Aufgrund dieser Beobachtungen wurde für die weiteren Experimente in diesem Kapitel 40 imputierte Datenbanken und 8 Iterationen für den MI Teil der beiden MRCBR Methoden verwendet. Diese beiden Werte für die Parameter zeigten in den Experimenten im Vergleich zu den anderen Parametern ein stabiles und solides Ergebnis.

Die Leistung des MRCBR kann bei einer anderen Wahl der Parameter noch gesteigert werden. Allerdings kostet die geringfügige Verbesserung der Ergebnisse zusätzliche Berechnungszeit und birgt eine gewisse Unsicherheit. Die Berechnungszeit kann mit ei-

ner geringeren Zahl von imputierten Datenbanken auch verkürzt werden. Dies geht zwar zu Lasten der Korrektheit der Ergebnisse des MRCBR, wie in den folgenden Abschnitten jedoch gezeigt wird, reicht dies immer noch für eine hervorragende Leistung im Vergleich zu den anderen Methoden aus. Es ist anzuraten für jede neue Datenbank die Parameter auf einem Teil dieser Datenbank als Training zu bestimmen und so optimal anzupassen.

3.2 MCAR Umgebung

Dieser Abschnitt untersucht das Verhalten des MRCBR und der konkurrierenden Methoden in der MCAR Umgebung für verschiedene Arten von Variablen und Kombinationen aus diesen. Die Versuche, welche aus den Variablen und Kombinationen bestehen, werden für einen Anteil fehlender Daten mit steigender Rate betrachtet.

Die Ergebnisse werden für ausgewählte Versuche und Methoden detailliert in Abbildungen gezeigt und für alle Versuche und Methoden zusammengefasst in Tabellen präsentiert. Eine ausführliche Analyse des Experiments erörtert die Ergebnisse in Bezug auf die Leistung der verschiedenen Methoden unter verschiedenen Gesichtspunkten, so dass eine möglichst vielfältige Sicht auf das Verhalten gegeben ist.

Der Aufbau des Experiments, Definitionen und Erläuterungen zu Fehlern und Methoden, der Variablen und fehlenden Daten, sowie der Abkürzungen finden sich in den jeweiligen Abschnitten des Abschnitt 2.2.

3.2.1 Verhalten ausgewählter Methoden für eine steigende Rate

Die einzelnen Abbildungen in 3.2 zeigen die Veränderung des mittleren absoluten Fehlers (MAF) und der Standardabweichung (STD) für die *Top* 20 Fälle einiger ausgewählter Methoden bei steigender Rate von fehlenden Daten innerhalb der Variablen, welche in dem fehlenden Daten Typ MCAR gelöscht wurden.

Zur Darstellung wurden exemplarisch die Variablen Integer, Kategoriale und die Kombination aus beiden ausgewählt. Die Methoden verhalten sich für Integer und

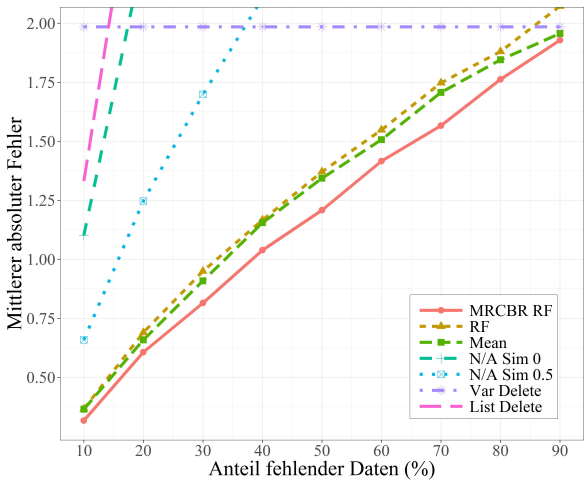
Float sehr ähnlich, wie sich in den folgenden ausführlichen Tabellen dieses Abschnittes zeigen wird. Auch alle Kombinationen mit Kategorialer folgen einem Muster. Daher sind die Abbildungen repräsentativ für alle erkennbaren Trends.

Des Weiteren werden nur solche Methoden in den Abbildungen vorgeführt, welche interessant im Vergleich sind und sich nicht in den Kurven zu sehr überschneiden. Diese sind *Var Delete*, *List Delete*, *N/A Sim 0*, *N/A Sim 0.5*, *Mean*, *RF*, und *MRCBR RF*. Jede Abbildung beinhaltete die Methode der Variablen Löschung. Sie ist die einfachste und schnellste Methode, sowohl in der Implementierung als auch der Durchführung, und nimmt innerhalb eines Versuchs für jede Rate von fehlenden Daten denselben Fehler an. So fungiert sie in den Anforderungen und Erwartungen als obere Schranke für alle anderen Methoden. Der listenweise Fallausschluss wurde für die Kombination aus Variablen nicht aufgeführt, da diese Methode mit steigender Rate von fehlenden Daten die Anzahl der Fälle in der Datenbank auf null schrumpft. Ab einer gewissen Rate ist keine Verarbeitung mehr möglich. Für eine übersichtliche Darstellung wurde aus den beiden Klassifikations-Algorithmen Entscheidungsbaum CART und Random Forest der letztere ausgewählt. Dieser schneidet im Großteil der Ergebnisse auch besser ab. Für das MRCBR wurde ebenfalls nur der Random Forest gezeigt. Im Gegensatz zu den singulären Imputations-Methoden mit Klassifikations-Algorithmen ist *MRCBR RF* in allen Experimenten dieser Arbeit *MRCBR CART* leicht überlegen.

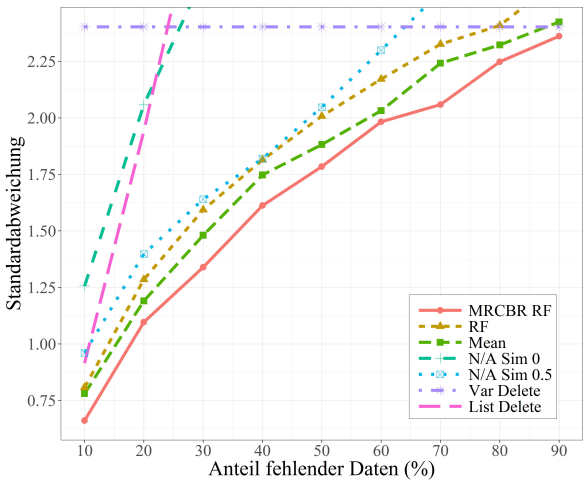
Die exemplarisch ausgewählten Versuche werden für die Integer-Variable in 3.2a und 3.2b, die kategoriale Variable in 3.2c und 3.2d und die Kombination aus beiden in 3.2e und 3.2f präsentiert.

3.2.2 Verhalten aller Methoden für die durchschnittliche Rate

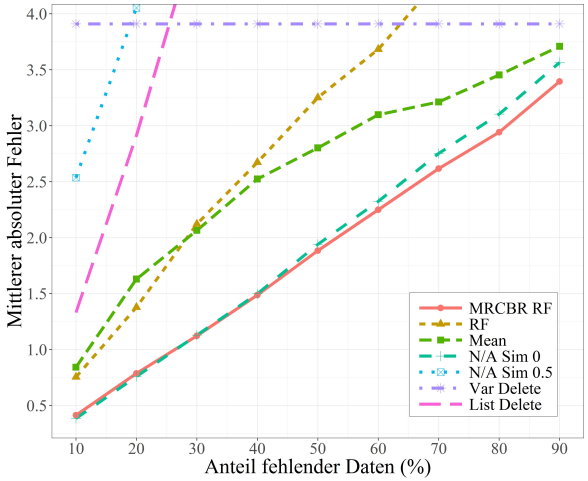
Die zusammengefassten Ergebnisse aller Versuche für alle untersuchten Methoden sind in den Tabellen 3.1 und 3.2 aufgeführt. Die dort verwendete mittlere Fehler ist definiert als der jeweilige Durchschnitt der MAF und STD über die steigende Rate von fehlenden Daten und ist in Abschnitt 2.2.5 erläutert. Er kann als *Area under the Curve* mit wenigen Stützstellen angesehen werden, z.B. der mittlere MAF einer Methode



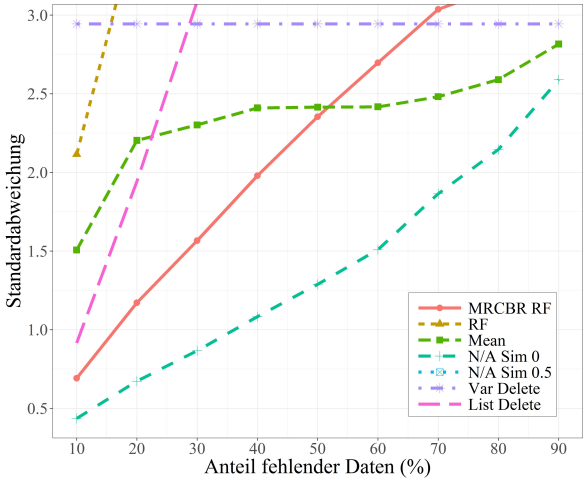
(a) Integer/ mittlerer absoluter Fehler.



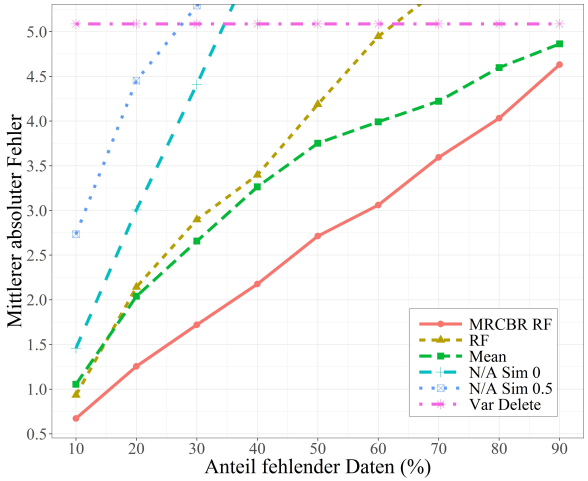
(b) Integer/ Standardabweichung.



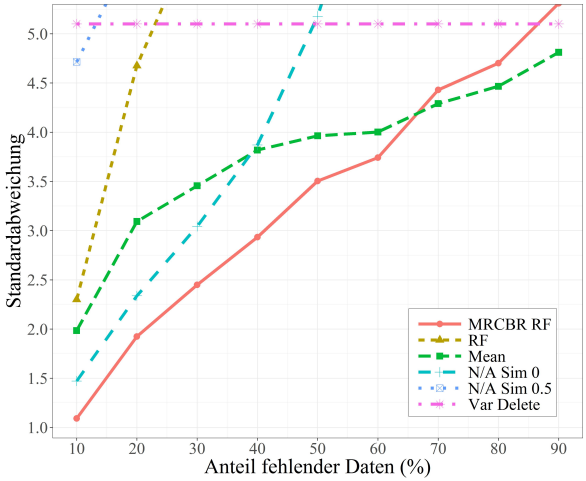
(c) Kategoriale/ mittlerer absoluter Fehler.



(d) Kategoriale/ Standardabweichung.



(e) Kombination/ mittlerer absoluter Fehler.



(f) Kombination/ Standardabweichung.

Abbildung 3.2: Korrektheit der Rangfolge ausgewählter Methoden in der MCAR Umgebung für die Integer-Variable 3.2a und 3.2b, die kategoriale Variable 3.2c und 3.2d und die Kombination aus beiden 3.2e und 3.2f in Bezug auf den MAF (links) und die STD (rechts) bei steigender Rate von fehlenden Daten.

angewendet auf die unvollständige Integer, als die Fläche unter der passenden Kurve von Abbildung 3.2a. Die besten drei Werte der Fehler für jeden Versuch sind in den Tabellen fettgedruckt. MAF ist der erste Wert, STD steht in Klammern dahinter.

In Tabelle 3.1 sind die mittleren Fehler für die einzelnen Variablen, Integer, Float und Kategoriale, dargestellt. Die Ergebnisse für ihren vier Kombinationen sind in Tabelle 3.2 aufgeführt.

Tabelle 3.1: Ergebnisse aller Methoden in Bezug auf den mittleren Fehler des MAF und der STD (in Klammern) über die steigende Rate von fehlenden Daten des Typs MCAR in Integer, Float und Kategorialer.

Methode	Int	Float	Kat
List Delete	9.61 (5.89)	9.61 (5.89)	9.61 (5.89)
Var Delete	1.99 (2.40)	1.93 (2.21)	3.91 (2.94)
Pair Delete	1.12 (1.62)	1.22 (1.63)	7.66 (7.25)
N/A Sim 0	4.66 (3.48)	4.55 (3.40)	**1.54** (**1.10**)
N/A Sim 0.5	1.94 (1.82)	1.86 (1.76)	4.66 (4.75)
Mean	**1.09** (**1.62**)	1.10 (1.47)	2.31 (**2.25**)
CART	1.14 (1.74)	1.13 (1.55)	2.56 (4.20)
RF	1.12 (1.72)	**1.05** (**1.40**)	2.60 (5.10)
MRCBR CART	**1.02** (**1.54**)	**1.04** (**1.39**)	**1.64** (2.41)
MRCBR RF	**1.00** (**1.50**)	**0.99** (**1.30**)	**1.51** (**1.93**)

Tabelle 3.2: Ergebnisse aller Methoden in Bezug auf den mittleren Fehler des MAF und der STD (in Klammern) über die steigende Rate von fehlenden Daten des Typs MCAR in den Kombinationen aus Integer, Float und Kategorialer.

Methode	I.F.	I.K.	F.K.	I.F.K.
Var Delete	3.12 (3.94)	5.09 (5.10)	5.06 (4.93)	6.46 (6.90)
Pair Delete	1.90 (2.48)	8.06 (8.36)	7.97 (8.16)	8.34 (9.02)
N/A Sim 0	8.16 (7.42)	6.13 (4.66)	5.99 (4.56)	9.69 (8.90)
N/A Sim 0.5	2.99 (2.83)	5.36 (6.10)	5.26 (5.94)	6.10 (7.15)
Mean	1.78 (2.44)	**3.00** (**3.52**)	**2.95** (**3.32**)	**3.52** (**4.25**)
CART	1.80 (2.47)	3.33 (5.37)	3.38 (5.29)	3.95 (6.10)
RF	**1.74** (**2.36**)	3.42 (6.24)	3.33 (5.99)	4.03 (7.07)
MRCBR CART	**1.62** (**2.12**)	**2.31** (**3.22**)	**2.31** (**3.17**)	**2.80** (**3.77**)
MRCBR RF	**1.60** (**2.11**)	**2.17** (**2.87**)	**2.11** (**2.63**)	**2.65** (**3.39**)

3.2.3 Analyse der Ergebnisse

Die Analyse der Tabellen 3.1 und 3.2 offenbart, dass sich die Methoden für den mittleren MAF und die mittlere STD bei Integer und Float sehr ähnlich verhalten. Die Ergebnisse der Methoden bei der kategoriale Variable und den Kombinationen daraus weichen jedoch deutlich davon ab. Für die mittlere STD gibt es leichte Unterschiede zum mittleren MAF, aber die Ergebnisse korrelieren größtenteils. Bei steigender Anzahl von Variablen, sowie einer steigenden Rate von fehlenden Daten, verschlechtern sich bei allen Methoden die beiden Fehler.

Verhalten des mittleren absoluten Fehlers

Die besten Ergebnisse in jedem Versuch für den mittleren MAF, sowohl für einzelne Variablen, als auch für Kombination daraus, erreichen die beiden Methoden *MRCBR CART* und *MRCBR RF*. Unter diesen Methoden schneidet *MRCBR RF* mit kleinem Abstand besser ab, was auch in allen weiteren Versuchen dieses Kapitels der Fall sein wird. Da *MRCBR CART* immer auf Platz zwei liegt, wird es nur bei Bedarf erwähnt. Für die Integer und Float liegt der mittlere MAF von *MRCBR RF* bei 1.0 und für die kategoriale Variable bei 1.5. Für mehr als eine Variable reichen die Werte bis 2.6 für die Kombination aus allen drei Variablen. Es sei daran erinnert, dass der mittlere MAF der Tabellen 3.1 und 3.2, auf welche sich die Analyse bezieht, bedeutet, um wie viele Plätze die Methode von ihrem wahren Rangfolgeplatz im Durchschnitt über alle Raten von fehlenden Daten abweicht. Die Abbildungen 3.2a, 3.2c und 3.2e werden zur tieferen Analyse zu Rate gezogen. Es ist ersichtlich, dass der Abstand von *MRCBR RF* in Bezug auf das Fehlermaß zu den anderen Methoden bei steigender Anzahl von Variablen mit fehlenden Daten, wie zum Beispiel I.F.C., größer wird. *MRCBR RF* erzielt eine eindeutige Verbesserung in diesem Fall, während die anderen Methoden Einbußen in ihrer Leistung erleiden. In Abbildung 3.2e wird dies im Kontrast zu Abbildung 3.2a deutlich sichtbar. In den Abbildungen sieht man auch, dass *MRCBR RF* interessanter Weise ein fast lineares Wachstum in seinen beiden Fehlern aufweist, wohingegen besonders die Kurve des *Mean* eine Abflachung beschreibt.

Für die weitere Analyse der Methoden werden die Ergebnisse für eine leichtere Übersicht in Prozentwerten im Abstand zu *MRCBR RF* wiedergegeben, z.B. *Methode Y* mit mittlerem MAF=3.0 ist 50% schlechter als *MRCBR RF* mit mittlerem MAF=2.0. Wenn eine Methode bessere Ergebnisse als *MRCBR RF* aufweist, wird dies explizit hervorgehoben. Dasselbe gilt auch für die später evaluierten Werte der mittleren STD.

Die stabilsten Werten für den mittleren MAF außer den *MRCBR* Methoden weist *Mean* auf. Dieser ist immer unter den Top 4 der besten Methoden in allen Versuchen. Für Integer und Float ist er nur 10% schlechter und fällt auf 50% für die kategoriale Variable. Für alle Kombinationen mit der kategorialen Variable belegt er sogar den dritten Platz.

Ähnlich wie *Mean* verhalten sich die Klassifikations-Algorithmen *RF* und *CART*. Sie sind immer in den Top 5 der besten Methoden. Für Float und seine Kombinationen ist *RF* auf dem dritten Platz. Allerdings sind beide Methoden für kategoriale Variable 70% schlechter als *MRCBR RF* und für Kombinationen daraus immer noch 20% schlechter als *Mean*.

Pair Delete zeigt gute Ergebnisse von bis zu 13% für die numerischen Variablen und ihre Kombinationen, die nah an *Mean* und den Klassifikations-Algorithmen liegen. Doch für die kategoriale Variable scheitert die Methode mit 400% und auch für die Kombination daraus ist sie um die 200% schlechter als *MRCBR RF*.

List Delete divergiert aus den oben genannten Gründen mit steigender Rate von fehlenden Daten sehr schnell, gleichgültig ob für numerische oder kategoriale Variablen, wie in Abbildung 3.2c ersichtlich ist. Es ist mit großem Abstand die schlechteste Methode und für eine höhere Anzahl von Variablen mit fehlenden Daten nicht verwendbar. Wie in Abschnitt 1.3.2 erwähnt ist *List Delete* eine der meist genutzten Methoden in der Praxis.

Die universelle CBR Methode *N/A Sim 0* weist für Integer und Float und ihrer Kombination die schlechtesten Ergebnisse auf, nur gefolgt von *List Delete*, und ist über 360% schlechter. Ganz anders verhält sich *N/A Sim 0* für die kategoriale Variable, in der es fast genauso gut wie *MRCBR RF* abschneidet und beide einen Abstand von 50%

zur nächstbesten Methode *Mean* haben. Dieser Vorsprung geht für die Kombinationen aus kategorialer und numerischen Variablen verloren, so dass *N/A Sim 0* bis zu 266% schlechter ist. In Abbildung 3.2c sieht man Kluft zwischen *MRCBR RF*, *Mean* und *N/A Sim 0* sehr eindeutig im Kontrast zu dem Verhalten in Abbildung 3.2a.

N/A Sim 0.5 dagegen ist mit 90% bis 210% in allen Versuchen im hinteren Feld der Methoden und besonders bei der kategorialen Variablen und Kombination daraus schlecht. Die Werte sind sehr nah an denen von *List Delete* .

Var Delete ist mindestens 95% schlechter als *MRCBR RF*. Wie bereits erwähnt, wird es aufgrund seiner statischen Werte als obere Schranke für alle Methoden betrachtet. Diese Schranke wird von *N/A Sim 0*, *N/A Sim 0.5*, *Pair Delete* und *List Delete*, aber auch von *RF* teilweise schon früh durchbrochen werden, wie Abbildung 3.2e zeigt. Dagegen konvergieren *MRCBR RF* und *Mean* zu dieser Schranke mit steigender Rate von fehlenden Daten, wie in allen Abbildungen gut sichtbar.

Verhalten der Standardabweichung

Die Auswertung der Tabellen zeigt, dass die meisten Werte der mittleren STD sich ähnlich wie für den mittleren MAF verhalten. Doch es gibt ein paar Unterschiede, die nicht unerwähnt bleiben sollen, da sie einen Trend im Verhalten der Methoden aufzeigen, der auch in den nächsten Abschnitten bestehen bleibt. Von genauen Werten wird in dieser Vergleichsanalyse abgesehen und nur solche Ergebnisse hervorgehoben, die einen Unterschied zu der Analyse des MAF darstellen.

Insgesamt sind die Abstände der Methoden zur besten Methode *MRCBR RF* geringer als beim mittleren MAF, aber immer noch deutlich. Besonders *Mean* behält sein stabiles Verhalten und liegt noch enger an *MRCBR RF*. Seine gemittelte STD ist im Vergleich zu den anderen Methoden bei der Kategorialen relativ gering. In Abbildung 3.2d und Abbildung 3.2f sieht man, dass *MRCBR RF* von *Mean* ab einer Rate von 50% fehlender Daten überholt wird. Davor schneidet es jedoch sehr viel besser ab.

Die Klassifikations-Algorithmen sind wieder ähnlich von ihrem Verhalten und *RF* und *CART* weisen eine stabile STD auf. Jedoch für die Kategoriale und alle ihrer

Kombinationen ist ihre gemittelte STD teilweise doppelt so hoch wie ihr gemittelter MAF, was für keine der anderen Methoden gilt.

N/A Sim 0 hat für die kategoriale Variable den besten Fehler, welcher sogar um 43% kleiner ist als *MRCBR RF*. Abbildung 3.2d verdeutlicht dies. Für die Kombinationen mit der kategorialen Variablen behält *N/A Sim 0* gute Werte, allerdings bei weitem nicht mehr besser als *MRCBR RF*. Interessant ist, dass *N/A Sim 0* im Gegensatz zu allen anderen Methoden einen kleineren Wert in der mittleren STD hat, als für den mittleren MAF. Nur *List Delete* hat für einzelne Variablen dasselbe Verhalten.

Bei der mittleren STD sticht *N/A Sim 0.5* für numerische Variablen und ihre Kombination hervor und kommt fast an die Werte der drei besten Methoden heran, wie in Abbildung 3.2b sichtbar ist.

Bis auf diese Ausnahmen verhalten sich die restlichen Methoden ähnlich wie bei dem mittleren MAF.

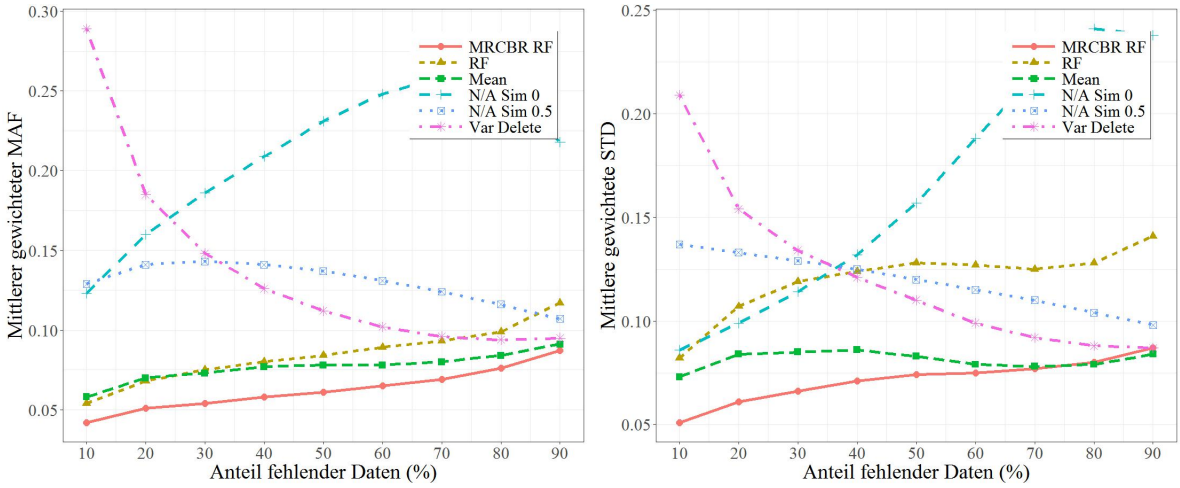
3.2.4 Vergleichsübersicht

Zum Abschluss der Analyse des Verhaltens der Methoden in der MCAR Umgebung wird eine Gesamtübersicht der Leistung präsentiert, welche die besprochenen Ergebnisse innerhalb einer Abbildung durch einen relativen Fehler verdichten soll. Der Fehler basiert auf einem mittleren gewichteten Fehler für den MAF und die STD über alle Versuche für jede Methode, um die Ergebnisse der Versuche für jede Rate von fehlenden Daten in relativem Bezug zusammenzufassen. Die Erläuterung der mittleren gewichteten Fehler ist in Abschnitt 2.2.5 gegeben.

Abbildung 3.3 zeigt die beiden mittleren gewichteten Fehler für den MAF und die STD bei einer steigenden Rate von fehlenden Daten für die selbe Auswahl von Methoden wie in Abschnitt 3.4.1.

MRCBR RF ist eindeutig die beste Methode in beiden Fehlerbetrachtungen. *Mean* liegt dicht darüber und beide verschlechtern sich leicht für eine steigende Rate von fehlenden Daten. *Var Delete* und *N/A Sim 0.5* haben für eine kleine Rate keine guten Ergebnisse, flachen aber mit steigernder Rate ab. Es ist ersichtlich, dass für die beiden

mittleren gewichteten Fehler fast alle Methoden mit steigender Rate von fehlenden Daten am Ende der 90% zusammenlaufen und eine ähnliche Leistung aufweisen, die sich *Var Delete* annähert. Nur *N/A Sim 0* schert von diesem Verhalten für den mittleren gewichteten MAF aus und wird immer schlechter. Bei der mittleren gewichteten STD verhält es sich ähnlich, nur bei dieser bricht auch *RF* am Ende aus, welcher für den mittleren gewichteten MAF noch nah an *MRCBR RF* und *Mean* lag. Abschließend lässt sich sagen, dass bei einer sehr hohen Rate von fehlenden Daten die meisten Methoden immer weniger eine verlässliche Leistung bringen, die einer völligen Löschung der Variable mit *Var Delete* gleichkommt. Doch auch in diesem extremen Fall haben *MRCBR RF* und *Mean* die besten Ergebnisse.



(a) Für den mittleren gewichteten MAF. **(b)** Für die mittlere gewichtete STD.

Abbildung 3.3: Korrektheit der Rangfolge der ausgewählten Methoden in der MCAR Umgebung für alle Versuche in Bezug auf den mittleren gewichteten Fehler des MAF (links) und der mittleren gewichteten STD (rechts) bei steigender Rate von fehlenden Daten.

3.2.5 Zusammenfassung

Nach der Analyse der verschiedenen Versuche, welche in den Tabellen und Abbildungen dargestellt wurden, haben sich mit Hilfe unterschiedlicher Fehler Betrachtungen klare Ergebnisse im Verhalten der unterschiedlichen Methoden gezeigt.

Zusammenfassend lässt sich sagen, dass *MRCBR RF* in der MCAR Umgebung die beste Methode darstellt. Besonders für eine höhere Anzahl von Variablen mit fehlenden Daten hat *MRCBR RF* einen großen Vorsprung zu den restlichen Methoden. Seine Schwestermethode *MRCBR CART* folgt knapp hinter seinen Ergebnissen. Beide zeigen in allen Versuchen eine Leistung, die die anderen Methoden übertrifft. Dies gilt sowohl unter den Gesichtspunkten des mittleren absoluten Fehlers, als auch für die Standardabweichung. Nur für eine einzelne Kategoriale ist die universelle CBR-Methode *N/A Sim 0* in beiden Fehlern die beste Methode, ansonsten weist sie keine guten Ergebnisse auf. Allerdings tun sich alle Methoden mit der Kategorialen schwer. Als einzige andere Methode sticht der *Mean* hervor, der ebenfalls gute Ergebnisse erreicht und stabil in jedem Fehler bleibt. Besonders die klassischen Methoden, wie *List Delete* und *Pair Delete*, konnten nicht überzeugen und scheiterten.

3.3 MNAR Umgebung

Analog zu den Experimenten der MCAR Umgebung in Abschnitt 3.2 wird in diesem Abschnitt für die Evaluation des MRCBR die Leistung der unterschiedlichen Methoden in verschiedenen Versuchen unter der MNAR Umgebung geprüft. Die Versuche, Variablen und die Kombinationen daraus, welche in den entsprechenden Abbildungen und Tabellen präsentiert werden, sind konform mit denen aus dem letzten Abschnitt für die MCAR Umgebung, so dass ein klarer Vergleich gezogen werden kann.

3.3.1 Verhalten ausgewählter Methoden für eine steigende Rate

Die einzelnen Abbildungen in Abbildung 3.4 präsentieren das Verhalten des MRCBR und einiger essentiellen Methoden unter dem mittleren absoluten Fehler (MAF) und der Standardabweichung (STD) für eine steigende Rate von fehlenden Daten mit des Typ MNAR in ausgewählten Variablen. Die Begründung für die Auswahl der Methoden und Variablen findet sich in Abschnitt 3.4.1.

Die Ergebnisse für die Integer-Variable werden in 3.4a und 3.4b, für die kategoriale Variable in 3.4c und 3.4d und für die Kombination aus beiden in 3.4e und 3.4f gezeigt.

3.3.2 Verhalten aller Methoden für die durchschnittliche Rate

Die Tabellen 3.3 und 3.4 führen die mittleren Fehler von MAF und STD (in Klammern) für alle Versuche mit allen untersuchten Methoden auf. Eine detaillierte Erläuterung des mittleren Fehlers findet sich in Abschnitt 2.2.5.

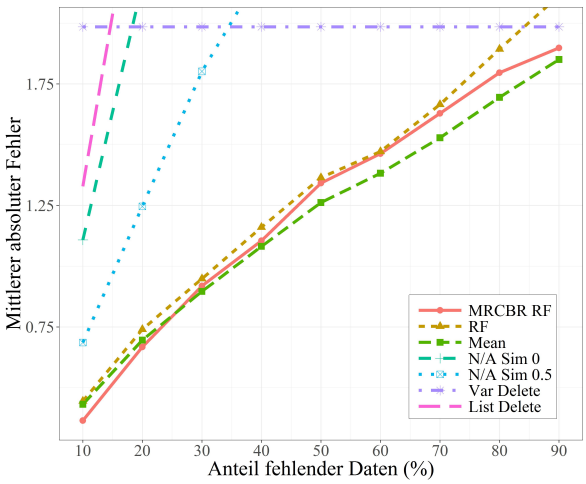
Die Ergebnisse für die Integer, Float und Kategoriale befinden sich in Tabelle 3.3 und für die vier Kombinationen aus diesen in Tabelle 3.4.

Tabelle 3.3: Ergebnisse aller Methoden in Bezug auf den mittleren Fehler des MAF und der STD (in Klammern) über die steigende Rate von fehlenden Daten des Typs MNAR in Integer, Float und Kategorialer.

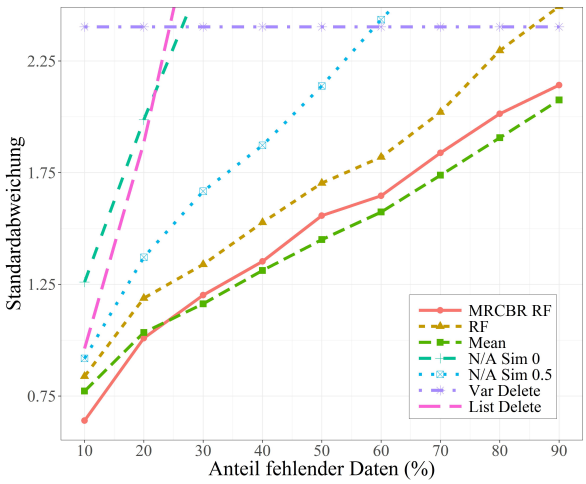
Methode	Int	Float	Kat
List Delete	10.03 (6.16)	9.99 (6.07)	12.80 (7.33)
Var Delete	1.99 (2.40)	1.93 (2.21)	3.91 (2.94)
Pair Delete	**1.09** (**1.35**)	**1.09** (**1.33**)	7.25 (6.98)
N/A Sim 0	4.90 (3.76)	4.62 (3.55)	**1.62** (**1.16**)
N/A Sim 0.5	1.98 (1.87)	1.81 (1.72)	4.51 (4.57)
Mean	**1.04** (**1.29**)	**1.04** (**1.29**)	2.55 (**2.25**)
CART	1.16 (1.53)	1.21 (1.52)	2.98 (3.50)
RF	1.11 (1.49)	1.12 (1.41)	4.18 (5.75)
MRCBR CART	1.12 (1.39)	1.12 (1.37)	**2.43** (2.81)
MRCBR RF	**1.07** (**1.32**)	**1.06** (**1.29**)	**2.31** (**2.40**)

3.3.3 Analyse der Ergebnisse

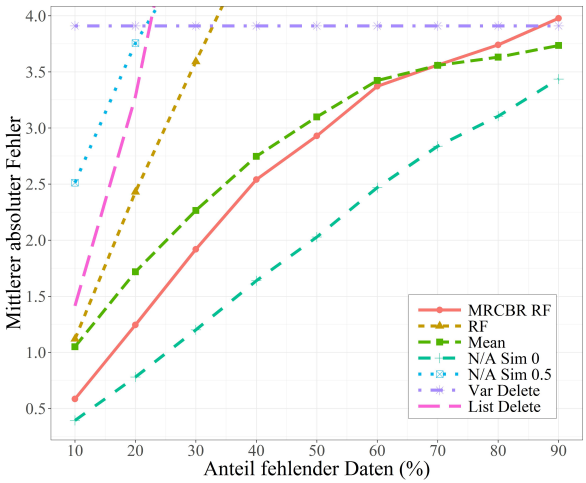
Im Vergleich zu der MCAR Umgebung 3.3 gibt es einige signifikante Unterschiede im Verhalten der Methoden in der MNAR Umgebung sowohl in der Betrachtung unter dem MAF als auch der STD. Dies sticht vor allem für die kategoriale Variable und ihre Kombinationen heraus. Integer und Float zeigen wieder ein ähnliches Verhalten.



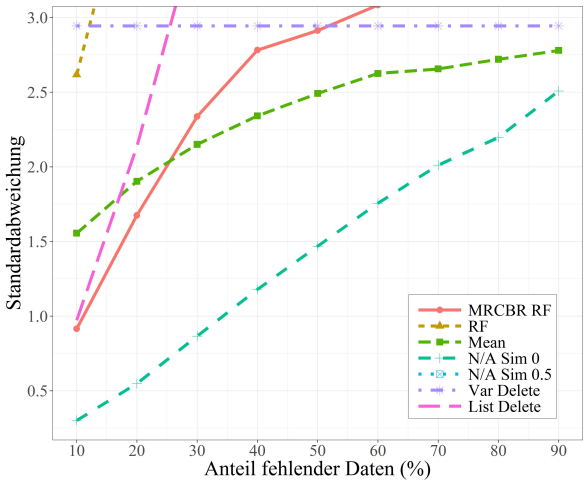
(a) Integer/ mittlerer absoluter Fehler.



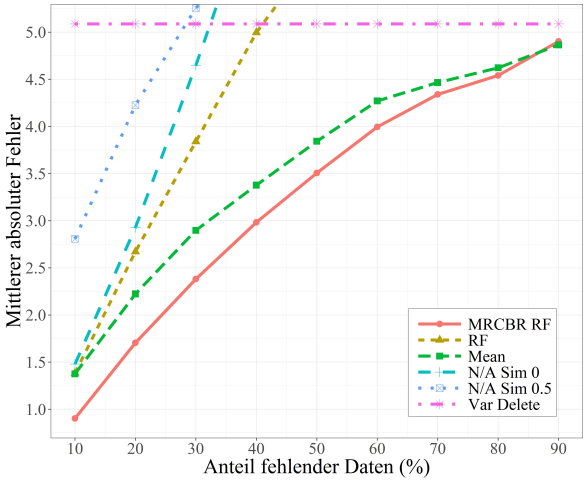
(b) Integer/ Standardabweichung.



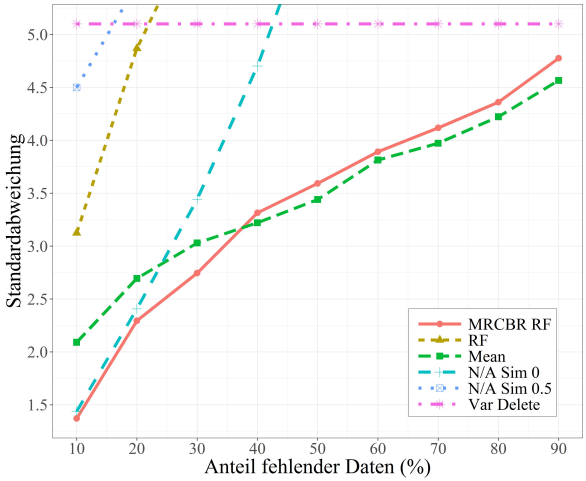
(c) Kategoriale/ mittlerer absoluter Fehler.



(d) Kategoriale/ Standardabweichung.



(e) Kombination/ mittlerer absoluter Fehler.



(f) Kombination/ Standardabweichung.

Abbildung 3.4: Korrektheit der Rangfolge ausgewählter Methoden in der MNAR Umgebung für die Integer-Variable 3.4a und 3.4b, die kategoriale Variable 3.4c und 3.4d und die Kombination aus beiden 3.4e und 3.4f in Bezug auf den MAF (links) und die STD (rechts) bei steigender Rate von fehlenden Daten.

Tabelle 3.4: Ergebnisse aller Methoden in Bezug auf den mittleren Fehler des MAF und der STD (in Klammern) über die steigende Rate von fehlenden Daten des Typs MNAR in den Kombinationen aus Integer, Float und Kategorialer.

Methode	I.F.	I.K.	F.K.	I.F.K.
Var Delete	3.12 (3.94)	5.09 (5.10)	5.06 (4.93)	6.46 (6.90)
Pair Delete	1.72 (2.06)	7.49 (7.66)	7.58 (7.73)	7.85 (8.35)
N/A Sim 0	8.52 (7.69)	6.56 (5.32)	6.25 (5.07)	10.41 (9.60)
N/A Sim 0.5	3.06 (2.99)	5.37 (5.97)	5.35 (5.88)	6.39 (7.25)
Mean	**1.64** (**2.01**)	**3.21** (**3.18**)	**3.19** (**3.23**)	**3.76** (**3.98**)
CART	1.86 (2.32)	3.59 (4.50)	3.73 (4.75)	4.35 (5.53)
RF	1.78 (2.22)	4.50 (6.51)	4.40 (6.23)	5.10 (7.16)
MRCBR CART	**1.71** (**2.04**)	**2.94** (**3.40**)	**2.98** (**3.50**)	**3.52** (**4.12**)
MRCBR RF	**1.64** (**1.96**)	**2.83** (**3.05**)	**2.86** (**3.13**)	**3.39** (**3.72**)

Verhalten des mittleren absoluten Fehlers

MRCBR RF hat für die numerischen einzelnen Variablen einen mittleren MAF von 1.07, für die kategoriale Variable liegt der Fehler bei 2.31 und für die Kombination aus allen drei steigt es bis zu 3.39. Dies zeigt, dass sich die Werte für Kategoriale und die Kombinationen daraus im Kontrast zu der MCAR Umgebung verschlechtern. Fast alle Methoden sind von diesem Effekt betroffen. *MRCBR CART* ist für MNAR nicht immer die zweitbeste Methode, wie zuvor bei MCAR, denn die besten Methoden für MNAR sind viel dichter beisammen und die Abstände zwischen diesen geringer. So kommt es, dass zum Beispiel für Integer die sechs besten Methoden nur einen Unterschied von 10% aufweisen. Jedoch bleibt *MRCBR CART* bei Kombinationen aus den Variablen immer in den Top 3. *MRCBR RF* ist weiterhin für Kombinationen auf dem ersten Platz und kann seinen Vorsprung gegenüber den anderen Methoden für eine steigende Anzahl von gelöschten Variablen immer weiter ausbauen. Für die einzelnen Variablen fällt es allerdings auf den zweiten Platz, für numerische Variablen knapp, für kategoriale Variable deutlich. *MRCBR RF* wird wieder als Referenzmethode genommen und die Ergebnisse der konkurrierenden Methoden in prozentualem Bezug zu seinen Ergebnissen gesetzt, mit Methode Y ist x% schlechter oder besser als *MRCBR RF*.

Mean ist für beide numerische Variable für den mittleren MAF 3% besser als *MRCBR RF* und auch für die Kombination aus beiden genauso gut wie *MRCBR RF*. Abbildung 3.4a zeigt diese Beobachtung. Auch bei den anderen Versuchen behält *Mean* seine guten Werte und ist maximal 10% schlechter als *MRCBR RF*. Nur für die Kategoriale fällt er auf den vierten Platz, aber ist ansonsten immer unter den drei besten Methoden. Zusammen mit *MRCBR RF* hat er teilweise einen großen Abstand zu den nächstfolgenden Methoden, wie in Abbildung 3.4e gut erkennbar.

Die Klassifikations-Algorithmen *RF* und *CART* verhalten sich unter dem mittleren MAF nicht mehr so ähnlich zueinander wie in der MCAR Umgebung, sind aber wie zuvor in den Top 5 der besten Methoden. Wenn die Kategoriale Teil des Versuches ist, wird *RF* schlechter als *CART*. Für die einzelne kategoriale Variable sogar <80%, was in Abbildung 3.4c durch sein frühes erreichen der *Var Delete* Linie sichtbar ist. Beide Methoden sind für numerische Variablen und ihre Kombination eng an den besten Methoden, aber für die Kategoriale und alle Kombination daraus verschlechtern sich ihre Werte mindestens um 30%.

N/A Sim 0 schneidet für die numerischen Variablen und ihre Kombination mit bis zu 420% sehr schlecht ab. Im Gegensatz dazu ist *N/A Sim 0* für die kategoriale Variable mit Abstand die beste Methode und 30% besser als *MRCBR RF*. Abbildung 3.4c verdeutlicht den Vorsprung gegenüber *Mean* und *MRCBR RF*. Dieses Ergebnis verbessert auch das Verhalten für Kombinationen daraus, aber nicht zufriedenstellend (> 130%).

N/A Sim 0.5 zeigt in Kontrast zur MCAR Umgebung ein stabiles Verhalten in allen Versuchen betreffs seines Abstands zu *MRCBR RF* und ist durchschnittlich 80% schlechter. Auch für die kategoriale Variable bleibt dieses Verhalten bestehen.

Pair Delete gehört für Integer, Float und ihre Kombination zu den Top 3. Auch wenn das Feld der besten Methoden bei diesen drei Versuchen äußerst eng ist. Für die kategoriale Variable und auch alle Kombinationen fällt *Pair Delete* jedoch stark zurück (> 130%).

Die Änderung der Umgebung hat keinen Einfluss auf *List Delete*, denn es leidet weiterhin an dem Problem des Datenverlustes durch die Löschung der Fälle ($> 540\%$). Für mehrere Variablen mit fehlenden Daten ist es nicht verwendbar.

Var Delete hat selbstverständlich denselben mittleren MAF wie für den Versuchen in der MCAR Umgebung und agiert weiterhin als obere Schranke für die anderen Methoden. Es ist mindestens 70% schlechter als *MRCBR RF*. Allerdings konvergieren für eine steigende Rate von fehlenden Daten zu mindestens immer die zwei besten Methoden jedes Versuches hin zu seiner Schranke, was in allen Abbildungen von Abbildung 3.4 gut sichtbar ist.

Verhalten der Standardabweichung

Die Analyse der mittleren STD aufgrund der Tabellen offenbart, dass die Werte mit denen des mittleren MAF korrelieren. Mehr als es bei der MCAR Umgebung der Fall gewesen ist. Die Rangliste bezüglich der Leistung der Methoden entspricht der beim mittleren MAF. Abbildung 3.4a und Abbildung 3.4b spiegeln dies grafisch wider. Nur für die Kategoriale gibt es ein paar Unterschiede. Es werden in der Analyse nur die Ergebnisse besprochen, die in Kontrast zu der vorherigen Analyse des MAF stehen.

MRCBR RF ist für alle Kombinationen auf dem ersten Platz und bei den numerischen Variablen auf dem zweiten Platz. Für die kategoriale Variable fällt es auf den dritten Platz und stößt bei einer 50% Rate fehlender Daten durch die Schranke von *Var Delete*. *MRCBR CART* ist für Kombinationen in den Top 3 der besten Methoden, ansonsten auf dem vierten Platz.

Mean ist für die numerischen Variablen nahezu gleichauf mit *MRCBR RF* und bei der kategorialen Variable 6% besser. Sonst ist *Mean* immer auf dem zweiten Platz. Bei den Kombinationen liegen *Mean* und *MRCBR RF* eng beieinander mit Abstand zu den restlichen Methoden außer *MRCBR CART*. In Abbildung 3.4f sieht man, dass ab einer 40% Rate fehlender Daten die beiden ihre Führung wechseln.

N/A Sim 0 ist für die kategoriale Variable in Bezug auf die mittlere STD noch stärker als für den mittleren MAF und sogar 52% besser als *MRCBR RF*. Abbildung 3.4d

demonstriert deutlich den Abstand zu *MRCBR RF* und *Mean*. *N/A Sim 0* ist wie im MCAR Experiment die einzige Methode, deren mittlere STD kleiner ist als ihr mittlerer MAF.

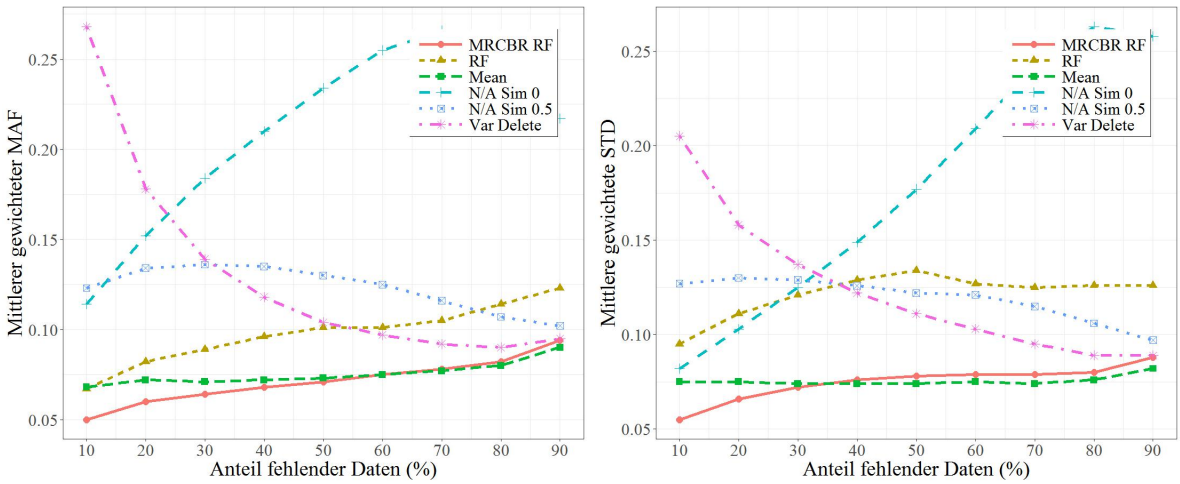
Alle anderen Methoden haben schon bei einer frühen Rate von fehlenden Daten unzureichende Ergebnisse, noch ausgeprägter als beim mittleren MAF. In Abbildung 3.4f durchbrechen *RF*, *N/A Sim 0* und *N/A Sim 0.5* die Linie von *Var Delete* bei maximal 40%. Für die kategorische sind *RF* und *N/A Sim 0.5* sogar sofort schlechter als *Var Delete*.

3.3.4 Vergleichsübersicht

Eine zusammenfassende Übersicht wird in Abbildung 3.5 für die Selektion von Methoden aus Abschnitt 3.3.1 gegeben. Wie schon im letzten Abschnitt 3.3 werden in dieser Betrachtung die Methoden unter einem mittleren gewichteten Fehler für den MAF und die STD in Verbindung gesetzt und können so relativ verglichen werden. Die Erläuterung für diesen Fehler finden sich in Abschnitt 2.2.5.

Die Abbildungen zeigen deutlich, dass *MRCBR RF* und *Mean* die besten Methoden sind. *MRCBR RF* hat für kleine Rate von fehlenden Daten einen klaren Vorsprung gegenüber *Mean*. Für den mittleren gewichteten MAF verlaufen beide ab 60% gemeinsam und für die gewichtete STD liegt ab 40% *Mean* leicht über *MRCBR RF*. Beide verschlechtern sich für eine steigende Rate von fehlenden Daten. Alle Methoden, auch *RF* und *N/A Sim 0.5* haben die Tendenz am Ende der 90% Rate in *Var Delete* überzulaufen, was auch in den vorherigen Abbildungen erkennbar ist. Nur *N/A Sim 0* bricht aus diesem Verhalten bei beiden Fehlern aus und verschlechtert sich rapide. Für die mittlere gewichtete STD ist auch *RF* deutlich schlechter für eine höhere Rate als der Rest der Methoden.

Abschließend lässt sich sagen, dass die Leistung der verschiedenen Methoden sich für eine steigende Rate von fehlenden Daten meist immer weiter angleicht. Doch selbst bei sehr hohen Raten von fehlenden Daten liefern *MRCBR RF* und *Mean* die verlässlichste Lösung.



(a) Für den mittleren gewichteten MAF. **(b)** Für die mittlere gewichtete STD.

Abbildung 3.5: Korrektheit der Rangfolge der ausgewählten Methoden in der MNAR Umgebung für alle Versuche in Bezug auf den mittleren gewichteten Fehler des MAF (links) und der mittleren gewichteten STD bei steigender Rate von fehlenden Daten.

3.3.5 Zusammenfassung

Die verschiedenen Analysen der Versuche in der MNAR Umgebung weisen auf ein analoges Verhalten der Methoden zu den Ergebnissen des Experiments aus der MCAR Umgebung hin.

MRCBR RF gehört zu den besten Methoden. Für eine steigende Anzahl von Variablen mit fehlenden Daten kann es seine Stärken am besten ausspielen und vergrößert seinen Abstand zu den anderen Methoden in Bezug auf die Fehlermaße. Auch *MRCBR CART* zeigt gute Ergebnisse, allerdings nicht mehr ganz so dicht an denen von *MRCBR RF*, wie zuvor in der MCAR Umgebung. Als eine stabile Methode präsentiert sich wieder *Mean*, die besonders bei einzelnen Variablen gut funktioniert. Für kategoriale Variablen haben alle Methoden wieder bis auf *N/A Sim 0* Probleme. Allerdings ist *N/A Sim 0* für alle anderen Versuche nicht geeignet gewesen. Der Rest der Methoden fällt für die MNAR Umgebung weiter zurück und überspringt teilweise früh die Schranke von *Var Delete*

Zusammenfassend lässt sich feststellen, dass wie erwartet die MNAR Umgebung die Leistung der Methoden mindert und ihnen aufgrund des Informationsverlustes erschwert gute Resultate aufgrund einer Vorhersage zu erzeugen.

3.4 Unterschiedliche Größen der Datenbank und *Top N*

In den letzten beiden Abschnitten wurde das Verhalten des MRCBR und der konkurrierenden Methoden in der MCAR und MNAR Umgebung betrachtet. Für die Versuche wurden die unvollständigen Datenbanken mit Hilfe der unterschiedlichen Methoden für das Case-based Reasoning zur Verarbeitung nutzbar gemacht und jeweils das Retrieval mit der Ausgabe der ähnlichsten Fälle durchgeführt. Anschließend wurden diese Ergebnisse der Methoden mit denen des CBR Retrieval auf der vollständigen wahren Datenbank für die 20 ähnlichsten Fälle verglichen. Unter realen Bedingungen in der Praxis gibt es jedoch drei wichtige Gesichtspunkte, deren Auswirkungen auf die Leistung der Methoden in diesem Abschnitt untersucht werden sollen.

Zum einen variiert die Größe der Datenbank je nach Anwendungsbereich von sehr kleinen Datenbanken mit nur ein paar dutzend Fällen bis zu Datenbanken mit mehreren tausend Fällen und mehr. MOSAIQ ist eine dieser Beispiele und sie wächst jedes Jahr um Tausende neuer Fälle. Mit fortschreitender Sammlung und Archivierung von Daten, besonders im Medizinischen Bereich, ist in der Zukunft nach oben keine Grenze mehr gegeben. Aber es werden auch immer kleinere Teilmengen mit spezifischen Fragestellungen interessant bleiben. Daher wird für die Untersuchung des Einflusses der Größe die bisherige genutzte wahre Datenbank mit 474 Fällen um 50% auf 237 Fälle und um 25% auf 118 Fälle verkleinert. Einzelheiten zu dem Verfahren der Verkleinerung finden sich in Abschnitt 2.2.1.

Zum anderen betrachten die spezifischen CBR Anwendungen in ihren Anwendungsbereichen eine unterschiedliche Anzahl von besten Fällen nach dem Retrieval für die weitere Verarbeitung in den folgenden Phasen. Manche beschränken sich nur auf drei Fälle, um eine sichere Aussage mit den wirklich ähnlichsten Fällen zu treffen, andere

dagegen brauchen eine Vielzahl von Fällen, um eine sinnvolle Prognose daraus ableiten zu können. In den letzten Abschnitten wurden die 20 besten Fälle, *Top* 20, für die Bewertung der Methoden genutzt. Für eine differenzierte Betrachtung werden nun die *Top* 20 in Vergleich zu den *Top* 5 gesetzt. So kann ein Blick auf die beiden Enden der im Allgemeinen genutzten Anzahl der *Top N* geworfen werden. Die genaue Erläuterung der *Top N* wird in Abschnitt 2.2.5 gegeben.

Die Versuche werden nicht wie zuvor auf unterschiedlichen Arten von Variablen und ihren Kombinationen durchgeführt, sondern auf dem künstlich wahren Fall Szenario (kwFS). Das kwFS beinhaltet eine höhere Anzahl von Variablen mit fehlenden Daten, welche aus verschiedenen Arten von Variablen bestehen und unterschiedliche Typen von fehlenden Daten aufweisen. Es bietet aufgrund dessen eine realistische praxisnahe Umgebung, da es unter anderem auch versucht die natürliche Verteilung der unterschiedlichen Arten von unvollständigen Variablen in echten Datenbanken wiederzugeben. Details zum kwFS und seiner Zusammenstellung sind in Abschnitt 2.2.3 gegeben. Im nächsten Abschnitt 3.5 wird das kwFS in Kontrast zur MCAR und MNAR Umgebung gesetzt und seine beschriebenen Eigenschaften nachgewiesen.

Der restliche Aufbau der Versuche des Experiments und die dazugehörigen Parameter sind analog zu den Experimenten der MCAR Umgebung 3.2, beziehungsweise MNAR Umgebung 3.3.

3.4.1 Verhalten ausgewählter Methoden für eine steigende Rate

Die Veränderung des mittleren absoluten Fehler (MAF) und der Standardabweichung (STD) bei steigender Rate von fehlenden Daten in der kwFS Umgebung wird für einige ausgewählte Methoden, dieselben wie schon in den letzten Abschnitten, in Abbildung 3.6 für MAF und in Abbildung 3.7 für STD gezeigt.

Die Rate der fehlenden Daten ist im kwFS für jede unvollständige Variable dieselbe, so dass einige unvollständige Fälle dadurch eine sehr hohe Anzahl von fehlenden Daten aufweisen können. Die Gegenüberstellung in den Abbildungen bietet einen anschau-

lichen Vergleich des Verhaltens der Methoden in unterschiedlich großen Datenbanken (474, 237, 118) für unterschiedliche *Top N* (20, 5).

3.4.2 Verhalten aller Methoden für die durchschnittliche Rate

Die Tabelle 3.5 beinhalten die Ergebnisse des mittleren Fehlers von MAF und STD für alle untersuchten Methoden in der kwFS Umgebung.

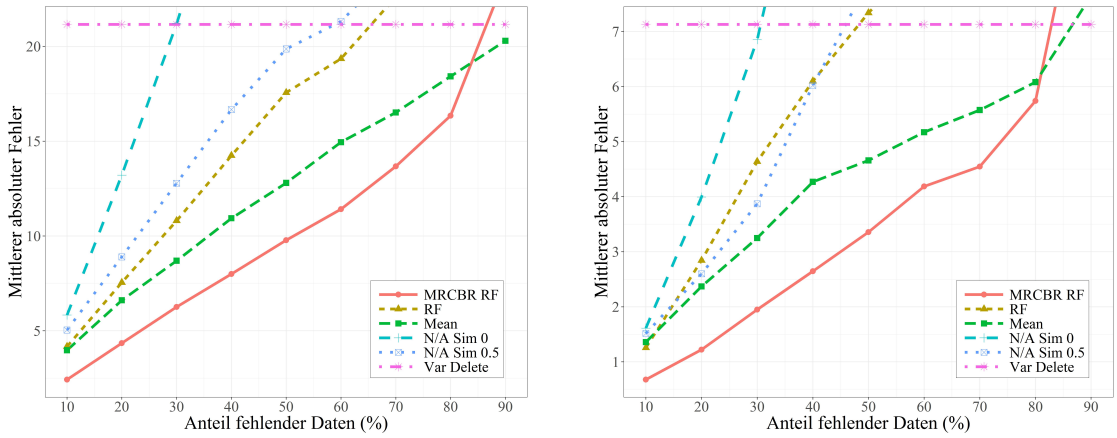
Eine Erklärung des mittleren Fehlers im Detail findet sich in Abschnitt 2.2.5. Kurzgefasst ist er der Durchschnitt über die Raten von fehlenden Daten für jeden einzelnen Versuch. Für eine einfache Übersicht unterteilen sich die Ergebnisse der Tabellen zum einen in die Größe der Datenbank und zum anderen in die zwei *Top N*.

Tabelle 3.5: Ergebnisse aller Methoden für das künstlich wahre Fall Szenario in Bezug auf den mittleren Fehler des MAF und der STD (in Klammern) über die steigende Rate von fehlenden Daten für unterschiedliche Größen der Datenbank (474, 237, 118) und *Top N* (20,5)

	Datenbank 100%		Datenbank 50%		Datenbank 25%	
Methode	*Top* 20	*Top* 5	*Top* 20	*Top* 5	*Top* 20	*Top* 5
Var Delete	21.2 (25.6)	7.1 (10.0)	14.3 (16.2)	6.6 (8.1)	10.4 (11.2)	4.3 (5.5)
Pair Delete	15.0 (20.3)	6.0 (8.6)	11.0 (13.7)	4.6 (5.8)	7.9 (8.9)	3.7 (4.6)
N/A Sim 0	30.4 (31.4)	13.3 (14.6)	21.7 (21.3)	10.4 (10.8)	14.7 (13.4)	7.6 (7.1)
N/A Sim 0.5	15.6 (19.1)	5.9 (7.7)	11.4 (13.0)	4.7 (5.5)	8.1 (8.6)	3.8 (4.2)
Mean	**10.6** (**14.0**)	**3.8** (**5.1**)	**8.5** (**10.2**)	**3.2** (**4.0**)	**6.2** (**7.1**)	**2.6** (**3.0**)
CART	12.1 (18.9)	4.7 (7.1)	8.9 (12.4)	3.4 (4.5)	6.2 (7.4)	2.6 (3.1)
RF	13.8 (21.2)	5.8 (8.8)	9.7 (14.0)	4.2 (5.9)	6.7 (8.7)	2.9 (3.9)
MRCBR CART	**8.2** (**11.0**)	**2.7** (**3.7**)	**6.7** (**8.0**)	**2.5** (**3.1**)	**5.3** (**5.9**)	**2.0** (**2.4**)
MRCBR RF	**8.0** (**10.4**)	**2.7** (**3.5**)	**6.7** (**7.8**)	**2.5** (**3.1**)	**5.2** (**5.8**)	**2.0** (**2.4**)

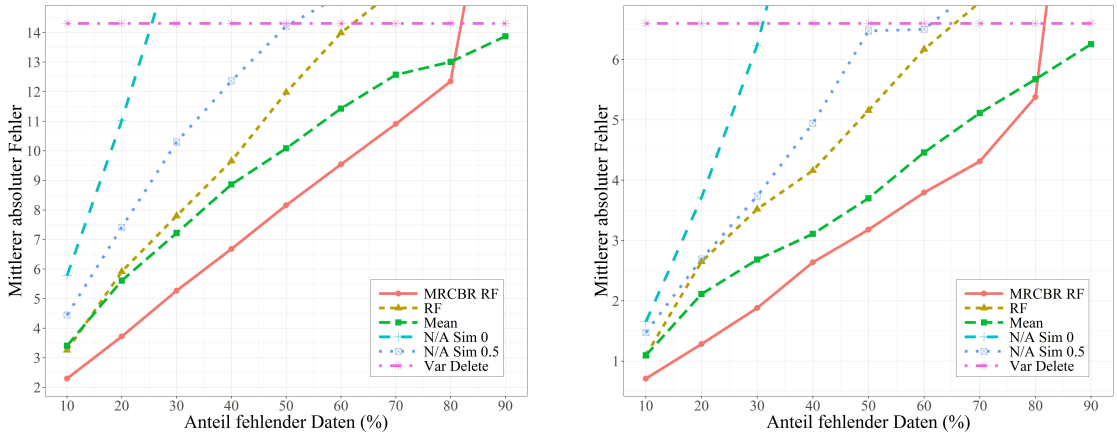
3.4.3 Analyse der Ergebnisse

Die Ergebnisse der Abbildungen 3.6 und 3.7 als auch der Tabelle 3.5 spiegeln die Resultate der letzten Abschnitte wieder, was die Leistung der Methoden angeht. Besonders wenn man an die Versuche mit einer höheren Anzahl von unvollständigen Variablen zurückdenkt. Dies gilt nicht nur für die Ergebnisse auf der Datenbank in der Originalgröße und *Top* 20. Die Methoden zeigen in allen Versuchen mit der Größe der Datenbank



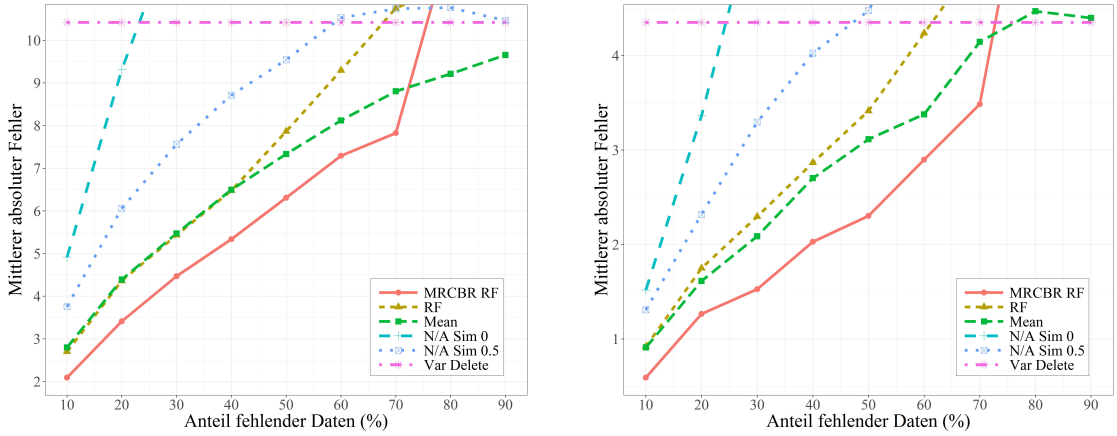
(a) Fälle = 474/ *Top* 20

(b) Fälle = 474/ *Top* 5



(c) Fälle = 237/ *Top* 20

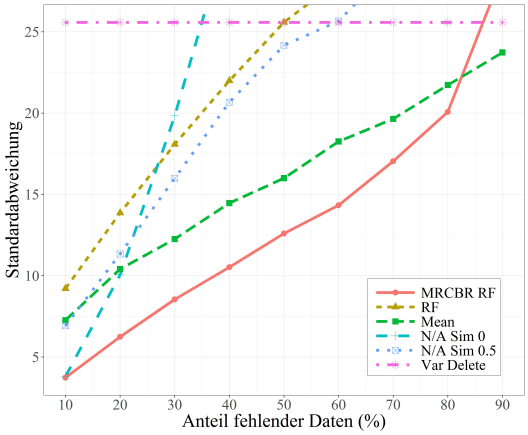
(d) Fälle = 237/ *Top* 5



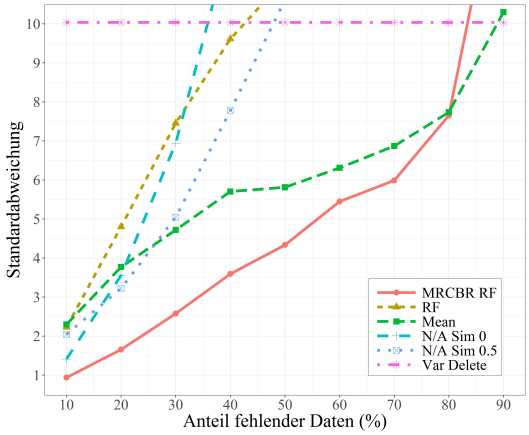
(e) Fälle = 118/ *Top* 20

(f) Fälle = 118/ *Top* 5

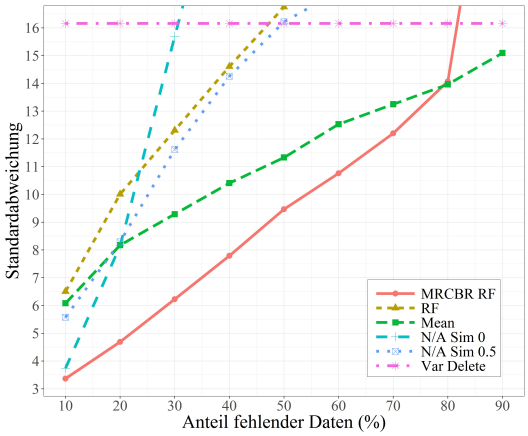
Abbildung 3.6: Korrektheit der Rangfolge ausgewählter Methoden für das künstlich wahre Fall Szenario in Bezug auf den MAF bei steigender Rate von fehlenden Daten. Die Ergebnisse für die *Top* 20 sind links, für die *Top* 5 rechts aufgeführt. Die Größe der Datenbank (474, 237, 118) sinkt von oben nach unten.



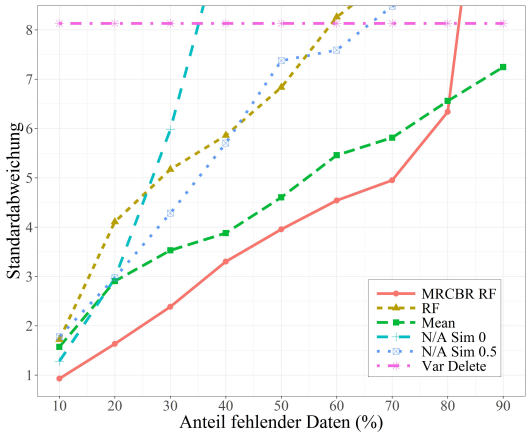
(a) Fälle = 474/ *Top* 20



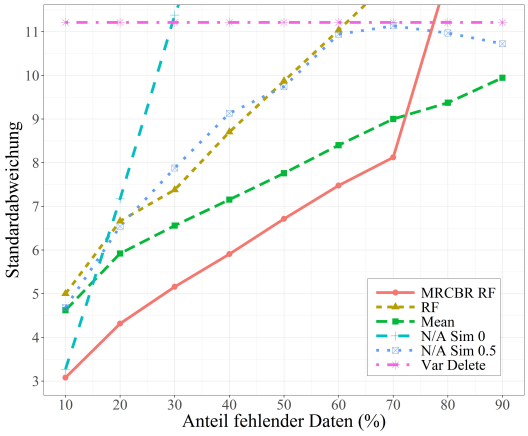
(b) Fälle = 474/ *Top* 5



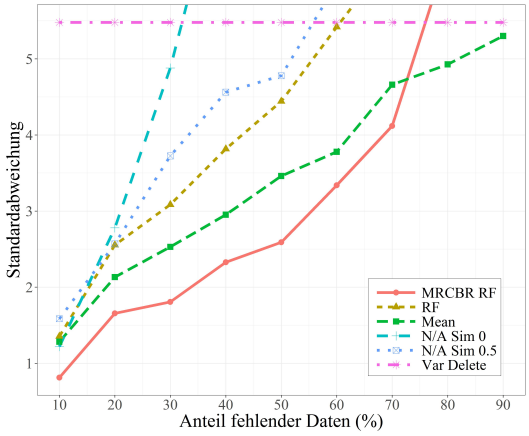
(c) Fälle = 237/ *Top* 20



(d) Fälle = 237/ *Top* 5



(e) Fälle = 118/ *Top* 20



(f) Fälle = 118/ *Top* 5

Abbildung 3.7: Korrektheit der Rangfolge ausgewählter Methoden für das künstlich wahre Fall Szenario in Bezug auf die STD bei steigender Rate von fehlenden Daten. Die Ergebnisse für die *Top* 20 sind links, für die *Top* 5 rechts aufgeführt. Die Größe der Datenbank (474, 237, 118) sinkt von oben nach unten.

und der Anzahl der *Top N* dieselbe Rangfolge betreffs ihres Verhaltens in Bezug auf MAF und STD. Damit ist offensichtlich, dass Größe der Datenbank und die Anzahl der *Top N* keinen wesentlichen Einfluss auf die Leistung der Methoden hat.

Aus der Tabelle 3.5 geht klar hervor, dass *MRCBR RF* und *MRCBR CART* gefolgt von *Mean* die besten Methoden sind. Mit gewissem Abstand folgen *CART* und *RF*. Der Rest der Methoden fällt weit zurück. Wie in allen Abbildungen von Abbildung 3.6, beziehungsweise von Abbildung 3.7, ersichtlich ist, führt *MRCBR RF* mit großen Abstand bis zu einer Rate von mindestens 70%. Dann bricht es ein und wird von *Mean* ab 80% abgelöst.

Der Vergleich der Fehler für die unterschiedlichen Datenbankgrößen und die zwei *Top N* offenbart allerdings, dass sich zwar die Methoden in ihrer Leistung in den Versuchen gleich verhalten, aber die Größe der Werte der beiden Fehler sich jeweils beträchtlich in den entsprechenden Versuchen unterscheidet. Somit unterscheidet sich auch die relative Distanz der Fehlergrößen der Methoden zueinander. Die Auswirkung der Größe der Datenbank und die Anzahl der *Top N* auf die Größe der Fehlerwerte von MAF und STD wird im Weiteren genauer analysiert.

Verhalten des mittleren absoluten Fehlers

Bei sinkender Größe der Datenbank wird MAF kleiner, sowohl für die *Top* 5 als auch *Top* 20. Dies kann gut für den mittleren MAF in Tabelle 3.5 gesehen werden. Allerdings fällt MAF mit sinkender Größe der Datenbank für die *Top* 20 schneller als für die *Top* 5. In einer kleineren Datenbank sind für hohe *Top N* auch mehr ähnliche Fälle in den *Top N* enthalten, da sie einen viel größeren Anteil der Gesamtdatenbank stellen. Wogegen es sich auf kleine *Top N* nicht so sehr auswirkt, da sich das Verhältnis Datenbankgröße zu *Top N* nicht so rasch ändert. Je größer die Datenbank wird, umso weniger tritt dadurch dieser Effekt in Erscheinung.

Zur Erinnerung, MAF bedeutet für eine feste Anzahl von ähnlichsten Fälle, um wie viele Plätze jeweils ein Fall aus der unvollständigen Datenbank, welche durch eine Methode für CBR nutzbar gemacht wurde, in der Rangfolge nach dem Retrieval von

seinem wahren Platz in der vollständigen Datenbank im Durchschnitt abweicht. Eine Veränderung der Größe der Datenbank bedeutet darum, dass die Wahrscheinlichkeit für die N ähnlichsten Fälle an ihrer wahren Position zu liegen, in kleineren Datenbanken unweit größer ist, als in großen Datenbanken. *MRCBR RF* hat zum Beispiel in Tabelle 3.5 für die *Top* 20 in der vollen Datenbank einen mittleren MAF von 8.0, in der 25% Datenbank einen mittleren MAF von 5.2.

Dieser Effekt ist so drastisch, dass für *Top* 5 in der vollen Datenbank nur die beiden *MRCBR* Methoden, *Mean* und *CART* unter einem mittleren MAF von 5 liegen, so dass sie in den wahren *Top* 5 komplett enthalten sein können. Mit sinkender Größer der Datenbank schaffen diese Hürde immer mehr Methoden, so dass für die 50% Datenbank nur *Var Delete* und *N/A Sim 0* dies nicht erfüllen und für die 25% Datenbank nur noch *N/A Sim 0*. Abbildung 3.6 zeigt dies anschaulich.

Mit diesem Effekt bei der Verkleinerung der Größe der Datenbank geht auch einher, dass der relative Abstand der Methoden zueinander in Bezug auf den mittleren MAF mit sinkender Größe der Datenbank auch kleiner wird. So ist *Mean* in unserem eben genannten Beispiel für die *Top* 20 in der vollen Datenbank 33% schlechter als *MRCBR RF*, in der 25% Datenbank nur noch 18%.

Für die *Top* 5 treten diese beiden zusammenhängenden Effekte in abgeschwächter Form in Erscheinung. Die Veränderung der Fehlerwerte mit sinkender Datenbankgröße ist nicht so drastisch, wie in den *Top* 20. *MRCBR RF* für die *Top* 5 hat in der vollen Datenbank einen mittleren MAF von 2.7, in der 25% Datenbank einen mittleren MAF von 2.0. Der relative Abstand der Methoden zueinander verkleinert sich somit auch nicht so stark wie in den *Top* 20. *Mean* ist für die *Top* 5 in der vollen Datenbank 43% schlechter als *MRCBR RF*, in der 25% Datenbank nur noch 27%. Die beschriebenen Effekte verhalten sich für alle anderen Methoden ähnlich.

In den beiden beschriebenen Beispielen für *Top* 5 und *Top* 20 kann jedoch ein Unterschied in ihren Auswirkungen gesehen werden. Die Methoden haben für *Top* 5 einen größeren Abstand zu *MRCBR RF* in Bezug auf den mittleren MAF als für *Top* 20. *Mean* ist in der vollen Datenbank 33% schlechter als *MRCBR RF* für *Top* 20, dagegen

43% für *Top* 5. Dies gilt in allen drei untersuchten Größen der Datenbank. Doch auch wenn sich der Abstand zu *MRCBR RF* bei kleineren *Top N* vergrößert, wird der MAF aller Methoden wie bereits erwähnt mit sinkender Größe der Datenbank allgemein kleiner. Interessanterweise ist *MRCBR CART* in allen Versuchen genauso gut wie *MRCBR RF*, außer für *Top* 20 in der vollen Datenbank. Für *Top* 5 nehmen beide sogar immer dieselben Werte ein.

Verhalten der Standardabweichung

Die Ergebnisse für STD korrelieren mit für MAF. Die für MAF beschriebenen Effekte in Bezug auf die Größe der Datenbank und *Top N* bleiben bestehen. Es kann in Abbildung 3.7 und Abbildung 3.6 beobachtet werden, dass sich die beschriebenen Effekte allerdings stärker auswirken.

Wie schon in den letzten Abschnitten für die MCAR und MNAR Umgebung ist STD der Methoden höher als der zugehörige MAF. Da im kwFS sieben Variablen unvollständig ist, sind auch Methoden davon betroffen, die für eine einzelne unvollständige Variable eine relative stabile STD aufwiesen, wie es *Mean* aber auch *N/A Sim 0* der Fall war. Vor allem *MRCBR RF* zeigt für die mittlere STD in allen Versuchen aus Tabelle 3.5 gute Werte, so dass die Abstände zu den anderen Methoden noch weit größer sind, als für den mittleren MAF. *Mean* ist zusätzlich im Schnitt noch ca.2-3% schlechter, als zuvor bei MAF. Auch *MRCBR CART* ist 6% schlechter und schließt sich erst für sinkende Datenbankgrößen in *Top* 5 auf.

Wie bereits in der einleitenden Analyse erwähnt, ist die Rangfolge der Methoden auch bezüglich ihrer Fehlerwerte in STD gleich.

3.4.4 Zusammenfassung

Die Analyse der Versuche zeigte, dass die Größe der Datenbank und die Anzahl der zu verwendenden besten N ähnlichsten Fälle keinen Einfluss auf die Rangfolge der Methoden betreffs ihrer Leistung in Fehlern MAF und STD haben. Es konnte jedoch beobachtet werden, dass die Fehlermaße mit sinkender Größe der Datenbank sowohl

für *Top* 20 als auch *Top* 5 kleiner werden, sowohl für MAF als auch STD. Einhergehend mit diesem Effekt verringert sich auch der relative Abstand der Methoden zueinander bezüglich ihrer Fehlerwerte. Die beiden *MRCBR* Methoden wiesen für *Top* 5 einen größeren Vorsprung zu den anderen Methoden auf, als für *Top* 20. Aber in allen Versuchen waren sie deutlich besser als die restlichen Methoden.

In der Essenz bedeutet dies, dass je kleiner die Datenbank und je kleiner die *Top N*, desto höher die Wahrscheinlichkeit, dass eine Methode die ähnlichsten Fälle richtig wählt. Andererseits bedeutet dies auch, dass *MRCBR* für große Datenbanken im Vergleich zu den konkurrierenden Methoden immer besser wird.

3.5 Umgebungen im direkten Vergleich

Nachdem in den letzten drei Abschnitten die Auswirkungen verschiedener Umgebungen für die unvollständigen Daten untersucht wurden, werden nun diese Umgebungen in Vergleich zueinander gesetzt und ihre Unterschiede und Ähnlichkeiten ergründet. Diese Umgebungen waren die MCAR Umgebung in Abschnitt 3.2, die MNAR Umgebung in Abschnitt 3.3 und das komplexere künstlich wahre Fall Szenario kwFS mit einer gemischten Umgebung in mehreren Variablen. Der Vergleich der Umgebungen soll zeigen, ob das Nichtwissen um die Herkunft und Typ der fehlenden Daten, wie es in der Praxis zumeist der Fall ist, ein Risiko in Bezug auf die Wahl der richtigen Methode darstellt oder ob es einige Methoden gibt, die allgemein verlässlich sind.

Im letzten Abschnitt konnte gezeigt werden, dass die Größe der Datenbank und die Anzahl der zu verwendenden besten N ähnlichsten Fälle keine Auswirkungen auf die Leistung der Methoden bezüglich ihrer Rangfolge untereinander haben. Daher wird sich das Experiment dieses Abschnitts wieder auf die volle Datenbank und *Top* 20 beziehen.

3.5.1 Verhalten aller Methoden in den Umgebungen

Tabelle 3.6 präsentiert die zusammengefassten Ergebnisse für alle Methoden in der MCAR, MNAR und kwFS Umgebung beruhend auf den Ergebnissen des mittleren absoluten Fehlers (MAF) und der Standardabweichung (STD). Der verwendete Fehler ist der mittlere gewichtete Fehler des MAF und der STD für alle Raten über alle Versuche.

Dieser wurde bereits in den Abschnitten der Vergleichsübersicht in der MCAR Umgebung 3.2.4 und MNAR Umgebung 3.3.4 dort in den vergleichenden Abbildungen für jeweils eine einzelne Rate verwendet. Kurz gesprochen verdichtet der mittlere gewichtete Fehler die Ergebnisse aller Raten in allen Versuche des Experiments der jeweiligen Umgebungen in einem Ergebnis und erlaubt eine relative Betrachtung. So ist es möglich die Ergebnisse der unterschiedlichen Methoden in den drei Umgebungen direkt zu vergleichen. Seine Erläuterung und Vorteile sind in Abschnitt 2.2.5 aufgeführt.

Tabelle 3.6: Ergebnisse aller Methoden in Bezug auf den mittleren gewichteten Fehler des MAF und der STD (in Klammern) über die steigende Rate von fehlenden Daten in allen Versuchen für die MCAR, MNAR und kwFS Umgebung.

Methode	MCAR	MNAR	kwFS
Var Delete	0.132 (0.122)	0.125 (0.124)	0.157 (0.149)
Pair Delete	0.156 (0.148)	0.137 (0.136)	0.111 (0.118)
N/A Sim 0	0.217 (0.153)	0.218 (0.168)	0.225 (0.183)
N/A Sim 0.5	0.135 (0.123)	0.128 (0.124)	0.115 (0.111)
Mean	**0.075** (**0.081**)	**0.073** (**0.074**)	**0.079** (**0.082**)
CART	0.082 (0.109)	0.084 (0.098)	0.090 (0.110)
RF	0.081 (0.118)	0.095 (0.122)	0.102 (0.123)
MRCBR CART	**0.063** (**0.076**)	**0.072** (**0.080**)	**0.061** (**0.064**)
MRCBR RF	**0.060** (**0.069**)	**0.069** (**0.073**)	**0.059** (**0.061**)

3.5.2 Analyse der Ergebnisse

Die Auswertung von Tabelle 3.6 zeigt, dass sich die Tendenzen des Verhalten der Methoden auch in der Gesamtzusammenfassung fortsetzen, welche schon in den einzelnen Abschnitten für die Umgebungen sichtbar geworden sind. Die Rangfolge der Methoden

ist in Bezug auf den mittleren gewichteten Fehler in allen Umgebungen dieselbe, bis auf kleine Abweichungen bei den schlechteren Methoden. Dies gilt sowohl für MAF als auch STD.

MRCBR RF ist für alle Umgebungen die beste Methode, dicht gefolgt von *MRCBR CART*. Der *Mean* bleibt für den mittleren gewichteten MAF bei MCAR, MNAR und kwFS auf dem dritten Platz. Allerdings in der MNAR Umgebung ist er für die mittlere gewichtete STD besser als *MRCBR CART* und auf Platz zwei. Die nachfolgenden Methoden mit leichtem Abstand zu *Mean* sind *RF* und *CART*, wobei letzterer im Gegensatz zu den multiplen Gegenstücken bei den *MRCBR* Methoden immer besser ist als *RF*. Alle anderen Methoden liegen weit zurück und sind mindestens 85% schlechter in allen Umgebungen als *MRCBR RF*.

Der Vergleich der Umgebungen miteinander zeigt, dass sie sich zwar in der Rangfolge der Methoden gleich verhalten, jedoch in der MNAR Umgebung die Distanz der *MRCBR* Methoden zu den anderen Methoden schrumpft. Auch der *RF* verschlechtert sich in dieser Umgebung. Dies spiegelt die Ergebnisse des Experiments der MNAR Umgebung wieder, dass sich die Klassifikations-Algorithmen schwer mit dem unwiederbringlichen Informationsverlust des MNAR tun. Für die mittlere gewichtete STD ist dieser Effekt noch stärker ausgeprägt, als für den mittleren gewichteten MAF.

Die Abstände der Methoden zu *MRCBR RF* sind für die mittlere gewichtete STD in der MCAR und MNAR Umgebung geringer ausgeprägt. Für die kwFS Umgebung wiederholt sich das Verhalten, welches in den Experimenten für MCAR und MNAR beobachtet wurde, dass die *MRCBR* Methoden für eine höhere Anzahl von unvollständigen Variablen ihren Vorsprung zu den anderen Methoden ausbauen. Bis auf beide *MRCBR* Methoden und *Mean* verschlechterten sich alle weiteren Methoden deutlich in einer solchen Umgebung aus gemischten Typen von fehlenden Daten.

3.5.3 Zusammenfassung

In den Gesamtergebnissen der untersuchten Umgebungen der letzten Abschnitte schneiden die beiden *MRCBR* Methoden mit Abstand, sowohl für den MAF, als auch die

STD, am besten ab. *Mean* liegt hinter den beiden Methoden und war nur bei MNAR in der STD besser als *MRCBR CART*, jedoch nie als *MRCBR RF*. Die singuläre Klassifikations-Algorithmen *RF* und *CART* kommen dicht danach. Alle anderen Methoden weisen keine Leistung auf, die für solide Resultate in einer Anwendung brauchbar wäre.

Die Auswertung der Experimente der MCAR, MNAR und kwFS zeigt, dass das Wissen um die Typen der fehlenden Daten nicht zwingend erforderlich ist, wie dies bei realen Anwendungen in der Praxis so gut wie immer der Fall ist. Dabei spielt der Wahl der richtigen Methode eine entscheidende Rolle um verlässliche Ergebnis zu erhalten.

3.6 Einordnung der MAR Umgebung

Zum Abschluss der verschiedenen Experimente findet eine Einordnung des fehlenden Daten Typs MAR innerhalb der beiden anderen Typen MCAR und MNAR statt. Die Definition und Wirkungsweise der drei Typen von fehlenden Daten ist in Abschnitt 1.3.1 ausführlich erläutert.

Da die Unvollständigkeit einer Variable in der MAR Umgebung von einer anderen Variable abhängt, ist eine akkurate experimentelle Analyse dieser Umgebung schwierig und aufwändig. Zum einen muss die Korrelation der Variablen untereinander in Betracht gezogen werden, da diese einen direkten Einfluss auf das Verhalten von MAR hat. Bei numerischen Variablen kann die Korrelation unter anderem, wie in Abschnitt 2.2.1 beschrieben, mit dem Korrelationskoeffizienten nach Pearson bestimmt werden. Bei kategorialen Variablen ist eine Bestimmung der Korrelation untereinander schon weit komplexer. Doch die Interpretation des Verhältnisses zwischen numerischen und kategorialen Variablen bleibt weiterhin schwierig. Ein einheitlicher vergleichender Koeffizient dieser drei Möglichkeiten ist nicht verfügbar. Zum anderen bietet eine Datenbank eine Vielzahl von Variablen von der eine Variable in der MAR Umgebung abhängen kann. In der in dieser Arbeit genutzten Datenbank (2.2.1) wären dies 16 Variablen für die mögliche Abhängigkeit. In jeder zweier Kombination entstehen jeweils vier neue

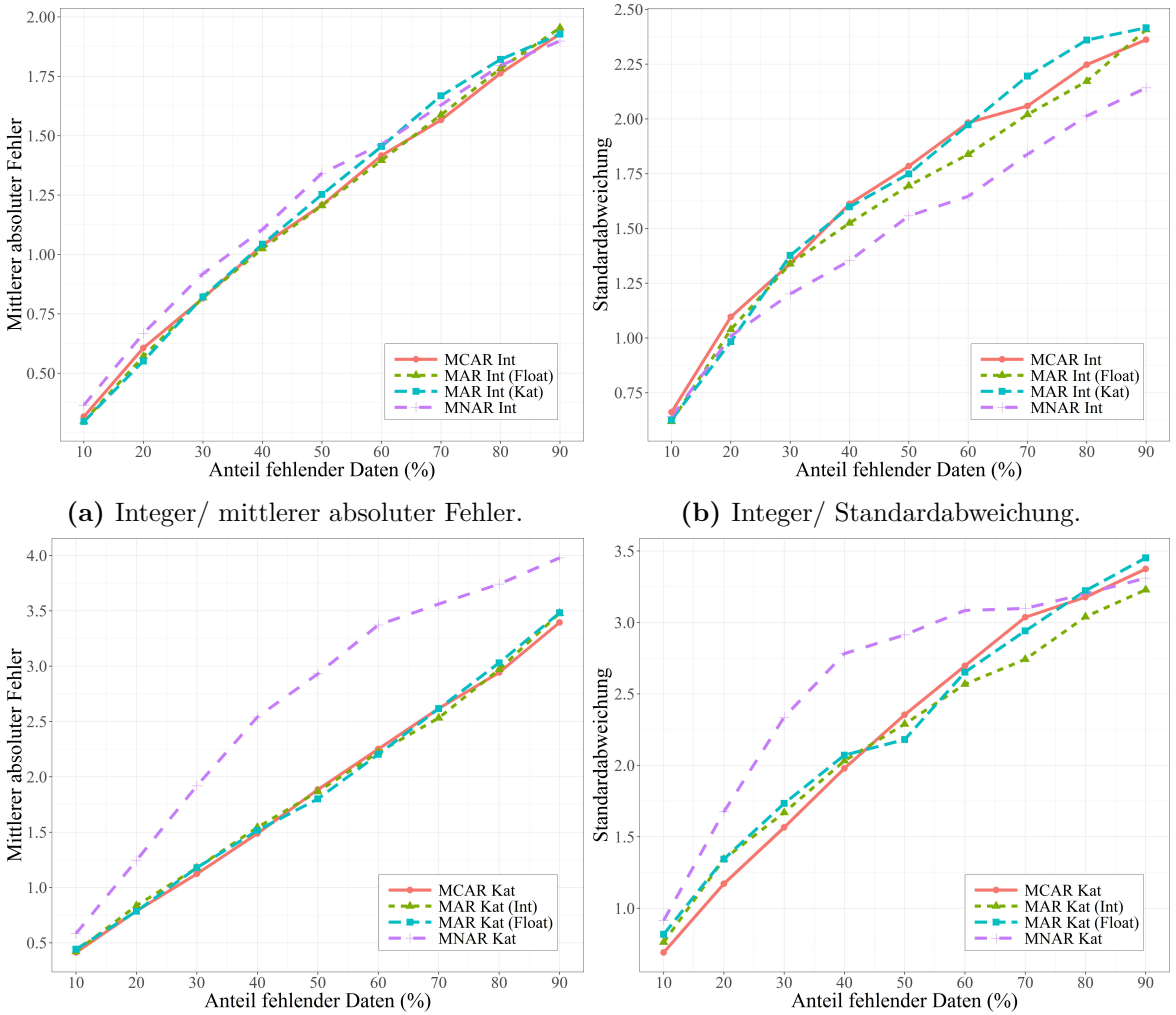
Versuche, für dreier Kombinationen 8 Versuche. Selbst wenn man die Versuche nur für Integer, Float und Kategoriale durchführen würde, wie in den bisherigen Experimenten dieses Kapitels, wären dies bereits 23 Versuche, deren Analyse schwer zugänglich wäre. Die schiere Anzahl von Versuchen würde somit den Rahmen dieser Arbeit sprengen und ist auch nicht ihr Fokus. In den bisherigen Experimenten kam die MAR Umgebung daher nur im künstlich wahren Fall Szenario mit einer festen Abhängigkeit von einer Variablen vor.

Nichtsdestotrotz soll in diesem Abschnitt ein kurzer experimenteller und theoretischer Nachweis vollzogen werden, dass die MAR Umgebung in ihrer Wirkungsweise zwischen der MCAR und MNAR Umgebung liegt, um ein stimmiges Gesamtbild des Verhaltens der Methoden und des Einflusses der unterschiedlichen Typen von fehlenden Daten zu erhalten.

3.6.1 Verhalten der drei Umgebungen

Die einzelnen Abbildungen in Abbildung 3.8 zeigen die Veränderung des mittleren absoluten Fehler (MAF) und der Standardabweichung (STD) für die *Top* 20 Fälle des *MRCBR RF* bei steigender Rate von fehlenden Daten innerhalb der Integer und Kategorialen für die MCAR, MAR und MNAR Umgebung.

Da die Methode des *MRCBR RF* sich in den Ergebnissen dieses Kapitels mit nur wenigen speziellen Ausnahmen als beste Methode erwiesen hat, wird das Verhalten dieser unter den verschiedenen Umgebungen als Referenz für alle anderen Methoden betrachtet, um eine einfache Übersicht zu ermöglichen. Die vier Kurven in den Abbildungen geben das jeweilige Verhalten des *MRCBR RF* in der jeweiligen Umgebung wieder. Die Integer hängt in der MAR Umgebung einmal von Float und einmal von der Kategorialen ab, die Kategoriale von Integer und Float.



(a) Integer/ mittlerer absoluter Fehler.

(b) Integer/ Standardabweichung.

(c) Kategoriale/ mittlerer absoluter Fehler.

(d) Kategoriale/ Standardabweichung.

Abbildung 3.8: Korrektheit der Rangfolge des *MRCBR RF* in Bezug auf den MAF (links) und die STD (rechts) bei steigender Rate von fehlenden Daten in der MCAR, MAR (mit abhängiger Variable) und MNAR Umgebung für die Integer in 3.8a und 3.8b und für die Kategoriale in 3.8c und 3.8d.

3.6.2 Analyse der Ergebnisse

Alle Abbildungen in Abbildung 3.8 zeigen, dass MAR sehr nah im Bereich der Kurve von MCAR verläuft. Dies gilt sowohl für MAF als auch STD. Bei beiden Variablen in den Abbildungen 3.8a und 3.8c überschneiden sich die Kurven von MCAR und MAR für den MAF fast vollständig. In Abhängigkeit von der Integer in Abbildung 3.8d ist

MAR für eine höhere Rate von fehlenden Daten bezüglich der STD sogar besser als MCAR.

Außer für die STD bei der Integer bildet MNAR wie zu erwarten eine obere Schranke für MCAR und MAR. Besonders bei der Kategorialen in Abbildung 3.8c ist dieses Verhalten sehr ausgeprägt, wie auch bereits die MNAR Analyse in Abschnitt 3.3 ergab. Interessanterweise ist MNAR bei der Integer für die STD besser als die beiden anderen Umgebungen, wie Abbildung 3.8b deutlich zeigt.

MAR zeigt in keiner der Versuche einen Ausschlag, der dem bisherigen Verhalten von MCAR und MNAR widersprechen würde. Es verhält sich mit kleinen Abweichungen wie eine Mischung aus beiden.

3.6.3 Theoretische Erklärung

Die Analyse der Versuche zeigte, dass die MAR Umgebung in ihrem Einfluss auf das Verhalten des *MRCBR RF* zwischen dem von MCAR und MNAR liegt. Genauer gesagt zeigte MAR fast dasselbe Verhalten wie MCAR. Diese Beobachtung soll theoretisch untermauert werden.

MCAR ist für eine Analyse-Methode der harmloseste Typ von fehlenden Daten, welcher auch einfach für die Imputations-Methoden zu verarbeiten ist. Die Unvollständigkeit entsteht durch eine zufallsbedingte Löschung der Werte innerhalb einer Variablen. So bleiben die Verteilung und die statistischen Parameter der ursprünglich vollständigen Variablen auch innerhalb der unvollständigen Variable bestehen. Des Weiteren ist die Unvollständigkeit völlig unabhängig vom Rest der Datenbank.

MNAR dagegen ist, aufgrund seines unwiederbringlichen Informationsverlustes, ein Worstcase-Szenario für die Verarbeitung der Daten mit Hilfe einer Methode. Die Unvollständigkeit einer Variablen hängt direkt von ihren Eigenschaften selbst ab und die Information über einen Wertebereich kann vollständig verloren gehen. Zum Beispiel in einer Variablen über die Augenfarbe, dass die Farbe Blau nie existierte. Diese Information kann auch von der besten Methode ohne Hintergrundinformationen nicht wiederhergestellt oder geschätzt werden.

Die Unvollständigkeit einer Variablen bei MAR hängt von einer anderen Variablen ab. Je nach Größe der Korrelation zu der anderen Variable zeigen sich für MAR zwei Richtungen auf. Entweder hat es bei einer sehr schwachen Korrelation dieselbe Zufälligkeit in der Löschung wie MCAR. Oder es hat bei einer sehr starken Korrelation eine totale Abhängigkeit, nicht nur von der Information der anderen Variablen, sondern damit auch von ihrer eigenen Information. Eine solch starke Korrelation ist allerdings sehr selten der Fall. Zwischen diesen beiden Extremen bewegt sich MAR aufgrund der Stärke der Korrelation der Variablen in der Datenbank.

3.6.4 Zusammenfassung

Die MAR Umgebung erwies sich in ihrem Verhalten sehr ähnlich zu MCAR. Wie in der Analyse und der theoretischen Erläuterung diskutiert, pendelt das Verhalten von MAR für eine Variable zwischen dem Verhalten von MCAR und MNAR je nach den statistischen Eigenschaften der Datenbank.

Die Versuche und Resultate dieses Abschnitts werfen jedoch kein neues Licht auf die Ergebnisse der Experimente dieses Kapitels. Auch wenn die Versuche dieses Abschnittes nur für das *MRCBR RF* unternommen wurden, können die Ergebnisse der MCAR Umgebung 3.2, beziehungsweise MNAR Umgebung 3.2, mit gewisser Vorsicht auf die MAR Umgebung übertragen werden bis weitere Untersuchungen in der Zukunft gemacht wurden. In Abschnitt 3.5 wurde gezeigt, dass die Wahl der richtigen Methode den entscheidendsten Einfluss auf das Gesamtergebnis hat.

4 Diskussion

In diesem Kapitel werden die gewonnenen Erkenntnisse aus Kapitel 3 besprochen und interpretiert um ein Gesamtbild der Leistung des Multiple Retrieval Case-based Reasoning (MRCBR) in Kontrast zu den konkurrierenden Methoden zu erhalten. Die unterschiedlichen Experimente des letzten Kapitels werden dabei unter neuen Gesichtspunkten betrachtet, so dass sich die Stärken und Schwächen der Methoden, welche sie im Allgemeinen mitbringen, herauskristallisieren und ein Resümee gezogen werden kann.

Die Aufteilung folgt dabei nicht starr der Struktur der Abschnitte von Kapitel 3, sondern versucht eine argumentative Kette zu formen, um neben den bekannten Ergebnissen der Analyse Schwerpunkte im Verhalten der Methoden herauszuarbeiten. Der Hauptfokus liegt dabei auf dem Verhalten von MRCBR im Kontrast zu den anderen Methoden. Im Besonderen werden die allgemeinen Erkenntnisse über die Folgen der fehlenden Daten diskutiert. Des Weiteren wird auch eine Einschätzung für die Einbettung des MRCBR in tatsächliche Anwendungen und möglicher praktischer Einsatzgebiete gegeben. Den Abschluss bildet ein Ausblick auf zukünftige Forschungsschwerpunkte, die sich aus der Methode und den Ergebnissen dieser Arbeit entwickeln können.

4.1 Verhalten und Leistung der Methoden

Die verschiedenen Versuche in den Experimenten zeigten eindeutig, dass die Art der Variable, ob numerische oder kategoriale, einen stärkeren Effekt auf das Verhalten der Methoden ausübte, als der Typ der fehlenden Daten, MCAR oder MNAR, von welchem die Variable betroffen war. Die Kombination aus beiden Tatsachen steigerte den

jeweiligen Effekt noch. Unter dem Strich war die einzelne Kombination aus Kategorialer und MNAR Umgebung am kompliziertesten für die Methoden zu verarbeiten, was sich in den Ergebnissen klar niederschlug. Auch eine Vielzahl von unvollständigen Variablen machte den meisten Methoden zu schaffen. Auf beide Tatsachen wird im späteren Verlauf nochmals eingegangen.

Auch wenn die Leistung der Methoden aufgrund des mittleren absoluten Fehler (MAF) und der Standardabweichung (STD) bestimmt wurde, zeigte sich ein ähnliches Verhalten der Methoden in beiden Fehlern, so dass im Weiteren, wenn nicht anders erwähnt, auf beide gleichermaßen eingegangen wird. Der auffälligste Unterschied im Verhalten beider Fehler war, dass die besprochenen Effekte für MAF stärker ausfallen, als für STD. Somit sind die Abstände der Methoden bezüglich ihrer Leistung für STD kleiner, aber bis auf wenige Ausnahmen bleibt die Rangfolge aufgrund der Fehlerwerte wie bei MAF. Das Verhalten unter STD zeigt, dass die Rangfolge der Methoden relativ stabil ist und es keine großen Ausreißer gibt.

Die besten drei Methoden

Die Ergebnisse der Experimente für die MCAR Umgebung in Abschnitt 3.2, für die MNAR Umgebung in Abschnitt 3.3 und den Versuchen zur Größe der Datenbank und der Anzahl der verwendeten ähnlichsten Fälle für das künstlich wahre Fall Szenario in Abschnitt 3.4, sowie die vergleichende Gegenüberstellung der Umgebungen in Abschnitt 3.5 zeigten die Überlegenheit des Verfahrens des *MRCBR* in beiden Versionen im Vergleich zu den anderen Methoden. Nahezu für alle Versuche war es die verlässlichste Methodik in Bezug auf eine möglichst korrekte Rangfolge der ähnlichsten Fälle nach dem Retrieval. Sein Verhalten erwies sich als stabil und vorhersagbar sowohl für verschiedene Typen von fehlende Daten (MCAR, MNAR, gemischte Umgebungen), unterschiedliche Arten und Gruppierungen von Variablen (Int, Float, Kat, gemischt), und Veränderungen in der Größe der Datenbank und der *Top N*.

Damit hat sich die Annahme der Idee der Methode aus Abschnitt 2.1.2 bestätigt, dass die Nutzung von mehreren imputierten vollständigen Datenbanken mit einem moder-

nen Klassifikations-Algorithmus einen positiven Effekt auf die Resultate des Retrieval eines CBR System hat und so eine möglichst korrekte Rangfolge der Fälle gewährleistet werden kann. Es soll hervorgehoben werden, dass sich *MRCBR RF* in allen Versuchen einen Tick besser präsentierte als *MRCBR CART*.

Obwohl *Mean*, die Imputation des Mittelwertes oder MODE für die fehlenden Daten, eine einfache und leicht zu implementierende Methode ist, war das Verhalten überraschend gut und relativ unabhängig von dem Typ der fehlenden Daten. In der Mehrheit der Versuche war es nach den beiden Methoden des *MRCBR* die beste Methode und zeigte nie ein unberechenbares Verhalten. Der Abstand zu *MRCBR RF* bezüglich der Fehler war manchmal recht klein, aber mit steigender Anzahl von unvollständigen Variablen vergrößerte er sich merklich. Keine andere Methode außer diesen drei konnte eine solche Stabilität und Verlässlichkeit der Ergebnisse gewährleisten.

Restliche Methoden

Unabhängig von der Art des Versuches war *List Delete*, die Löschung aller Fälle mit fehlenden Daten, mit großem Abstand die schlechteste Methode. Die Gründe dafür sind eindeutig der wachsende Verlust an Information mit steigender Anzahl von fehlenden Daten und zeigen eindrucksvoll die Abhängigkeit und Anfälligkeit eines CBR Systems gegenüber der inne liegenden Information der Datenbank. Bedauerlicher Weise ist diese Methode, wie bereits erwähnt, eine der am meist genutzten.

Auch *Var Delete*, die Löschung der kompletten Variablen mit fehlenden Daten, erwies sich als nicht empfehlenswert. Allerdings konvergierten mit steigender Rate von fehlenden Daten die besseren Methoden, wie *MRCBR RF* und *Mean*, gegen die statische Schranke von *Var Delete*. Dies ist aufgrund des steigenden Informationsverlustes bei hohen Raten nicht verwunderlich. Manche schwächere Methoden, wie *Pair Delete* oder *RF*, durchstießen diese Schranke bereits für eine mittlere Anzahl von fehlenden Daten.

Zwei Methoden zeigten sich besonders empfindlich gegenüber der Art der Variable. *Pair Delete*, die Nutzung nur der vollständigen Werte eines Falles, erbrachte akzepta-

ble Ergebnisse für numerische Variable, aber in allen Versuchen mit der Kategorialen scheiterte es. Ganz im Gegenteil dazu *N/A Sim 0*, die statische Einsetzung der lokalen Ähnlichkeit eines fehlenden Wertes mit 0. Für die Kategoriale erzielte es beste Ergebnisse, in der MNAR Umgebung lag es sogar vor *MRCBR RF*. Ansonsten aber war es selbst mit einer geringen Rate von fehlenden Daten schlechter als *Var Delete*.

Die feste Imputation *N/A Sim 0.5* einer fehlenden lokalen Ähnlichkeit mit 0.5 zeigte dagegen in keinem der Versuche gute Ergebnisse. Beide *N/A Sim* Methoden sind außer der Methodik des *MRCBR* die einzigen Methoden, welche speziell für das Problem des Case-based Reasoning auf einer unvollständigen Datenbank entwickelt wurden und allgemein ohne Einschränkungen nutzbar sind, wie in Abschnitt 1.4.1 und 2.1.2 ausgeführt wurde.

Die Methoden *CART* und *RF*, welche eine Imputation der fehlenden Werte mit den Klassifikations-Algorithmen Entscheidungsbaum und Random Forest durchführen, waren fast immer unter den fünf besten Methoden, aber weitaus schlechter als derselbe Klassifikations-Algorithmus innerhalb des *MRCBR*. Interessanterweise wechselten *CART* und *RF* sich in ihrer Leistung in den verschiedenen Versuchen ab, da die zufällig generierten Bäume des Random Forest wohl für Kategoriale nicht zu optimalen Ergebnissen führten. Wogegen *MRCBR RF* grundsätzlich besser arbeitete als *MRCBR CART*. Dies ist darauf zurückzuführen, dass der Random Forest hier die Stärke seiner vielen Entscheidungsbäume auf mehreren Datenbanken noch besser ausspielen kann, so dass die Mittelung all dieser Entscheidungen am Ende beste Ergebnisse erzielt und damit auch der Durchschnitt all dieser Ergebnisse auf den vielen Datenbanken.

4.2 Stärken des MRCBR

Die Methodik des *MRCBR* birgt einige Vorteile gegenüber den anderen Methoden, welche besonders in der Praxis von entschiedener Bedeutung und Nutzen sind. Diese Vorteile gelten für die Methodik des *MRCBR* im Allgemeinen, also für *MRCBR RF* und *MRCBR CART* gleichermaßen.

Anzahl der unvollständigen Variablen

In den Experimenten zeigte sich, dass sich mit einer steigenden Anzahl von unvollständigen Variablen jedweden Typs von fehlenden Daten der Abstand des *MRCBR* zu den anderen Methoden gemessen mit Hilfe der Fehlermaße vergrößerte. Das unvermeidliche Anwachsen seines Fehlers bei mehreren unvollständigen Variablen war sichtbar gemäßigter als bei allen anderen Methoden. Selbst die Methode *Mean* litt unter der höheren Anzahl von Variablen, wenn auch nicht so stark, wie die restlichen Methoden.

Dies wurde besonders für das künstlich wahre Fall Szenario (kwFS) aus Abschnitt 3.4 und 3.5 deutlich. Das kwFS versuchte die tatsächliche klinische Realität widerzuspiegeln, in der verschiedene Variablen aus den unterschiedlichsten Gründen unvollständig sind, ohne dass man ermitteln kann um welchen Typ von fehlenden Daten es sich handelt. In dieser Umgebung von sieben unvollständigen Variable, jede mit einem anderen Typ der fehlenden Daten belegt, von MCAR über MAR zu MNAR, zeigte sich, dass *MRCBR* gut mit dieser Problematik umgehen kann. Die hohe Anzahl von imputierten vollständigen Datenbanken aus dem Multiple Imputation Teil des MRCBR erwiesen sich auch hier wieder als Vorteil, da sich in diesem Falle nicht nur eine Variable in jeder imputierten vollständigen Datenbank unterscheidet, sondern jede der vorher unvollständigen Variablen. Dies sorgt dafür, dass viele Kombinationen aus diesen Variablen mit jeweils unterschiedlich imputierten Werten entstehen, manche gut, manche schlecht. In ihrer Gesamtheit an möglichen wahren Informationen verringern sie aber im Retrieval Pooling den Fehler, im Gegensatz zu einer einzelnen verwendeten Datenbank der anderen Methoden.

Wie in Abschnitt 1.3.1 erläutert wurde, gibt es nur für MCAR einen Hypothesentest und dieser ist ein Omnibustest. Sobald eine gemischte Umgebung mehrerer unvollständiger Variablen auftritt, bekommt man ein negatives Ergebnis und weiß nichts Neues über die Beschaffenheit der Variablen. Da dies in klinischen Datenbanken der Normalfall ist, kann man im seltensten Fall eine Aussage über den Typ der fehlenden Daten treffen. *MRCBR* präsentierte in jeder Umgebung fast immer die besten Ergebnisse,

besonders für eine wachsende Anzahl von unvollständigen Variablen in der Datenbank, und ist daher die plausibelste Wahl in diesem Falle.

Größe der Datenbank und *Top N*

Eine der Herausforderungen der Neuzeit ist das stetige Anwachsen von realen Datenbanken bezüglich ihrer Größe und Informationsdichte, bekannt unter dem Schlagwort Big Data. Dies geschieht in allen Bereichen in denen Daten gesammelt werden oder wurden. Doch im Gegensatz zu den großen IT-Firmen, die sich seit einigen Jahren darauf spezialisiert haben diese Informationen zu nutzen, beginnt im klinischen Bereich diese Revolution gerade erst. Einerseits bietet dies eine immense Chance den Wissensschatz zu bergen, Wissen möglichst schnell und effizient zugänglich zu machen und dadurch neue Lösungen abzuleiten. Mit einhergeht aber auch der Fluch der unvollständigen Daten, die in demselben Tempo innerhalb der Datenbanken anwachsen wie die Information. Wie in Abschnitt 1.1 beschrieben wurde, werden verschiedene Wege beschritten dem entgegenzuwirken. Doch vermeidbar sind die fehlenden Daten nicht.

Aus Abschnitt 3.4 ging hervor, dass die Größe der Datenbank und die Anzahl der *Top N* keinen signifikanten Einfluss auf die Rangfolge der Methoden betreffs ihrer Leistung hatten. Allerdings war zu erkennen, dass mit anwachsender Größe der Datenbank die Fehlerwerte der anderen Methoden schneller anwuchsen als die der *MRCBR* Methoden. Damit vergrößerte sich der Abstand aller anderen Methoden zu den beiden *MRCBR* Methoden. Derselbe Effekt stellte sich für eine sinkende Anzahl der *Top N* ein.

Die mit Hilfe moderner Klassifikations-Algorithmen und statistischen Verfahren aufbereiteten verschiedenen vollständigen Datenbank ermöglichen es *MRCBR* die volle Information der unvollständigen Datenbank zu nutzen und diese Information in einer einzigen Lösung der globalen Ähnlichkeiten zu verdichten. In Hinblick auf Big Data bietet daher die Methodik des *MRCBR* die beste derzeitige Antwort auf die Problematik CBR in großen unvollständigen Datenbank. Des Weiteren verwenden die meisten CBR

System nur eine geringe Anzahl von ähnlichen Fällen für die weitere Verarbeitung, was wiederum *MRCBR* einen Vorteil verschafft.

4.3 Schwächen des MRCBR

Die Schwachstelle von *MRCBR* liegt in der Verarbeitung einer einzelnen Variable, da es in diesem Fall sein Potenzial nicht ausspielen kann. Auch wenn es dieses Manko schon bei der Kombination von zwei Variablen wieder aufhebt, ist es eine Einschränkung. In der MCAR Umgebung bedeutet dies nur, dass für einzelne Variable die Abstände zu der nächstbesten Methode gering sind. In der MNAR Umgebung ist es dagegen nur die zweite Wahl. Bei einer einzelnen numerischen Variable dort ist *MRCBR RF* minimal schlechter als *Mean* und *MRCBR CART* nur noch die fünf beste Methode, wenn auch nur mit kleinem Abstand. Bei einer einzelnen kategorialen Variable verhält sich *N/A Sim 0* eindeutig besser als alle anderen Methoden und beide *MRCBR* Methoden folgen mit Abstand als nächstbeste Methoden.

Dieses Verhalten beider *MRCBR* Methoden im Falle von MNAR macht deutlich, welche Nachteile das Verfahren mit sich bringt. Eine Kategoriale besteht aus einer gewissen Anzahl an Leveln. Diese Level kommen, wie in den Histogrammen der Abbildung 2.3 aus Abschnitt 2.2.1 präsentiert, mal mehr, mal weniger häufig in einer Variable vor und damit in allen Fällen der Datenbank. Dadurch sind seltene Level anfällig für ein totales Verschwinden ihrer Information bei dem Typ der fehlenden Daten MNAR.

Der mögliche völlige Informationsverlust in der MNAR Umgebung mehrerer Level der kategorialen Variable führt dazu, dass Methoden wie *MRCBR*, welche die volle Information der Datenbank für ihre Vorhersage nutzen, keine optimale Leistung erbringen. Sowohl die eingebetteten Klassifikations-Algorithmen, als auch die mehrfach imputierten Datenbank können diesen Informationsverlust nicht kompensieren, da die Information für alle Zeit verloren gegangen ist.

Im Gegensatz dazu hat eine Methode wie *N/A Sim 0* einen Vorteil mit der Substitution von 0 für die lokale Ähnlichkeit eines fehlendes Wertes einer Kategorialen. Da

kategoriale Variable nur lokale Ähnlichkeiten zum Wert des Zielfalles von 0 oder 1 aufweisen und seltene Level möglicherweise gelöscht wurden, hat diese Methode eine gute Chance richtig zu liegen. Allerdings hängt dies auch stark von der Verteilung und Anzahl der Level einer Kategorialen ab und das richtige Einsetzen von statischen Werten verlangt auch etwas Glück. Aus diesem Grund wurde für die Versuche eine Kategoriale gewählt, in der auch seltene Level vorkommen, und diese wurden auch gleichwertig in den verwendeten Zielfällen des CBR repräsentiert.

Nichtsdestotrotz zeigt dieser Effekt, welchen Einfluss kategoriale Variable auf das Verhalten des Retrieval und damit auf das gesamte CBR haben. Mit den aufgeführten Problemen bei kategorialen Variablen, vom Verlust der Information von Leveln bis hin zu ihrer binären lokalen Ähnlichkeit, tun sich alle Methoden schwer. Da kategoriale Variable in den unterschiedlichsten Formen auftreten können, müssen weitere Untersuchungen für ihre Verarbeitung mit Algorithmen unternommen werden, um neue Lösungen für diese Problemstellung zu finden.

Es sei jedoch angemerkt, dass in der praktischen klinischen Anwendung bisher kategoriale Variable weit seltener auftreten als numerische Variable. Dies liegt an der Ursache, dass Laborwerte und andere Messungen betreffs der Patienten meist in numerischer Form gegeben sind und nur wenige Marker kategorialer Natur sind. Weiterhin ist so gut wie immer die ganze Datenbank von fehlenden Daten betroffen, die sich auf viele Variable erstrecken, wie bei MOSAIQ der Fall. Der Fokus auf eine einzelne Variable ist daher nur äußerst selten der Fall, weil man für ein optimales Ergebnis die Gesamtheit der Datenbank und die Information darin betrachtet oder betrachten sollte.

4.4 Ignorieren von fehlenden Daten

Es wird oft die Frage oder Forderung gestellt, ob es nicht am einfachsten wäre die fehlenden Daten zu ignorieren und sich allein auf den vollständigen Teil der Datenbank

zu beschränken. Die Experimente in dieser Arbeit haben eindeutig belegt, dass dies in keinem Fall eine ratsame Vorgehensweise ist.

Die Methode der Löschung der betroffenen Variable *Var Delete* erwies sich als obere Schranke, an welche alle besseren Methoden bei steigender Rate von fehlenden Daten konvergierten. Da der Fehlerwert dieser Methode für jede Rate von fehlenden Daten derselbe bleibt, ist eine komplette Löschung einer Variable daher nur sinnvoll, wenn das Aufkommen von fehlenden Werten innerhalb der Variable bei mindestens 90% liegt.

Die Methode der Löschung der betroffenen Fälle *List Delete* war in keinem der Versuche brauchbar. Was den Umstand umso schlimmer macht, dass sie gerne genutzt wird. In den Versuchen mit einer Variable war es schon bei einem Anteil von 30% von fehlenden Daten schlechter als *Var Delete*. Für mehrere unvollständige Variable war *List Delete* gar nicht mehr nutzbar, da so viele Fälle gelöscht wurden, dass keine sinnvolle Verarbeitung mehr möglich war.

Anhand eines Beispiels kann man die Auswirkungen beschreiben, welche die beiden Methoden mit sich bringen. Abbildung 3.6a zeigt die Entwicklung des künstlich wahren Fall Szenarios unter dem mittleren absoluten Fehler in der Datenbank voller Größe bei steigender Rate von fehlenden Daten. Bei einer 50% Rate erreicht *MRCBR RF* einen MAF von 10 und die Methode *Mean* hat bereits einen MAF von 13. Das bedeutet, dass der Durchschnitt der *Top* 20 Fälle für die beiden Methoden innerhalb der Schranke von 20 liegt. *Var Delete* dagegen hat einen MAF von 21 und ist damit außerhalb des Intervalls von 20. *List Delete* letztendlich erzeugt gar keine Ergebnisse mehr, weil es keine Fälle für die Verarbeitung übriglässt.

Dieses Beispiel zeigt sehr eindrucksvoll, dass sich der Einsatz einer passenden Methode für unvollständige Datenbanken immer positiver auswirkt als die simple Löschung von Daten. Da so gut wie alle realen Datenbanken fehlende Daten aufweisen und CBR mit seinen voneinander abhängenden Phasen besonders anfällig bezüglich dieser Tatsache ist, bleibt es unvermeidlich sich dem Problem zu stellen und Lösungen dafür zu finden.

4.5 MRCBR in der Praxis

Der Fokus dieser Arbeit lag darauf die möglichst ähnlichsten Fälle während des Retrieval zu finden und ihre korrekte Rangfolge zu gewährleisten. Ursprünglich wurde das MRCBR für das CBR-TDS des Tumorboards in der Onkologie entwickelt, um auf der unvollständigen Datenbank MOSAIQ zu arbeiten. Gleichzeitig ist es aber eine allgemeine Methodik, die nicht auf einen festen Anwendungsbereich beschränkt ist und für jedes CBR System passend ist. Außerdem wurde es so konzipiert, dass es jede der vorgestellten Methoden im Bedarfsfall verwenden kann. Im Folgenden soll erörtert werden für welche medizinischen Einsatzbereiche sich das MRCBR in der standardmäßig kompletten Form eignet und welche es nicht bedienen kann.

Entscheidung über Imputation

MRCBR führt standardmäßig keine permanente Imputation der fehlenden Werte durch. Die Imputation der Werte findet innerhalb der mehrfachen Datenbanken statt und wird für die Berechnung der am Ende gemittelten globalen Ähnlichkeiten benutzt. Danach bleibt wieder nur die unvollständige Datenbank zurück und die generierten imputierten Datenbanken werden verworfen. Im folgenden Durchlauf wird dies erneut ausgeführt.

Dies geschieht mit Absicht. Zum einen wäre eine Mittelung der imputierten Werte der mehrfachen Datenbanken nicht sinnvoll, um eine einzige imputierte Datenbank zu erzeugen, wie die Literatur in Abschnitt 1.3.4 eindeutig belegt. Die herausragende Leistung von MRCBR entsteht durch die Anwendung des Retrievals auf jeder dieser imputierten Datenbanken und der anschließenden Mittelung der globalen Ähnlichkeiten im Retrieval Pooling.

Zum anderen birgt eine permanente Imputation Risiken für die Datenbank. Die Werte können nicht als gesichert und wahr angenommen werden, so dass eine weitere Verarbeitung dieser Werte mit großen Risiken behaftet ist. Zu sehen war dies in den Ergebnissen der Klassifikation-Algorithmen, welche eine singuläre Imputation durchführen und immer schlechter abschnitten als MRCBR. In diesem Falle betraf es nur die

Rangfolge der ähnlichsten Fälle. Die imputierten Werte der fehlenden Daten selbst sind wesentlich verzerrter von ihrem wahren Wert. Dies ist der entscheidende Unterschied von MRCBR zu anderen Methoden und der Anwender muss die Entscheidung treffen, ob er eine Imputation der Werte für seinen Einsatzbereich benötigt oder ob ihm die korrekte Rangfolge der ähnlichsten Fälle wichtiger erscheint.

Differenzierung der Einsatzbereiche

Ein Entscheidungsunterstützungssystem, wie zum Beispiel CBR-TDS, ist dafür konzipiert, dass Fachärzte eine Gruppe von ähnlichsten Fällen zu ihren Patienten präsentiert bekommen, die interessanten und relevanten Informationen dieser Fälle, wie Verlauf der Krankheit oder Medikation, erhalten und aufgrund dessen ihre finale Entscheidung von diesem neu gewonnenen Wissen vergangener Fälle ableiten können. Für diesen Einsatzbereich ist die Gewährleistung der möglichst ähnlichsten Fälle zum Zielfall zwingend notwendig. Weiterhin gibt es Anwendungsfelder innerhalb der CBR Systeme, wie in Abschnitt 1.2 ausgeführt, in denen die Informationen einer gewissen Anzahl von ähnlichsten Fälle zur maschinellen Weiterverarbeitung in der Reuse-Phase verwendet werden, um einen optimale Lösung für die Patienten auf automatischem Wege zu finden. Dies kann zum Beispiel die Ermittlung der passenden Medikation oder den bestmöglichen Zeitpunkt für einen Kontrolltermin beinhalten. Für solche Arten von Anwendungen erfüllt MRCBR alle Erfordernisse auf die zurzeit bestmögliche Weise. Das es hierfür keine Alternative gibt wurde im letzten Abschnitt bereits ausführlich dargestellt.

In anderen klinischen Anwendungsbereichen, wie Planungsunterstützungssystemen in der Strahlentherapie, welche in Abschnitt 1.2.3 beschrieben wurden, ist das Finden eines ähnlichsten Falles mit gesicherten wahren Werten von Nöten, um daraus die weiteren Berechnungen nicht geschätzt, sondern gesichert durchzuführen. Dies ist in den genannten Planungsunterstützungssystemen für die Bestimmung des Bestrahlungswinkels unabdingbar. Für einen solchen Einsatzbereich muss abgewogen werden, ob die Methodik des MRCBR sinnvoll ist. Auch wenn es mit Nachteilen behaftet ist, kann in diesem Fall die alleinige Nutzung der vollständigen Fälle vorzuziehen sein. Die Anzahl

der korrekten ähnlichsten Fälle würde zwar sinken, aber die Informationen dieser Fälle wären gesichert und vollständig. Natürlich bedarf es dafür einer Ähnlichkeitsfunktion, die das Fehlen der Daten in korrekter Weise kompensiert. Falls ein Fehlerspielraum akzeptabel ist, kann auch eine Imputation für diese Werte vorgenommen werden. Um das Risiko einzugrenzen muss vorher in diesem Falle auf den vollständigen Daten in einem Training eine Fehlertoleranz bestimmt werden.

Generell muss für jeden Einsatzbereich im Vornherein die Zielsetzung und Erwartung klar festgelegt werden, ob die ähnlichsten Fälle trotz ihrer Unvollständigkeit möglichst akkurat wiederzugeben sind oder ob die tatsächlichen vollständigen Werte der Datenbank von Interesse sind. Selbstverständlich kann sich daraus mit Hilfe von weiteren Untersuchungen eine gemischte Form dieser beiden Richtungen ergeben, um ein optimales Ergebnis zu erlangen. Im Übrigen können die Resultate der Fälle, welche unvollständige Daten aufwiesen, gemarkert werden, so dass der Anwender selbst entscheiden kann, ob diese in seiner Analyse einen Anteil haben sollen oder nicht.

4.6 Ausblick

Die Experimente dieser Arbeit haben gezeigt, wie entscheidend das Bewusstsein für die Unvollständigkeit in den Datenbanken ist. Case-based Reasoning war in diesem Sinne nur ein Beispiel für die Auswirkungen der fehlenden Daten auf ein Verfahren. Die meisten Verfahren des Maschinellen Lernens sind äußert anfällig für diese Problematik, dies betrifft Random Forest genauso wie Neuronale Netzwerke. Dies stellt eine der großen Herausforderungen in der Zukunft dar. Die Notwendigkeit Lösungen hierfür zu entwickeln ist zwingend erforderlich und der Schlüssel für die Nutzung des Datenschatzes, der besonders im klinischen Bereich liegt.

Nachdem die Untersuchungen dieser Arbeit die Verlässlichkeit und Stärke des Multiple Retrieval Case-based Reasoning auf einer wahren bekannten Datenbank bewiesen haben, wäre der nächste Schritt dieses Verfahren auf realen Datenbanken in einer praktischen Anwendung zu erforschen. MOSAIQ wäre als Einsatzbereich hierfür bestens

geeignet, um das MRCBR mit eingebettetem CBR-TDS für das Tumorboard in der Praxis zu testen. Untersuchungen müssten zeigen, welche Stolpersteine sich dort zeigen würden und wie mit diesen umzugehen ist. Unabdingbar wäre dafür die direkte Zusammenarbeit mit den Fachärzten, um die Korrektheit der Ergebnisse des MRCBR-TDS zu verifizieren und Vertrauen in dieses Verfahren zu schenken. Erweiterungen in andere klinische Bereiche sind danach durchaus denkbar. Da MRCBR nicht nur allein für das Entscheidungsunterstützungssystem CBR-TDS funktioniert, sondern als Blackbox für jedes CBR System verwendet werden kann, sind die Einsatzbereiche nicht nur auf die Medizin beschränkt.

Es haben sich speziellen Fragestellungen nach der Analyse der Experimente eröffnet, deren weitere Untersuchung lohnenswert erscheint. Dazu zählt vor allem das Verhalten der Methoden für kategoriale Variable und welchen Einfluss ihre Anzahl und Verteilung der Level auf die Ergebnisse ausüben. Dies gilt auch für numerische Variable, die keine einfache Normalverteilung aufweisen. Weitere Forschung an diesen Extremfällen wäre von Interesse. Mit diesem Wissen ließe sich eine Verbindung zwischen dem Typ der fehlenden Daten und der Verteilung der Variablen auf das Gesamtergebnis erkennen. Einhergehend mit der Einbindung von neuen Imputations-Algorithmen innerhalb des MRCBR würden es die Ergebnisse dieser Forschungen ermöglichen ganz gezielte Fragestellungen mit passenden Algorithmen zu verarbeiten. Dies würde ein sehr viel komplexeres System als bisher erfordern, aber einhergehend vermutlich noch bessere Prognosen liefern.

5 Zusammenfassung

In dieser Arbeit wurde das Multiple Retrieval Case-Based Reasoning (MRCBR) vorgestellt und evaluiert. Es bietet eine Antwort auf die Fragestellung Case-Based Reasoning (CBR) auf einer unvollständigen Datenbank zu verwenden und zugleich eine verlässliche Lösung in der Retrieve-Phase zu erhalten, von der alle anderen Phasen abhängen.

Stand der Technik und Motivation

Den bisherigen Methoden für dieses Problem mangelte es an einer adäquaten Strategie fehlende Daten in der Retrieve-Phase von CBR zu handhaben. Auch fehlte ein statistisch fundierter Vergleich der möglichen Verfahren für diese Problemstellung. Es gab nur wenige Methoden, welche unterschiedliche Arten von Variablen verarbeiten konnten. Besonders für kategoriale Variable waren die Lösungen recht spärlich und unzureichend. Der Einfluss des Typs von fehlenden Daten wurde überhaupt nicht in Betracht gezogen. Auch andere Parameter, wie die Größe der Datenbank oder spezifische Eigenschaften des CBR, wurden außer Acht gelassen.

MRCBR schließt diese Lücken und liefert eine fundierte Lösung. Es wurde für das klinische Entscheidungsunterstützungssystem CBR-TDS in der Onkologie entworfen und auf einer medizinischen Datenbank getestet wurde. Allerdings ist es auch ein allgemeines Verfahren, welches nicht allein auf dieses Anwendungsfeld beschränkt ist. Es stellt für jedes beliebige CBR System ein Framework für den Umgang mit fehlenden Daten bereit ohne dabei das CBR System an sich zu verändern.

Ansatz und Methodik des MRCBR

Die Methodik des MRCBR nutzt die Stärken von modernen Methoden des maschinellen Lernens und der Statistik, um die fehlenden Daten innerhalb der Retrieve-Phase des CBR verarbeitbar zu machen. Genauer gesagt, enthält MRCBR einen weiteren Schritt am Anfang seines Kreislaufs, die Multiple Imputation mit einem Klassifikations-Algorithmus zur Imputation, und erweitert danach die klassische Retrieve-Phase in die Multiple Retrieve-Phase. Diese Erweiterung bezieht die Unsicherheit betreffs der Imputation der fehlenden Daten mit in den Algorithmus ein und ignoriert sie nicht.

Das Retrieval wird jeweils auf den vielfältigen imputierten vollständigen Datenbanken der Multiple Imputation durchgeführt, um die lokale Ähnlichkeit aller Fälle zum betreffenden Zielfall für jede der imputierten Datenbanken zu bestimmen. Das Verfahren beinhaltet so die verschiedenen denkbaren Resultate des Retrievals aufgrund der fehlenden Information in der unvollständigen Datenbank, welche auf der möglichen Verteilung der fehlenden Daten gründen. Die einzelnen Ergebnisse der Ähnlichkeiten zum Zielfall jeder imputierten Datenbank werden im Retrieval Pooling zum einem Ergebnis verschmolzen. Auf diese Weise wird eine korrekte Rangfolge der ähnlichsten Fälle zum Zielfall mit der größtmöglichen Verlässlichkeit gewährleistet.

Aufbau der Evaluation

In einer speziell für diese Fragestellung entworfen Evaluation wurde MRCBR in zwei Versionen mit unterschiedlichen zugrundeliegenden Klassifikations-Algorithmen, Entscheidungsbaum und Random Forest, in Vergleich zu den derzeitigen Methoden gesetzt, welche für CBR im Kontext von unvollständigen Datenbanken zur Verfügung stehen. Diese Methoden schlossen die klassischen Verfahren der Löschung der Daten, die singulären Imputations-Methoden mit statischer Substitution oder Klassifikations-Algorithmen und die speziell für diese Problematik entworfenen CBR Systemen ein, so dass insgesamt 10 unterschiedliche Methoden mit verschiedensten Ausrichtungen verglichen wurden.

Das Verhalten und die Leistung der Methoden wurde durch den mittleren absoluten Fehler, der Standardabweichung und Erweiterungen von beiden in Bezug auf ihre korrekte Rangfolge gegenüber den Ergebnissen auf der wahren vollständigen Datenbank ermittelt.

Die Evaluation wurde unter verschiedenen Bedingungen, was die Anzahl fehlender Daten, den Typ der fehlenden Daten und die Art der Variable angeht, auf einer freien medizinischen Datenbank durchgeführt. Auch der Einfluss der Größe der Datenbank und die Anzahl der verwendeten ähnlichsten Fälle wurde in Betracht gezogen. Für eine optimale statistische Aussagekraft ist das Ergebnis jeder Methode in jedem Versuch der Durchschnitt aus 200 einzelnen Durchläufen, die sich aus einer anderen Zufälligkeit der Löschung und mehreren Zielfällen ergeben.

Leistung und Verhalten der Methoden

Die Experimente dieser Arbeiten in verschiedenen Umgebungen von fehlenden Daten für die Variablen demonstrierten, dass die Methodik des MRCBR in nahezu jedem Versuch die anderen Methoden überflügelte. Vor allem MRCBR mit Random Forest als Imputations-Algorithmus erwies sich als herausragend. Besonders für mehrere unvollständige Variable, große Datenbanken und gemischte Typen von fehlenden Daten konnte es seine Vorteile ausspielen und der Abstand zu den anderen Methoden, bestimmt mit Hilfe der beiden Fehlermaße, wurde signifikant größer.

Die Methodik des MRCBR in beiden Versionen war auch immer weit besser als die singuläre Imputation mit denselben Klassifikations-Algorithmen. Die Substitution des Mean für die fehlenden Daten war zumeist die nächstbeste Methode nach MRCBR, was aufgrund seiner Einfachheit überraschte. Völlig fehl schlugen die bisher bewährten Techniken der Löschung der unvollständigen Variablen oder der Fälle mit fehlenden Daten. Diese Methoden stellten sich als die schlechtesten in allen Versuchen heraus.

Es zeigte sich, dass die Art der unvollständigen Variablen, ob Integer, Gleitkommazahl oder Kategoriale, einen erheblichen Einfluss auf das Verhalten der Methoden ausübte. Für Integer und Gleitkommazahl verhielten sich die Methoden fast identisch,

wogegen für die Kategoriale klare Unterschiede zu erkennen waren. Im Allgemeinen war es für die Methoden schwieriger unvollständige Kategoriale mit guten Ergebnissen zu verarbeiten.

Dies steht in direktem Bezug zu den Typen der fehlenden Daten MCAR, MAR und MNAR, mit welchen die Daten in den Variablen gelöscht wurden. MNAR erwies sich, wie aufgrund seines möglichen totalen Informationsverlustes zu erwarten, als die komplizierteste Umgebung für alle Methoden. Wenn in MCAR die Leistung der besten Methoden noch eng beieinanderlag, klafften in MNAR Lücken zwischen den Methoden. Speziell die Klassifikations-Algorithmen konnten ihre Stärken in dieser Umgebung nicht voll ausspielen. Des Weiteren wurde experimentell gezeigt, dass MAR in seinen Auswirkungen zwischen dem Verhalten von MCAR und MNAR liegt.

Die geschilderten Effekte bezüglich der Art der Variable und dem Typ der fehlenden Daten verstärkten sich in Kombination aus beiden. Wenn einzelne numerischen Variablen in der MCAR Umgebung für die Methoden einfacher zu verarbeiten waren, war eine einzelne Kategoriale in der MNAR Umgebung dies keineswegs. Die Kategoriale mit MNAR war für MRCBR der einzige Versuch, in der es einer anderen Methode, der universellen CBR Methode der Substitution der lokalen Ähnlichkeit mit 0, eindeutig in beiden Fehlern mit einem klaren Abstand unterlag. Unabhängig jedoch von der jeweiligen Umgebung war MRCBR eindeutig überlegen, wenn mehrere unvollständige Variablen gleicher oder verschiedener Art auftraten.

Das künstlich wahre Fall Szenario, welches reale Bedingungen in der Praxis widerspiegelte, in dem es sieben Variablen mit jeweils einem anderen Typ der fehlenden Daten löschte, offenbarte, dass MRCBR unter diesen Voraussetzungen bei weitem die beste Methode war. Für dieses Szenario zeigten unterschiedliche Größen der Datenbank und eine andere Anzahl der verwendeten ähnlichsten Fälle keine Veränderung in der Reihenfolge der Methoden in Bezug auf ihre Leistung. Allerdings wuchs der Vorsprung von MRCBR zu den anderen Methoden für große Datenbanken stetig an.

Die Vergleichsübersicht der Versuche aller Experimente präsentierte beide Versionen von MRCBR grundsätzlich als die alleinigen besten Methoden. Diesen folgte die

Mean Substitution und die anschließenden singulären Klassifikations-Algorithmen. Das Schlusslicht bildeten die klassischen Eliminierungsverfahren und die universellen CBR Methoden zur Verarbeitung von fehlenden Daten.

Dies verdeutlichte, dass das Ignorieren von fehlenden Daten keine sinnvolle Alternative zu modernen Verfahren darstellt. Der auftretende Informationsverlust durch Eliminierung der betroffenen Werte, Fälle oder Variablen führt zu katastrophalen Ergebnissen. Die auf die Retrieve-Phase folgenden anderen Phasen von CBR und damit alle Vorhersagen von CBR sind direkt von deren Resultaten abhängig. Somit ist die möglichst korrekte Rangfolge der ähnlichsten Fälle des Retrievals eine notwendige Bedingung für die Anwendung von CBR auf unvollständigen Datenbanken. Im Hinblick auf die bevorstehende Herausforderung von Big Data, gerade im medizinischen Bereich, und dem damit einhergehenden Problem der steigenden Anzahl fehlender Daten ist es unerlässlich sich um diese Fragestellung zu kümmern und effiziente Lösungen zu finden.

Die Methodik des MRCBR hat in der Evaluation unter Beweis gestellt, dass es den geforderten Erwartungen entspricht und die bisherigen Methoden an Leistung übertrifft. Es gibt die ähnlichsten Fälle aus dem Retrieval möglichst korrekt wieder und verhält sich stabil und verlässlich in so gut wie allen getesteten Umgebungen. In großen Datenbanken und einer hohen Anzahl von unvollständigen Variablen verbesserte es sein Verhalten sogar noch deutlich.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Aamodt, Agnar und Plaza, Enric. "Case-Based Reasoning Foundational Issues, Methodological Variations, and System Approaches". In: *Artif. Intell. Commun.* 7 (1994), S. 39–59. DOI: 10.3233/AIC-1994-7104.

[2] Abdrabou, Essam und Salem, AbdEl-Badeeh M. "A Breast Cancer Classifier based on a Combination of Case-Based Reasoning and Ontology Approach". In: *Proceedings of the IMCSIT. Volume 5*. 2010, S. 3–10. DOI: 10.1109/IMCSIT.2010.5680045.

[3] Aggarwal, Charu C. *Data classification: Algorithms and applications*. Chapman & Hall / CRC data mining and knowledge discovery series. Boca Raton, FL.: CRC / Taylor und Francis, 2014. ISBN: 9781466586741. DOI: 10.1201/b17320.

[4] Agre, Grennady. "KBS maintenance as learning two-tiered domain representation". In: *Case-Based Reasoning Research and Development*. Bd. 1010. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1995, S. 109–120. ISBN: 978-3-540-60598-0. DOI: 10.1007/3-540-60598-3_11.

[5] Ahmed, Mobyen Uddin, Islam, Asif Moinul und Loutfi, Amy. "A case-based patient identification system using pulse oximeter and a personalized health profile". In: *Twentieth International Conference on Case-Based Reasoning (ICCBR 2012)*. 2012. DOI: 10.1016/j.artmed.2010.09.003.

[6] Aljuaid, Tahani und Sasi, Sreela. "Proper imputation techniques for missing values in data sets". In: *International Conference on Data Science and Engi-*

neering (ICDSE). IEEE, 2016, S. 1–5. ISBN: 978-1-5090-1281-7. DOI: 10.1109/ICDSE.2016.7823957.

[7] Allison, Paul D. *Missing data*. Bd. 136. Sage university papers Quantitative applications in the social sciences. Thousand Oaks, Calif: Sage Publ, 2009. ISBN: 9780761916727.

[8] Amailef, Khaled und Lu, Jie. "Ontology-supported case-based reasoning approach for intelligent m-Government emergency response services". In: *Decision Support Systems* 55.1 (2013), S. 79–97. ISSN: 01679236. DOI: 10.1016/j.dss.2012.12.034.

[9] Arakeri, Megha P. und Ram Mohana Reddy, G. "An intelligent content-based image retrieval system for clinical decision support in brain tumor diagnosis". In: *International Journal of Multimedia Information Retrieval* 2.3 (2013), S. 175–188. ISSN: 2192-6611. DOI: 10.1007/s13735-013-0037-5.

[10] Armengol, Eva u. a. "On Learning Similarity Relations in Fuzzy Case-Based Reasoning". In: *Rough sets and fuzzy sets*. Bd. 3135. Lecture Notes in Computer Science. Berlin: Springer, 2005, S. 14–32. ISBN: 978-3-540-23990-1. DOI: 10.1007/978-3-540-27778-1_2.

[11] Bacher, Johann, Pöge, Andreas und Wenzig, Knut. *Clusteranalyse: Anwendungsorientierte Einführung in Klassifikationsverfahren*. München: Oldenbourg, 2010. ISBN: 9783486584578. DOI: 10.1524/9783486710236.

[12] Banerjee, Sreeparna und Chowdhury, Amrita Roy. "Case Based Reasoning in the Detection of Retinal Abnormalities Using Decision Trees". In: *Procedia Computer Science* 46 (2015), S. 402–408. ISSN: 18770509. DOI: 10.1016/j.procs.2015.02.037.

[13] Bareiss, Ray und Chandrasekaran, B. *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning*. Burlington: Elsevier Science, 2014. ISBN: 9780120782604.

[14] Begum, Shahina, Barua, Shaibal und Ahmed, Mobyen Uddin. “Physiological sensor signals classification for healthcare using sensor data fusion and case-based reasoning”. In: *Sensors (Basel, Switzerland)* 14.7 (2014), S. 11770–11785. DOI: 10.3390/s140711770.

[15] Begum, Shahina u. a. “A Case-Based Decision Support System for Individual Stress Diagnosis Using Fuzzy Similarity Matching”. In: *Computational Intelligence* 25.3 (2009), S. 180–195. ISSN: 08247935. DOI: 10.1111/j.1467-8640.2009.00337.x.

[16] Begum, Shahina u. a. “Case-Based Reasoning Systems in the Health Sciences: A Survey of Recent Trends and Developments”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 41.4 (2011), S. 421–434. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2010.2071862.

[17] Bergmann, Ralph u. a. “Case-Based Reasoning - Introduction and Recent Developments”. In: *KI - Künstliche Intelligenz, German Journal on Artificial Intelligence* "Case-Based Reasoning".1 (Jan. 2009), S. 5–11.

[18] Bichindaritz, Isabelle. “Research Themes in the Case-Based Reasoning in Health Sciences Core Literature”. In: *Advances in data mining*. Bd. 7377. Lecture Notes in Computer Science. Berlin: Springer, 2012, S. 9–23. ISBN: 978-3-642-31487-2. DOI: 10.1007/978-3-642-31488-9_2.

[19] Bichindaritz, Isabelle und Marling, Cindy. “Case-based reasoning in the health sciences: What’s next?” In: *Artificial intelligence in medicine* 36.2 (2006), S. 127–135. DOI: 10.1016/j.artmed.2005.10.008.

[20] Billot, Antoine, Gilboa, Itzhak und Schmeidler, David. “Axiomatization of an exponential similarity function”. In: *Mathematical Social Sciences* 55.2 (2008), S. 107–115. ISSN: 01654896. DOI: 10.1016/j.mathsocsci.2007.08.002.

[21] Blanco, Xiomara u. a. “Case-Based Reasoning Applied to Medical Diagnosis and Treatment”. In: *Distributed Computing and Artificial Intelligence*. Bd. 217.

Advances in Intelligent Systems and Computing. Heidelberg: Springer, 2013, S. 137–146. ISBN: 978-3-319-00550-8. DOI: 10.1007/978-3-319-00551-5_17.

[22] Bortz, Jürgen und Schuster, Christof. *Statistik für Human- und Sozialwissenschaftler*. Springer-Lehrbuch. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010. ISBN: 9783642127694. DOI: 10.1007/978-3-642-12770-0.

[23] Breiman, Leo. "Random Forests". In: *Machine Learning* 45.1 (2001), S. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[24] Breiman, Leo u. a. *Classification and regression trees*. Boca Raton: Chapman & Hall, 1984. ISBN: 0412048418. DOI: 10.1002/cyto.990080516.

[25] Bulu, Hakan, Alpkocak, Adil und Balci, Pinar. "Ontology-based mammography annotation and Case-based Retrieval of breast masses". In: *Expert Systems with Applications* 39.12 (2012), S. 11194–11202. ISSN: 09574174. DOI: 10.1016/j.eswa.2012.03.058.

[26] Byar, D. P. und Green, S. B. "The choice of treatment for cancer patients based on covariate information". In: *Bulletin du cancer* 67.4 (1980), S. 477–490. ISSN: 0007-4551.

[27] Carbonell, Jaime G. u. a. *Case-Based Reasoning Research and Development*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. ISBN: 978-3-540-60598-0. DOI: 10.1007/3-540-60598-3.

[28] Choudhury, Nabanita und Ara, Shahin. "A Survey on Case-based Reasoning in Medicine". In: *International Journal of Advanced Computer Science and Applications* 7.8 (2016). ISSN: 2158107X. DOI: 10.14569/IJACSA.2016.070820.

[29] Colantonio, Sara, Perner, Petra und Salvetti, Ovidio. "Diagnosis of Lymphatic Tumors by Case-Based Reasoning on Microscopic Images". In: *Transactions on Case-Based Reasoning* Vol.2 (2008), S. 29–40. ISSN: 1864-9734.

[30] Cunningham, P. “A Taxonomy of Similarity Mechanisms for Case-Based Reasoning”. In: *IEEE Transactions on Knowledge and Data Engineering* 21.11 (2009), S. 1532–1543. ISSN: 1041-4347. DOI: 10.1109/TKDE.2008.227.

[31] Doove, L. L., van Buuren, S. und Dusseldorp, E. “Recursive partitioning for missing data imputation in the presence of interaction effects”. In: *Computational Statistics & Data Analysis* 72 (2014), S. 92–104. ISSN: 01679473. DOI: 10.1016/j.csda.2013.10.025.

[32] Enders, Craig K. *Applied missing data analysis*. Methodology in the social sciences. New York: Guilford Press, 2010. ISBN: 9781606236390.

[33] Enders, Craig K. “Multiple imputation as a flexible tool for missing data handling in clinical research”. In: *Behaviour research and therapy* 98 (2017), S. 4–18. DOI: 10.1016/j.brat.2016.11.008.

[34] Ennett, C. M., Frize, M. und Walker, C. R. “Influence of missing values on artificial neural network performance”. In: *Studies in health technology and informatics* 84.Pt 1 (2001), S. 449–453. ISSN: 0926-9630.

[35] Esfandiari, Nura u. a. “Knowledge discovery in medicine: Current issue and future trend”. In: *Expert Systems with Applications* 41.9 (2014), S. 4434–4463. ISSN: 09574174. DOI: 10.1016/j.eswa.2014.01.011.

[36] El-Fakdi, Andres u. a. “eXiTCDSS: A framework for a workflow-based CBR for interventional Clinical Decision Support Systems and its application to TAVI”. In: *Expert Systems with Applications* 41.2 (2014), S. 284–294. ISSN: 09574174. DOI: 10.1016/j.eswa.2013.05.067.

[37] Fan, Zhi-Ping u. a. “Hybrid similarity measure for case retrieval in CBR and its application to emergency response towards gas explosion”. In: *Expert Systems with Applications* 41.5 (2014), S. 2526–2534. ISSN: 09574174. DOI: 10.1016/j.eswa.2013.09.051.

[38] Finnie, Gavin und Sun, Zhaohao. "Similarity and metrics in case-based reasoning". In: *International Journal of Intelligent Systems* 17.3 (2002), S. 273–287. ISSN: 0884-8173. DOI: 10.1002/int.10021.

[39] Fionov, Andrey. "Case-Based Reasoning in Clinical Processes Using Clinical Data Banks". In: *International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON)*. Piscataway, New Jersey: IEEE, 2015. ISBN: 9781467391108.

[40] García-Laencina, Pedro J. u. a. "Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values". In: *Computers in biology and medicine* 59 (2015), S. 125–133. DOI: 10.1016/j.compbiomed.2015.02.006.

[41] Gierl, Lothar, Bull, Mathias und Schmidt, Rainer. "CBR in Medicine". In: *Case-Based Reasoning Technology*. Bd. 1400. Lecture Notes in Computer Science. Berlin und Heidelberg: Springer, 1998, S. 273–297. ISBN: 9783540645726. DOI: 10.1007/3-540-69351-3_11.

[42] Gómez-Vallejo, H. J. u. a. "A case-based reasoning system for aiding detection and classification of nosocomial infections". In: *Decision Support Systems* 84 (2016), S. 104–116. ISSN: 01679236. DOI: 10.1016/j.dss.2016.02.005.

[43] González, Carolina, López, Diego M. und Blobel, Bernd. "Case-based reasoning in Intelligent Health Decision Support Systems". In: *Studies in health technology and informatics* 189 (2013), S. 44–49. ISSN: 0926-9630.

[44] Gu, Dongxiao, Liang, Changyong und Zhao, Huimin. "A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis". In: *Artificial intelligence in medicine* 77 (2017), S. 31–47. DOI: 10.1016/j.artmed.2017.02.003.

[45] Guerdjikova, Ani. "Case-based learning with different similarity functions". In: *Games and Economic Behavior* 63.1 (2008), S. 107–132. ISSN: 08998256. DOI: 10.1016/j.geb.2007.10.004.

[46] Guessoum, Souad, Laskri, Mohamed Tayeb und Lieber, Jean. “RespiDiag: A Case-Based Reasoning System for the Diagnosis of Chronic Obstructive Pulmonary Disease”. In: *Expert Systems with Applications* 41.2 (2014), S. 267–273. ISSN: 09574174. DOI: 10.1016/j.eswa.2013.05.065.

[47] Hidalgo, J. Ignacio u. a. “glUCModel: a monitoring and modeling system for chronic diseases applied to diabetes”. In: *Journal of biomedical informatics* 48 (2014), S. 183–192. DOI: 10.1016/j.jbi.2013.12.015.

[48] Holt, Alex u. a. “Medical applications in case-based reasoning”. In: *The Knowledge Engineering Review* 20.03 (2005), S. 289. ISSN: 0269-8889. DOI: 10.1017/S0269888906000622.

[49] Hönigl, Jürgen und Küng, Josef. “A Data Quality Index with Respect to Case Bases within Case-Based Reasoning”. In: *Intelligent Information and Database Systems*. Bd. 8397. Lecture Notes in Computer Science. Berlin: Springer, 2014, S. 432–442. ISBN: 978-3-319-05475-9. DOI: 10.1007/978-3-319-05476-6_44.

[50] Horton, Nicholas J. und Kleinman, Ken P. “Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models”. In: *The American statistician* 61.1 (2007), S. 79–90. ISSN: 0003-1305. DOI: 10.1198/000313007X172556.

[51] Hüllermeier, Eyke. *Case-Based Approximate Reasoning*. Bd. 44. Theory and Decision Library. Berlin: Springer, 2007. ISBN: 9781402056949. DOI: 10.1007/1-4020-5695-8.

[52] Huque, Md Hamidul u. a. “A comparison of multiple imputation methods for missing data in longitudinal studies”. In: *BMC medical research methodology* 18.1 (2018), S. 168. DOI: 10.1186/s12874-018-0615-6.

[53] Ibrahim, Joseph G., Chu, Haitao und Chen, Ming-Hui. “Missing data in clinical studies: issues and methods”. In: *Journal of clinical oncology : official journal of the American Society of Clinical Oncology* 30.26 (2012), S. 3297–3303. DOI: 10.1200/JCO.2011.38.7589.

[54] Jagannathan, Rupa und Petrovic, Sanja. "A Local Rule-Based Attribute Weighting Scheme for a Case-Based Reasoning System for Radiotherapy Treatment Planning". In: *Case-Based Reasoning Research and Development*. Bd. 7466. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, S. 167–181. ISBN: 978-3-642-32985-2. DOI: 10.1007/978-3-642-32986-9_14.

[55] Jagannathan, Rupa und Petrovic, Sanja. "Dealing with missing values in a clinical case-based reasoning system". In: *2nd IEEE International Conference on Computer Science and Information Technology*. IEEE, 2009, S. 120–124. ISBN: 978-1-4244-4519-6. DOI: 10.1109/ICCSIT.2009.5234442.

[56] James, Gareth u. a. *An Introduction to Statistical Learning*. Bd. 103. New York, NY: Springer, 2013. ISBN: 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7.

[57] Jamshidian, Mortaza und Jalal, Siavash. "Tests of homoscedasticity, normality, and missing completely at random for incomplete multivariate data". In: *Psychometrika* 75.4 (2010), S. 649–674. ISSN: 0033-3123. DOI: 10.1007/s11336-010-9175-3.

[58] Kang, Hyun. "The prevention and handling of the missing data". In: *Korean journal of anesthesiology* 64.5 (2013), S. 402–406. ISSN: 2005-6419. DOI: 10.4097/kjae.2013.64.5.402.

[59] Khussainova, Gulmira, Petrovic, Sanja und Jagannathan, Rupa. "Retrieval with Clustering in a Case-Based Reasoning System for Radiotherapy Treatment Planning". In: *Journal of Physics: Conference Series* 616 (2015), S. 012013. ISSN: 1742-6588. DOI: 10.1088/1742-6596/616/1/012013.

[60] Leal, Yenny u. a. "Principal component analysis in combination with case-based reasoning for detecting therapeutically correct and incorrect measurements in continuous glucose monitoring systems". In: *Biomedical Signal Processing and Control* 8.6 (2013), S. 603–614. ISSN: 17468094. DOI: 10.1016/j.bspc.2013.05.008.

[61] Lee, Hyun Jung und Kim, Hee Sun. “eHealth Recommendation Service System Using Ontology and Case-Based Reasoning”. In: *IEEE International Conference on Smart City*. IEEE, 2015, S. 1108–1113. ISBN: 978-1-5090-1893-2. DOI: 10.1109/SmartCity.2015.217.

[62] Lee, Katherine J. und Simpson, Julie A. “Introduction to multiple imputation for dealing with missing data”. In: *Respirology (Carlton, Vic.)* 19.2 (2014), S. 162–167. DOI: 10.1111/resp.12226.

[63] Lenz, Mario u. a. *Case-Based Reasoning Technology: From Foundations to Applications*. Bd. 1400. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1998. ISBN: 9783540645726. DOI: 10.1007/3-540-69351-3.

[64] Li, Cheng. “Little’s Test of Missing Completely at Random”. In: *The Stata Journal: Promoting communications on statistics and Stata* 13.4 (2013), S. 795–809. ISSN: 1536-867X. DOI: 10.1177/1536867X1301300407.

[65] Li, Peng, Stuart, Elizabeth A. und Allison, David B. “Multiple Imputation: A Flexible Tool for Handling Missing Data”. In: *JAMA* 314.18 (2015), S. 1966–1967. DOI: 10.1001/jama.2015.15281.

[66] Liao, T. Warren, Zhang, Zhiming und Mount, Claude R. “Similarity measures for retrieval in case-based reasoning systems”. In: *Applied Artificial Intelligence* 12.4 (1998), S. 267–288. ISSN: 0883-9514. DOI: 10.1080/088395198117730.

[67] Liaw, Andy und Wiener, Matthew. *Breiman and Cutler’s Random Forests for Classification and Regression*. R package version 4.6-14. 2018. URL: https://cran.r-project.org/web/packages/randomForest/.

[68] Little, Roderick J. A. “A Test of Missing Completely at Random for Multivariate Data with Missing Values”. In: *Journal of the American Statistical Association* 83.404 (1988), S. 1198–1202. ISSN: 0162-1459. DOI: 10.1080/01621459.1988.10478722.

[69] Löw, Nikolas, Hesser, Jürgen und Blessing, Manuel. "Multiple retrieval case-based reasoning for incomplete datasets". In: *Journal of biomedical informatics* 92 (2019), S. 103127. DOI: 10.1016/j.jbi.2019.103127.

[70] Maher, Mary Lou, Balachandran, Muthaukumar und Zhang, Dong Mei. *Case-based reasoning in design*. Mahwah, NJ: Lawrence Erlbaum Associates, 1995. ISBN: 0805818324.

[71] Malekpoor, Hanif u. a. "An efficient approach to radiotherapy dose planning problem: a TOPSIS case-based reasoning approach". In: *International Journal of Systems Science: Operations & Logistics* 4.1 (2017), S. 4–12. ISSN: 2330-2674. DOI: 10.1080/23302674.2015.1135354.

[72] Marling, Cindy u. a. "The 4 Diabetes Support System: A Case Study in CBR Research and Development". In: Bd. 6880. Lecture notes in computer science Lecture notes in artificial intelligence. Berlin: Springer, 2011, S. 137–150. ISBN: 978-3-642-23290-9. DOI: 10.1007/978-3-642-23291-6_12.

[73] McSherry, David. "Precision and Recall in Interactive Case-Based Reasoning". In: *Case-Based Reasoning Research and Development*. Bd. 2080. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, S. 392–406. ISBN: 978-3-540-42358-4. DOI: 10.1007/3-540-44593-5_28.

[74] Mingyang Gu, Xin Tong und Agnar Aamodt. "Comparing Similarity Calculation Methods in Conversational CBR". In: *IEEE International Conference on Information Reuse and Integration*. Piscataway, NJ: IEEE Operations Center, 2005. ISBN: 0780390938.

[75] Miranda, Gisele Helena Barboni und Felipe, Joaquim Cezar. "Computer-aided diagnosis system based on fuzzy logic for breast cancer categorization". In: *Computers in biology and medicine* 64 (2015), S. 334–346. DOI: 10.1016/j.compbiomed.2014.10.006.

[76] Mohammed, Mazin Abed u. a. “Genetic case-based reasoning for improved mobile phone faults diagnosis”. In: *Computers & Electrical Engineering* 71 (2018), S. 212–222. ISSN: 00457906. DOI: 10.1016/j.compeleceng.2018.07.053.

[77] Molenberghs, Geert und Kenward, Michael G. *Missing Data in Clinical Studies*. Chichester, UK: John Wiley & Sons, Ltd, 2007. ISBN: 9780470510445. DOI: 10.1002/9780470510445.

[78] O’Kelly, Michael und Ratitch, Bohdana. *Clinical Trials with Missing Data: A Guide for Practitioners*. 1. Aufl. Statistics in Practice. s.l.: Wiley, 2014. ISBN: 9781306473118. DOI: 10.1002/9781118762516.

[79] Orchard, Terence und Woodbury, Max A. “A missing information principle: theory and applications”. In: Berkeley, Calif.: University of California Press, 1972, S. 697–715. DOI: 10.1111/j.1558-5646.1974.tb00803.x.

[80] Pal, Sankar K. und Shiu, Simon C. K. *Foundations of soft case-based reasoning*. Wiley series on intelligent systems. Hoboken, N.J: Wiley-Interscience, 2010. ISBN: 9780471644675. DOI: 10.1002/0471644676.

[81] Pampaka, Maria, Hutcheson, Graeme und Williams, Julian. “Handling missing data: analysis of a challenging data set using multiple imputation”. In: *International Journal of Research & Method in Education* 39.1 (2016), S. 19–37. ISSN: 1743-727X. DOI: 10.1080/1743727X.2014.979146.

[82] Pedersen, Alma B. u. a. “Missing data and multiple imputation in clinical epidemiological research”. In: *Clinical epidemiology* 9 (2017), S. 157–166. ISSN: 1179-1349. DOI: 10.2147/CLEP.S129785.

[83] Perner, Petra. “Case-Based Reasoning and the Statistical Challenges II”. In: *Man-machine interactions 3*. Bd. 242. Advances in Intelligent Systems and Computing. Cham: Springer, 2014, S. 17–38. ISBN: 978-3-319-02308-3. DOI: 10.1007/978-3-319-02309-0_2.

[84] Petrovic, Sanja, Khussainova, Gulmira und Jagannathan, Rupa. "Knowledge-light adaptation approaches in case-based reasoning for radiotherapy treatment planning". In: *Artificial intelligence in medicine* 68 (2016), S. 17–28. DOI: 10.1016/j.artmed.2016.01.006.

[85] Petrovic, Sanja, Mishra, Nishikant und Sundar, Santhanam. "A novel case based reasoning approach to radiotherapy planning". In: *Expert Systems with Applications* 38.9 (2011), S. 10759–10769. ISSN: 09574174. DOI: 10.1016/j.eswa.2011.01.109.

[86] Ping, Xiao-Ou u. a. "A multiple measurements case-based reasoning method for predicting recurrent status of liver cancer patients". In: *Computers in Industry* 69 (2015). ISSN: 01663615. DOI: 10.1016/j.compind.2015.01.007.

[87] Qi, Jin, Hu, Jie und Peng, Yinghong. "A new adaptation method based on adaptability under k-nearest neighbors for case adaptation in case-based design". In: *Expert Systems with Applications* 39.7 (2012), S. 6485–6502. ISSN: 09574174. DOI: 10.1016/j.eswa.2011.12.055.

[88] Qu, Gang u. a. "Study on Self-Adaptive Clinical Pathway Decision Support System Based on Case-Based Reasoning". In: Bd. 269. Lecture Notes in Electrical Engineering. Dordrecht: Springer, 2014, S. 969–978. ISBN: 978-94-007-7617-3. DOI: 10.1007/978-94-007-7618-0_95.

[89] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017. URL: https://www.R-project.org.

[90] Reynolds, A. P. u. a. "Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms". In: *Journal of Mathematical Modelling and Algorithms* 5.4 (2006), S. 475–504. ISSN: 1570-1166. DOI: 10.1007/s10852-005-9022-1.

[91] Ricci, Francesco und Avesani, Paolo. “Learning a local similarity metric for case-based reasoning”. In: *Case-Based Reasoning Research and Development*. Bd. 1010. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1995, S. 301–312. ISBN: 978-3-540-60598-0. DOI: 10.1007/3-540-60598-3_27.

[92] Richter, Michael M. und Weber, Rosina O. *Case-based reasoning: A textbook*. Berlin: Springer, 2013. ISBN: 9783642401671.

[93] Rubin, Donald. B. “Inference and missing data”. In: *Biometrika* 63.3 (1976), S. 581–592. ISSN: 0006-3444. DOI: 10.1093/biomet/63.3.581.

[94] Rubin, Donald. B. *Multiple Imputation for nonresponse in surveys*. Wiley series in probability and mathematical statistics Applied probability and statistics. New York, NY: Wiley, 1987. ISBN: 9780471655749. DOI: 10.1002/9780470316696.

[95] Salgado, Cátia M. u. a. “Missing Data”. In: *Secondary Analysis of Electronic Health Records*. Bd. 58. Springer, 2016, S. 143–162. ISBN: 978-3-319-43740-8. DOI: 10.1007/978-3-319-43742-2_13.

[96] El-Sappagh, Shaker und Elmogy, Mohammed Mahfouz. “Medical Case Based Reasoning Frameworks: Current Developments and Future Directions”. In: *International Journal of Decision Support System Technology* 8.3 (2016), S. 31–62. ISSN: 1941-6296. DOI: 10.4018/IJDSST.2016070103.

[97] El-Sappagh, Shaker, Elmogy, Mohammed und Riad, Alaa Eldin M. “A CBR system for diabetes mellitus diagnosis: case-base standard data model”. In: *International Journal of Medical Engineering and Informatics* 7.3 (2015), S. 191. ISSN: 1755-0653. DOI: 10.1504/IJMEI.2015.070116.

[98] Saraiva, Renata u. a. “Early diagnosis of gastrointestinal cancer by using case-based and rule-based reasoning”. In: *Expert Systems with Applications* 61 (2016), S. 192–202. ISSN: 09574174. DOI: 10.1016/j.eswa.2016.05.026.

[99] Sarkar, Indra Neil, Georgiou, Andrew und Marques, Paulo Mazzoncini de Azevedo. *A Hybrid Approach Using Case-Based Reasoning and Rule-Based Reasoning to Support Cancer Diagnosis: A Pilot Study*. Bd. volume 216, Part 1-2. Studies in health technology and informatics. Amsterdam: IOS Press, 2015. ISBN: 161499563X.

[100] Sartori, Fabio, Mazzucchelli, Alice und Di Gregorio, Angelo. "Bankruptcy forecasting using case-based reasoning: The CRePERIE approach". In: *Expert Systems with Applications* 64 (2016), S. 400–411. ISSN: 09574174. DOI: 10.1016/j.eswa.2016.07.033.

[101] Schank, Roger C. und Abelson, Robert P. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. The artificial intelligence series. Hillsdale, NJ: Erlbaum, 1977. ISBN: 0898591384.

[102] Schmidt, R. und Gierl, L. "Case-based reasoning for medical knowledge-based systems". In: *Studies in health technology and informatics* 77 (2000), S. 720–725. ISSN: 0926-9630.

[103] Schmidt, Rainer und Vorobieva, Olga. "Applying Case-Based Reasoning for Missing Medical Data in ISOR". In: *LWA 2007*. 2007, S. 275–280.

[104] Schober, Patrick, Boer, Christa und Schwarte, Lothar A. "Correlation Coefficients: Appropriate Use and Interpretation". In: *Anesthesia and analgesia* 126.5 (2018), S. 1763–1768. DOI: 10.1213/ANE.0000000000002864.

[105] Shah, Anoop D. u. a. "Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data Using MICE: A CALIBER Study". In: *American Journal of Epidemiology* 179.6 (2014), S. 764–774. ISSN: 0002-9262. DOI: 10.1093/aje/kwt312.

[106] Shen, Ying u. a. "Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system". In: *Journal of biomedical informatics* 56 (2015), S. 307–317. DOI: 10.1016/j.jbi.2015.06.012.

[107] Shokouhi, Samad Valipour, Skalle, Pål und Aamodt, Agnar. “An overview of case-based reasoning applications in drilling engineering”. In: *Artificial Intelligence Review* 41.3 (2014), S. 317–329. ISSN: 0269-2821. DOI: 10.1007/s10462-011-9310-2.

[108] Sterne, Jonathan A. C. u. a. “Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls”. In: *BMJ (Clinical research ed.)* 338 (2009). DOI: 10.1136/bmj.b2393.

[109] Stroock, Daniel W. *Probability theory: An analytic view*. Cambridge: Cambridge University Press, 2011. ISBN: 9780521132503. DOI: 10.1017/CBO9780511974243.

[110] Surma, Jerzy und Vanhoof, Koen. “Integrating rules and cases for the classification task”. In: Bd. 1010. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1995, S. 325–334. ISBN: 978-3-540-60598-0. DOI: 10.1007/3-540-60598-3_29.

[111] Tang, Fei und Ishwaran, Hemant. “Random Forest Missing Data Algorithms”. In: *Statistical analysis and data mining* 10.6 (2017), S. 363–377. ISSN: 1932-1864. DOI: 10.1002/sam.11348.

[112] Teodorović, Dušan, Šelmić, Milica und Mijatović-Teodorović, Ljiljana. “Combining case-based reasoning with Bee Colony Optimization for dose planning in well differentiated thyroid cancer treatment”. In: *Expert Systems with Applications* 40.6 (2013), S. 2147–2155. ISSN: 09574174. DOI: 10.1016/j.eswa.2012.10.027.

[113] Therneau, Terry und Atkinson, Beth. *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13. 2018. URL: https://cran.r-project.org/web/packages/rpart/.

[114] Thompson, Ken u. a. “Little evidence for limiting similarity in a long-term study of a roadside plant community”. In: *Journal of Ecology* 98.2 (2010), S. 480–487. ISSN: 00220477. DOI: 10.1111/j.1365-2745.2009.01610.x.

[115] van Buuren, Stef. *Flexible imputation of missing data*. Chapman & Hall/CRC interdisciplinary statistics series. CRC Press, 2012. ISBN: 9781439868249.

[116] van Buuren, Stef. "Multiple imputation of discrete and continuous data by fully conditional specification". In: *Statistical methods in medical research* 16.3 (2007), S. 219–242. ISSN: 0962-2802. DOI: 10.1177/0962280206074463.

[117] van Buuren, Stef und Groothuis-Oudshoorn, Karin. *mice: Multivariate Imputation by Chained Equations in R*. R package version 3.4.0. 2018. URL: https://cran.r-project.org/web/packages/mice/.

[118] van Buuren, Stef u. a. "Fully conditional specification in multivariate imputation". In: *Journal of Statistical Computation and Simulation* 76.12 (2006), S. 1049–1064. ISSN: 0094-9655. DOI: 10.1080/10629360600810434.

[119] Vorobieva, Olga, Rumyantsev, Alexander und Schmidt, Rainer. "A CBR Solution for Missing Medical Data". In: *5th Workshop on CBR in the Health Sciences*. Bd. 2007. 2007.

[120] Waljee, Akbar K. u. a. "Comparison of imputation methods for missing laboratory data in medicine". In: *BMJ open* 3.8 (2013). ISSN: 2044-6055. DOI: 10.1136/bmjopen-2013-002847.

[121] Willmott, C. J. und Matsuura, K. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". In: *Climate Research* 30 (2005), S. 79–82. ISSN: 0936-577X. DOI: 10.3354/cr030079.

[122] Xie, Xiaolong, Lin, Lin und Zhong, Shisheng. "Handling missing values and unmatched features in a CBR system for hydro-generator design". In: *Computer-Aided Design* 45.6 (2013), S. 963–976. ISSN: 00104485. DOI: 10.1016/j.cad.2013.02.004.

[123] Yeow, Wei Liang, Mahmud, Rohana und Raj, Ram Gopal. “An application of case-based reasoning with machine learning for forensic autopsy”. In: *Expert Systems with Applications* 41.7 (2014), S. 3497–3505. ISSN: 09574174. DOI: 10.1016/j.eswa.2013.10.054.

[124] Zhong, Qiuyan u. a. “The Method of Case Retrieving in the Emergency Field Based on CBR”. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, 2010, S. 53–56. ISBN: 978-1-4244-8482-9. DOI: 10.1109/WI-IAT.2010.112.

[125] Zhu, Guo-Niu u. a. “An integrated feature selection and cluster analysis techniques for case-based reasoning”. In: *Engineering Applications of Artificial Intelligence* 39 (2015), S. 14–22. ISSN: 09521976. DOI: 10.1016/j.engappai.2014.11.006.

Lebenslauf

Personalien

Name und Vorname:	Nikolas Immanuel Löw
Geburtsdatum:	04.12.1980
Geburtsort:	Baden-Baden
Familienstand:	Verheiratet

Ausbildung

Aug 1987 - Jun 1991:	Klosterschule Lichtenthal
Aug 1991 - Jun 2000:	Gymnasium Hohenbaden
Okt 2001 - Jun 2011:	Mathematik Diplom mit Nebenfach Philosophie an der Universität Heidelberg, Mathematische Fakultät

Berufserfahrung

Jun 2000 - Mai 2001:	Zivildienst in der Pfarrgemeinde Lichtenthal
Feb 2002 - Okt 2011:	Nachhilfelehrer für Studierende in Mathematik
Okt 2007 - Sep 2009:	Wissenschaftliche Hilfskraft der Bereichsbibliothek für Mathematik und Informatik der Universität Heidelberg
Jul 2011 - Mrz 2013:	Gap Year
Apr 2013 - Heute:	Wissenschaftlicher Mitarbeiter der experimentellen Strahlentherapie des Universitätsklinikums Mannheim

Danksagung

Begegnet uns jemand, der uns Dank schuldig ist, gleich fällt es uns ein.
Wie oft können wir jemand begegnen, dem wir Dank schuldig sind,
ohne daran zu denken!

Johann Wolfgang von Goethe 1809

Der Weg, den wir gehen, wird von vielen Wesen geprägt, die uns mit Wohlwollen auf den rechten Pfad führen. Ohne sie ist das Leben undenkbar und brachte mich an eben jenen Punkt, an dem ich nun stehe. So viel schwerer ist es all diesen ihren gebührenden Dank auszusprechen. Daher werden in den nächsten Zeilen nur jene Personen erwähnt, die einen direkten Anteil am Schaffen und Werden dieser vorliegenden Arbeit hatten.

An erster Stelle bin ich meinem Doktorvater Prof. Dr. rer. nat. Jürgen zu großem Dank verpflichtet. Er schenkte mir seinen konstruktiven Rat und stand mir zur Seite, wenn ich seine Unterstützung brauchte. Genauso wie er mir den kreativen Freiraum schaffte und die Zeit gewährte an den Themen meiner Wahl zu forschen und diese zu vertiefen. Nicht zuletzt bot er mir durch eine Anstellung am Universitätsklinikum die Gelegenheit meine Kenntnisse und Errungenschaften auch weiterzureichen.

Meiner Frau Dr. rer. nat. Alena Löw danke ich für ihre klugen Anregungen und die Korrekturen all meiner Arbeiten. Und für ihre nicht enden wollende Geduld, in den dunklen Stunden der Verzweiflung, die jeder Wissenschaftler nur zu gut kennt.

Jedem Kollegen der Arbeitsgruppe, welcher in vielen kleinen Hilfestellungen einen Beitrag leistete, spreche ich meinen Dank aus. Besonders Dr. rer. nat. Manuel Blessing und Lei Zheng möchte ich für ihre Denkanstöße danken.