2 Erster Hauptteil: Der Blick der Informatik auf maschinelles Lernen

2.1 Betrachtungsebenen

Ziel des ersten Hauptteils ist es, eine interdisziplinär verständliche technikнahe Beschreibung maschinellen Lernens als konkretem Teil der Informatik zu erstellen. Zu diesem Zweck wird eine geisteswissenschaftliche Innenansicht der Perspektive der Informatik erstellt. Einleitend hierzu noch einmal die bereits genannte Definition maschinellen Lernens:

> »Verfahren des machine learning sind die Grundlage von Programmsystemen, die aus ›Erfahrung‹ lernen, also neues Tatsachen- und Regelwissen gewinnen oder Priorisierungen adaptieren können. Sie sind u.a. auch für die Entdeckung zweckbestimmt relevanter Beziehungen in großen Datenmengen (›Data mining‹) von großer Bedeutung.«
>
> (Görz et al. 2003, S. 13)

Vor einer Aufarbeitung des maschinellen Lernens muss entschieden werden, auf welcher Abstraktionsebene das Gebiet untersucht werden soll. Auf der *höchsten* Ebene wären Themen der THEORETISCHEN INFORMATIK wie Berechenbarkeitstheorie oder Komplexitätstheorie zu behandeln. Entsprechend der dieser Arbeit zugrunde liegenden Idee soll keine Diskussion von Metaperspektiven auf die Informatik oder auf Algorithmen stattfinden. Die nachfolgenden Betrachtungen beschreiben auf einer *zweiten*, weniger abstrakten Ebene die Auflösung des maschinellen Lernens in Klassen von Al-

gorithmen und basieren auf Standardwerken der Informatik (Mitchell 1997; Görz et al. 2003; Russell et al. 2007; Alpaydin 2008; Brause 2010; Burkhard 2010) und der Wikipedia[1] (Wikipedia 2001). Die Auswahl und Hierarchisierung der LERNSTRATEGIEN stellt jedoch eine genuin eigene Klassifikation dar, die im Hinblick auf die interdisziplinäre Relevanz erstellt werden wird. In der Literatur liegen viele unterschiedliche Varianten der Bestimmung von Lernstrategien vor[2], demnach stimmt die in Abschnitt 2.3 vorgenommene Klassifikation mit manchen dieser Varianten zu größeren Teilen und mit manchen nur sehr eingeschränkt überein. Die *dritte* mögliche Abstraktionsebene entspricht der Verwendung von PSEUDOCODE zur Diskussion von Algorithmen. Ein Beispiel für einen in Pseudocode geschriebenen Algorithmus ist das folgende Pseudoprogramm.

Abbildung 1: Beispiel für Pseudocode

```
Programm: Essen_kochen
Variablen: Appetit, Gericht, Einkaufsliste
• Gericht = Gericht_auswählen(Appetit)
• Einkaufsliste = Liste_erstellen(Gericht)
• WIEDERHOLE
    Einkaufen
• BIS Einkaufsliste = LEER
• Kochen(Gericht)
• ENDE
```

Solcher Pseudocode kann verwendet werden, um über ein Programm zu sprechen, ohne die Syntax einer speziellen Programmiersprache verwenden zu müssen. Im Weiteren wird es um die Veranschaulichung von Algorithmen gehen und gelegentlich wird Pseudocode verwendet werden, um diese

1 Grundlage sind verschiedene Artikel der deutsch-, englisch- und in geringem Maße der französischsprachigen Version der Wikipedia. Zitiert wird im Weiteren jedoch ausschließlich aus der deutschsprachigen Version.

2 Sogar die englische, deutsche und französische Version der Wikipedia unterscheiden sich stark. Die deutsche Wikipedia führt etwa zunächst keine Unterscheidung zwischen Klassen maschinell lernender Algorithmen durch, sondern ordnet diese nach überwachtem, unüberwachtem und bestärkendem Lernen.

Algorithmen zu veranschaulichen. Diese Abstraktionsebene ist passend für die interdisziplinäre Diskussion maschinellen Lernens, allerdings müssen die nächsten beiden Ebenen zumindest noch mitgedacht werden. Auf der *vierten* Abstraktionsebene werden konkrete Algorithmen in einer speziellen Programmiersprache betrachtet. Ein Stück Pseudocode kann sehr unterschiedlich in konkrete Algorithmen umgeformt werden. Einige im Weiteren genannte Beispiele und Leistungskennzahlen werden sich auf konkrete Algorithmen beziehen, die Diskussion wird jedoch auf der dritten oder auf der zweiten Abstraktionsebene stattfinden und lediglich zwischen einzelnen Algorithmenklassen unterscheiden. Dieses Auflösungsvermögen ist bereits eine vergleichsweise starke Forderung, da in der Technikphilosophie üblicherweise nicht zwischen autoadaptiven und nicht-autoadaptiven Algorithmen unterschieden wird. Ein gewisses Verständnis der konkreten Algorithmen ist dennoch hilfreich, um ungefähr zu überblicken welche Leistungsfähigkeit verschiedene Formen maschinellen Lernens in der Praxis aufweisen. Die über die Betrachtung der konkreten Algorithmen hinausgehende *fünfte* Abstraktionsebene schließlich ist die Implementierung eines Algorithmus in einem physischen Objekt. Häufig werden die vierte und die fünfte Ebene nicht unterschieden, da die Programmierung in einer speziellen Programmiersprache nur theoretisch unabhängig von der Implementierung in einem elektronischen Bauteil erfolgen kann. In der Praxis des Programmierens werden ständig programmexterne Ressourcen beziehungsweise Funktionen benötigt und viele Details der Syntax werden üblicherweise von der Programmierumgebung automatisch beachtet. Dennoch ist es theoretisch möglich, die beiden Ebenen zu trennen, etwa kann man einen konkreten Algorithmus auch auf einem Stück Papier ausdrucken. Die Unterscheidung der vierten und fünften Ebene soll an dieser Stelle lediglich explizit machen, dass auch MLA immer schon eine fünfte Ebene aufweisen und dass deswegen nicht pauschal von maschinell lernenden *Algorithmen* gesprochen werden sollte[3]. Wenn die Hardware mitgedacht wird, können viele Missverständnisse vermieden werden, insbesondere solche über den Zusammenhang zwischen einer Nutzeroberfläche und einem MLA, sowie die Schwierigkeiten bei der Feststellung, ob ein MLA im jeweils aktuellen

3 Ein Algorithmus selbst kann nicht auf Reize reagieren, da er hierfür Eingabedaten benötigt, die erst dann auftreten können, wenn er in einem elektronischen Bauteil implementiert wurde.

Zustand noch lernt oder ob der Lernvorgang schon abgeschlossen ist. So wie ein Künstler nicht mit seinem Kunstwerk identisch ist, lassen sich auch MLA von ihren Strukturvorschlägen unterscheiden und dies gelingt besser, wenn die Hardware mitgedacht wird. Die Abkürzung MLA bezieht sich dementsprechend gezielt auf Artefakte, auch wenn die genaue Ausprägung des Artefaktes im Weiteren die meiste Zeit nicht von besonderer Bedeutung ist.

Eine Diskussion, bei der die konkrete Form des Artefaktes zentral *ist*, ist diejenige um das UBIQUITOUS COMPUTING oder UBICOMP beziehungsweise die AMBIENT INTELLIGENCE[4]. Das UbiComp wird im Weiteren nicht im Fokus stehen, allerdings erscheint es zunächst wie eine Betrachtung des maschinellen Lernens auf der fünften Abstraktionsebene. Zur Vermeidung einer falschen Erwartung an die nachfolgende Diskussion und zum besseren Verständnis der fünften Abstraktionsebene soll daher kurz dargestellt werden, was unter Ubiquitous Computing verstanden werden kann. Im UbiComp geht es insbesondere darum, SMARTE Artefakte zu konzeptionieren und zu konstruieren, das heißt Artefakte, die einen Nutzerwunsch antizipieren und erfüllen, bevor der Nutzer den Wunsch äußern kann oder muss. Die Artefakte treten dabei selbst nicht in Erscheinung, sondern sind ein unaufdringlicher Teil der Umwelt des Nutzers. Ein Beispiel ist die automatische Aktivierung der Beleuchtung, wenn man einen Raum betritt. Viele Artefakte des UbiComp basieren auf maschinellem Lernen, da ein Artefakt, das mittels eines Autoadaptionsprozesses aus den Aktionen des Nutzers lernt, besonders gut geeignet, ist zukünftige Aktionen zu antizipieren und smart zu agieren, bevor der Nutzer aktiv werden muss. Auch wenn das konkrete Artefakt im UbiComp nicht lernt, ist in dessen Erstellung häufig maschinelles Lernen eingeflossen, da in vielen Fällen automatisch das Verhalten von vielen anderen Nutzern beobachtet wurde, um ein Artefakt konstruieren zu können, das besonders gute Chancen hat, unauffällig im Hintergrund arbeiten zu können. Maschinelles Lernen wird daher häufig mitgedacht, wenn vom Bereich des UbiComp die Rede ist. Einerseits ist es jedoch in der diesbezüglichen Diskussion nicht immer zentral, woher die Artefakte die Regeln für ihre Vorgehensweise erhalten haben und anderer-

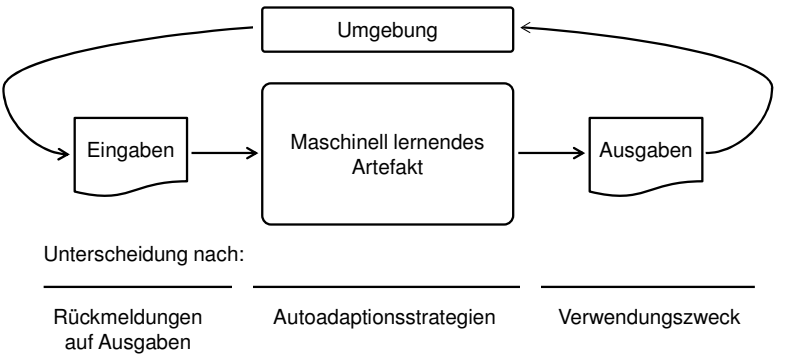
4 Beide Begriffe werden aktuell innerhalb der Informatik annähernd synonym verwendet, wobei die Bezeichnung Ubiquitous Computing häufiger anzutreffen ist.

seits muss ein MLA nicht zwangsweise unspürbar im Hintergrund agieren. UbiComp ist entsprechend zwar eine zentrale, aber nicht die einzige Anwendungsmöglichkeit für maschinelles Lernen. Ein Anknüpfungspunkt an die Verbindung zwischen UbiComp und der im zweiten Hauptteil entworfenen Welttechnik besteht in dem Konzept des FLOWS (Hassenzahl 2004, S. 16). Der Flow beschreibt die selbstvergessene Nutzung eines Artefaktes und stellt gegebenenfalls ein verwandtes Phänomen zu UbiComp und Welttechnik dar.

2.2 UNTERSCHIEDE ZWISCHEN LERNENDEN ALGORITHMEN

Nachdem die Auflösung der Analyse festgelegt wurde, ist der nächste Schritt einerseits zu bestimmen, nach welchen Kriterien maschinell lernende Algorithmen unterschieden und andererseits, wie sie zu Algorithmenklassen – den LERN- beziehungsweise AUTOADAPTIONSSTRATEGIEN – zusammengefasst werden können. Im Folgenden sollen zuerst die in der Informatik gebräuchlichen Unterscheidungsmerkmale zwischen lernenden Algorithmen dargestellt werden.

Abbildung 2: Möglichkeiten zur Unterscheidung von MLA



Diese Unterscheidungen finden sich in ähnlicher Form in jedem der eingangs genannten Standardwerke zu maschinellem Lernen. Die Unterscheidungen betreffen den gesamten Bereich des maschinellen Lernens und werden im Folgenden als Einstieg genutzt, um die für das maschinelle Lernen relevanten formalen Grundbegriffe der Informatik einzuführen. Die

Lernstrategien fallen mitunter etwas spezieller aus und bringen jeweils eigene, klassenspezifische Vokabeln mit.

2.2.1 Unterscheidung gemäß erhaltener Rückmeldungen

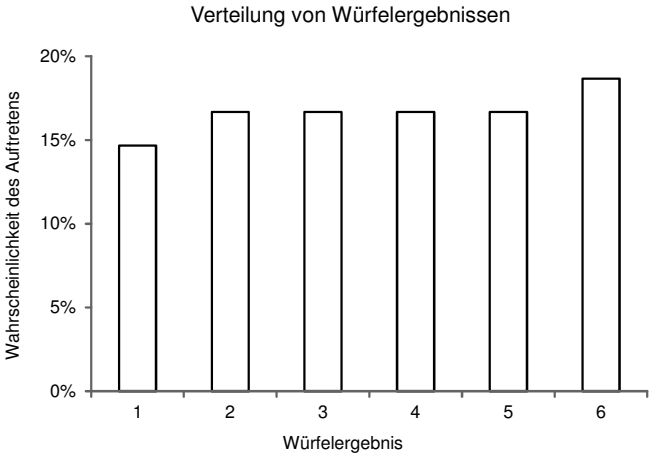
Das erste Unterscheidungskriterium für maschinell lernende Algorithmen und die korrespondierenden MLA basiert auf der Beobachtung, wie verschiedene Algorithmen im Rahmen ihres Autoadaptionsprozesses eine Rückmeldung über die Qualität der ausgegebenen Strukturen erhalten. Die Eingabedaten, die ein lernender Algorithmus während des Autoadaptionsprozesses erhält, werden in der Informatik als TRAININGSINSTANZEN bezeichnet, um sie von den TESTINSTANZEN abzugrenzen, die gemäß ihrem Namen verwendet werden um ein MLA nach Abschluss des Autoadaptionsprozesses zu testen. Im zweiten Hauptteil wird meist an Stelle von Trainingsinstanzen die Rede von ROHDATEN sein, wenn Daten gemeint sind, die ein Nutzer einem MLA übergibt und die einen Autoadaptionsprozess initiieren sollen, dessen Ergebnis zumindest teilweise offen ist. Demgegenüber wird von TRAININGSDATEN gesprochen, wenn das Ergebnis des Autoadaptionsprozesses im Vorfeld bereits detailliert festgelegt wurde. Diese Begriffsverschiebung motiviert sich zum einen daraus, dass der Begriff der Trainingsinstanzen den Eindruck erweckt, dass die Daten eine Struktur instanziieren und zum anderen wird impliziert, dass Algorithmen trainiert werden können. Ebenso leidet das Verständnis der Diskussion der Unterschiede zwischen lernenden Algorithmen, wenn Autoadaptionsergebnisse pauschal als von den Eingabedaten instanziiert gedacht werden. Die Frage, ob Rohdaten etwas instanziieren, wird im zweiten Hauptteil noch ausführlicher diskutiert, aber schon die Diskussion, ob autoadaptive Algorithmen im menschlichen Sinn lernen, soll hier vermieden werden und das gilt ebenso für die Diskussion, ob Algorithmen trainieren oder trainiert werden. Im zweiten Hauptteil wird aus diesem Grund neben dem Begriff der Trainingsinstanzen auch der präzisere Begriff der Trainingsdaten soweit wie möglich vermieden. Im Abschnitt zur Klassifikation von Lernstrategien wird der Begriff der Trainingsinstanzen allerdings dennoch Verwendung finden. Die reflektierte Rede von Trainingsinstanzen hilft in diesem Fall dabei, die Perspektive der Informatik auf das maschinelle Lernen präziser darstellen zu können, denn bei der Klassifikation von Lernstrategien wird es weniger um formale Kriterien auf der Betrachtungsebene der Algorithmen und mehr um

Ideen und Konzepte auf der Betrachtungsebene von Pseudocode gehen. Der Trennung zwischen Trainingsdaten und Rohdaten liegt somit die Unterscheidung zwischen unterschiedlichen Formen der Rückmeldung auf Ausgabedaten und damit eines der wesentlichsten Kriterien zur Differenzierung von maschinell lernenden Algorithmen zugrunde. Die unterschiedlichen denkbaren Formen der Rückmeldung lassen die Differenzierung von drei Varianten maschinellen Lernens zu. Diese drei Varianten werden als überwachtes Lernen, unüberwachtes Lernen und bestärkendes Lernen bezeichnet.

In fast allen Fällen nimmt ein MLA Eingabedaten auf und gibt Ausgabedaten aus – nur in Sonderfällen besitzt ein MLA keine formale Ausgabe, so dass die sich verändernde Struktur des MLA selbst analysiert werden muss beziehungsweise kann. In Konsequenz möchte ein Nutzer oder Entwickler ein MLA häufig dazu bringen, dass der Eingabe-Ausgabe-Zusammenhang eine bestimmte Funktion abbildet. Zu diesem Zweck kann er Eingabedaten zur Verfügung stellen, bei denen die korrekte beziehungsweise gewünschte Ausgabe bekannt ist und mit angegeben werden kann. In diesem Fall spricht man von ÜBERWACHTEM LERNEN. Bei überwachtem Lernen erhält das MLA während des Autoadaptionsprozesses Paare von Ein- und Ausgabewerten und das Ziel liegt darin eine Struktur zu erzeugen, die nach Abschluss des Lernprozesses auf Erhalt des Eingabedaten-Anteils eines Trainingsdatums den Ausgabedaten-Anteil ebenjenes Datenpunktes zurückgibt. Zur Veranschaulichung soll eine automatische Zahlenerkennung betrachtet werden. Es soll angenommen werden, dass dazu im Vorfeld eine Reihe von Bildern von Zahlen digital erfasst und vom Nutzer mit den entsprechenden numerischen Werten versehen wurde. Das MLA bekommt während des Autoadaptionsprozesses als Eingabe Datenpaare, deren Teile die Eingabe und die Ausgabe für das zu erstellende Lernergebnis darstellen. Ein Beispiel wäre die Übergabe des ersten Zahlenbildes zusammen mit der Aussage, dass dort eine Acht zu sehen ist. Das MLA könnte anschließend die Regel ›wenn 20 bis 21 Prozent des Bildes schwarz sind, ist die Ausgabe eine acht‹ erstellen. Wenn die Bilder der zu kategorisierenden Zahlen alle auf digitalen Darstellungen derselben Schriftart basieren, kann das MLA zehn Regeln dieser Art erstellen und hat die ZIELFUNKTION erlernt. Überwachtes Lernen wird häufig eingesetzt, wenn über die Struktur der Rohdaten schon Vorwissen besteht oder das Vorliegen gewisser Strukturen

zumindest vermutet wird. Wenn etwa die Rohdaten aus Messwerten bestehen, die einem bestimmten Muster entsprechen, wird dieses Muster als die VERTEILUNG der Rohdaten bezeichnet. Die Messwerte bei Beobachtung eines Würfelwurfes können beispielsweise die Werte von eins bis sechs sein und bei symmetrischen Würfeln treten alle Würfelergebnisse mit ungefähr der gleichen Wahrscheinlichkeit auf – die Ergebnisse sind in diesem Fall GLEICHVERTEILT. Nachfolgend zu Veranschaulichung die Verteilung der Messwerte für einen asymmetrischen Würfel, der dies nicht ganz erfüllt und häufiger die Sechs und dafür weniger häufig die Eins zeigt.

Abbildung 3: Beispielverteilung von Würfelergebnissen



Mit Hilfe eines MLA kann versucht werden, die Details der Verteilung der Würfelergebnisse eines speziellen, asymmetrischen Würfels zu bestimmen. In diesem Fall würde angenommen werden, dass die Würfelergebnisse unabhängig voneinander und zumindest ungefähr gleichverteilt sind. Wenn weiter über die Verteilung der Rohdaten im Vorfeld bekannt ist, dass jedes Würfelergebnis mit einer unbekannten aber festen Wahrscheinlichkeit auftritt und dass nur sechs Ergebnisse möglich sind, bedeutet das, ein MLA muss sechs PARAMETER bestimmen[5], um die Details der Verteilung der

5 Zu erlernen sind eigentlich nur fünf Parameter, da die Wahrscheinlichkeit für das sechste Ergebnis berechenbar ist, wenn die ersten fünf Wahrscheinlichkeiten bekannt sind.

Würfelergebnisse aufzuzeigen. In der Anwendungspraxis maschinellen Lernens wird sehr häufig im Vorfeld ein Modell erstellt, um darauf aufbauend die für den Autoadaptionsprozess relevanten Parameter zu bestimmen. Ein MLA erlernt anschließend diejenigen Werte für diese Parameter mit Hilfe derer das Modell den Rohdaten am besten entspricht. Überwachtes Lernen bietet sich besonders dann an, wenn ausreichend Messwerte vorliegen und in erster Linie die Parameter des ausgewählten Modells optimiert werden sollen. Im Falle des Würfelwurfes ist die zugrunde liegende Struktur sehr einfach und ein MLA würde nur wenig Mehrwert bieten, da die Auftretenswahrscheinlichkeiten und damit die Parameter der Verteilung auch unkompliziert vom Nutzer selbst auf Basis der Würfelergebnisse errechnet werden könnten.

Das überwachte Lernen setzt die Verfügbarkeit von bekannten Trainingsdaten im Sinne von Eingabe-Ausgabe-Paaren voraus. Sollten solche Trainingsdaten nicht zur Verfügung stehen, kann UNÜBERWACHTES LERNEN zum Einsatz kommen. Hier sind nur Eingabewerte in Form von Rohdaten gegeben und es sind Regeln und Modelle gesucht, nach denen die Eingabedaten strukturiert werden können. Ein Beispiel für die Erstellung solcher Modelle ist die CLUSTERANALYSE. Hierbei sollen noch unbekannte Rohdaten in Gruppen von ähnlichen Daten eingeteilt werden. Dies sollte nicht verwechselt werden mit der KLASSIFIZIERUNG von Daten, bei der die Klassen bereits im Vorfeld festgelegt wurden und die Rohdaten den Klassen zugeordnet werden sollen. Die Clusteranalyse erzeugt CLUSTER genannte Klassen von Rohdaten nach einem vorgegebenen oder erlernten ÄHNLICHKEITSMAßSTAB. Ein verwandtes Beispiel für unüberwachtes Lernen ist die Suche nach ASSOZIATIONSREGELN in Rohdaten, das heißt nach Aussagen, die für große Teile der Rohdaten zutreffend sind. Ein Beispiel für eine Assoziationsregel zu einer fehlenden Assoziation wäre die Aussage, dass die Wahrscheinlichkeit eine Zwei zu würfeln nicht davon abhängt, ob im vorherigen Versuch eine Eins oder eine Vier gewürfelt wurde. Ein anderes Beispiel ist die Suche nach möglichen Kaufempfehlungen, die sich darauf beziehen, dass bestimmte Produkte häufig zusammen gekauft werden.

Eine häufig eingesetzte und illustrative Mischform maschinellen Lernens verbindet menschliches Vorwissen, unüberwachtes und überwachtes Lernen. Diese Mischform beginnt damit, dass vom Nutzer zu einer Menge von Rohdaten mehrere Modelle bestimmt werden, die jeweils Teile der Struktur der Rohdaten abbilden. Anschließend werden die Rohdaten un-

überwacht durch ein MLA in Cluster unterteilt und es werden für jedes Cluster überwacht unterschiedliche Modelle parametrisiert, um optimale LOKALE MODELLE zu bestimmen. In diesem Fall müssen nutzerseitig für jedes Cluster Trainingsdaten beschafft werden, typischerweise über die Durchführung von Messungen oder über eine komplexe, das heißt zeitraubende mathematische Berechnung. Diese Mischform maschinellen Lernens kann dazu dienen, die Anzahl kostspieliger Messungen oder den Zeitaufwand der notwendigen Berechnungsprozesse zu reduzieren, indem nur lokale Aussagen angestrebt werden und kein einheitliches Modell für alle Rohdaten gesucht wird. Die nach Abschluss des Autoadaptionsprozesses entstandene Struktur kann beim Auftreten neuer Rohdaten typischerweise sehr schnell ausgewertet werden. Ein Beispiel ist die Suche nach einem optimalen Fahrverhalten für einen Autopiloten für Automobile. Hierbei kann das MLA zuerst die Strecke in Geraden und verschiedene Arten von Kurven unterteilen und anschließend für die Teilstücke ein optimales Beschleunigungs- und Lenkverhalten erlernen – etwa indem auf einem ebenen und großen Asphaltstück nur Kurven gefahren werden.

Die dritte Variante neben überwachtem und unüberwachtem Lernen ist das BESTÄRKENDE LERNEN. In diesem Fall führen MLA SEQUENZEN von Aktionen durch, deren Einzelschritte jeweils keine Bewertung durch den Nutzer herbeiführen und deren Länge variabel ist. Nur an ausgewählten Zwischenschritten beziehungsweise Punkten der Sequenz und nach Abschluss der Sequenz erhält das MLA eine Rückmeldung auf seine Ausgabe. Diese Vorgehensweise ist an einem Beispiel schnell verständlich. Angenommen, ein MLA soll erlernen Schach zu spielen, dann ist meist nicht bekannt, ob ein spezieller Zug besser oder schlechter ist als ein anderer möglicher Zug. Gleichzeitig können recht einfach die Bedingungen festgelegt werden, unter denen eine Zugfolge des MLA mit einem Sieg oder einer Niederlage beendet ist. Eine solche Situation erfüllt genau die beschriebenen Voraussetzungen für bestärkendes Lernen. Bestärkendes Lernen eignet sich dementsprechend gut für dynamische Umgebungen, da nur das Ziel des Autoadaptionsvorgangs vorgegeben wird. Die Durchführung bestärkenden Lernens setzt häufig auf BRUTE-FORCE. Diese Vorgehensweise simuliert schlicht alle oder doch möglichst viele der möglichen Sequenzen von Aktionen, um dann diejenige Sequenz mit der besten Gesamtbewertung auszuwählen. Diese Methode stößt jedoch schon beim Schachspiel an ihre

Grenzen, da mehr als 10^{40} Stellungen[6] möglich sind (Shannon 1949). Eine weniger aufwendige Variante der Brute-Force-Methode betrachtet eine große, aber beschränkte Anzahl von Sequenzen und schätzt deren Bewertung ab, um anschließend aus diesen Sequenzen eine neue Sequenz zu erstellen, die die Bewertung optimiert. Bestärkendes Lernen mit einer Sequenz der Länge eins schließlich entspricht formal genau dem überwachten Lernen. Bestärkendes Lernen kann auch in einigen weiteren Fällen als eine abgeschwächte Form des überwachten Lernens betrachtet werden, insbesondere bei kurzen Sequenzen mit vorgegebener Länge. Allerdings ist schon am Beispiel des Schachspiels ersichtlich, dass zwischen Anfang und Ende der Sequenz eine sehr große Vielzahl von Zügen möglich ist und dass dies nicht sehr gut der Idee des überwachten Lernen, ein Modell zu parametrisieren, entspricht.

2.2.2 Unterscheidung nach Suchstrategien

Eine andere Möglichkeit die Algorithmen des maschinellen Lernens zu klassifizieren, besteht darin, die eingesetzte Suchstrategie zu betrachten. Diese Betrachtung ist in gewisser Hinsicht verwandt mit der vorherigen Klassifizierung, da für die Analyse der Suchstrategien der Fokus unter anderem darauf gelegt wird, was gesucht werden kann und damit primär auf die Ausgaben des MLA – im Gegensatz zur obigen Fokussierung auf die Eingabedaten. Die Menge aller möglichen Ausgaben eines MLA bildet deren LÖSUNGS-, SUCH-, oder HYPOTHESENRAUM. Die drei Formulierungen werden in der Informatik meist synonym verwendet, können im Einzelfall jedoch auch betonen, dass das MLA ein klar umrissenes Problem lösen, einen Raum möglicher Ausgaben durchsuchen oder mögliche Konzepte und Modelle zu Rohdaten vergleichen oder suchen soll. Der Begriff des Hypothesenraumes scheint die Implikationen eines die Hypothese formulierenden Agenten mitzuführen. Das MLA wird in diesen Fällen jedoch nicht als konzeptbildend gedacht. Stattdessen werden sehr starke Vorstrukturierungen vorgenommen, die Konzepte abbilden und das Artefakt vergleicht oder optimiert diese Konzepte auf Basis von Eingabedaten. Weiterhin wird von

6 Die Anzahl der Atome im Körper von 10 Milliarden Menschen kann auf 10 zur 34. Potenz geschätzt werden (Bauer et al. 1999). Die Anzahl der möglichen Stellungen ist entsprechend unvorstellbar groß.

einem Hypothesenraum aus Sicht vieler Entwickler oder Nutzer genau dann gesprochen, wenn systematische Fehler von Algorithmen aufgedeckt werden sollen. Die Ausgabe eines MLA – sei es ein Element des Such-, Lösungs- oder Hypothesenraumes – wird im Weiteren, wie in Abschnitt 1.2 beschrieben, als STRUKTURVORSCHLAG bezeichnet. Einen Strukturvorschlag kann schon die Unterteilung einer Speisekarte in vegetarische und nicht-vegetarische Gerichte darstellen und die Ergebnisse der Autoadaptionsprozesse vieler MLA bestehen auch genau in solchen Strukturen geringer Komplexität[7]. Ein Beispiel aus der Praxis besteht darin, die Frage was Paris zu Paris macht, zu beantworten, indem eine automatische Zusammenstellung der das Stadtbild prägendsten visuellen Elemente erstellt wird (Doersch et al. 2012). Im Beispiel der Schach erlernenden MLA würde sich der Lösungsraum aus der Menge aller denkbaren Sequenzen zusammensetzen. Ein Entwickler könnte in diesem Fall den Lösungsraum beschränken, indem nur ein gewisser Fundus an Eröffnungen gespielt werden darf und indem für das Endspiel eine Datenbank hinzugezogen werden muss, die eine Vorgehensweise vorschreibt. Solche Vorgaben werden sehr häufig gemacht um den Autoadaptionsprozess schneller zu einem erfolgreichen Ende zu bringen.

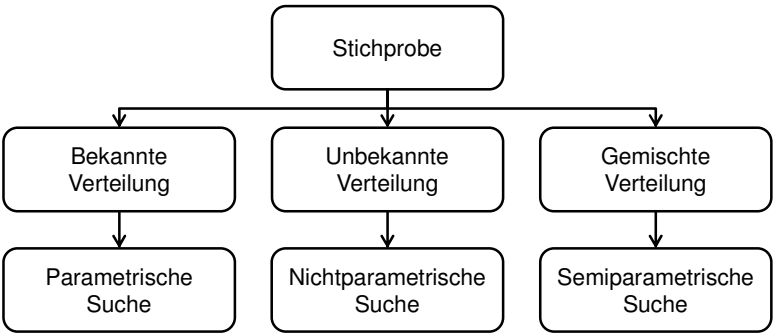
Wenn der Suchraum eines MLA nun aus Elementen beziehungsweise Strukturen besteht, die sich sinnvoll anordnen lassen, kann eine GEORDNETE SUCHE durchgeführt werden. Wenn etwa Passwörter gefunden werden sollen, kann dies mittels maschinellen Lernens versucht werden. Der Ausgaberaum wäre in diesem Fall alphanumerisch sortierbar und der Strukturvorschlag könnte nach Eingabe einiger Informationen über den Passwortinhaber die Kennwörter aller ›ähnlichen‹ Nutzer nach der Häufigkeit von deren Auftreten angeordnet ausgeben. In diesem Fall wäre die Aktualisierung der Ausgabereihenfolge auf Basis des Erfolges der Ausgaben ein möglicher Autoadaptionsprozess.

Eine alternative Suchstrategie besteht darin, einzelne Parameter, die die Entstehung von Strukturvorschlägen oder Lösungen beeinflussen, zu manipulieren, die entstehende Lösung zu bewerten und die Bewertungen zu vergleichen. Anschließend werden die betrachteten Parameter so angepasst,

7 Woraus nicht gefolgert werden sollte, dass der Weg hin zu Strukturvorschlägen geringer Komplexität ebenfalls trivial ist. Ein gut verständliches und sofort nützliches Lernergebnis muss nicht bereits im Vorfeld offensichtlich gewesen sein.

dass die resultierende Lösung in einer gewissen Hinsicht optimal ist. Ein abstraktes, aber anschauliches Beispiel hierfür ist der Versuch, in Modellrechnungen die Verkehrssicherheit von Pkw zu betrachten und dabei als Parameter die Bremskraft und die Genauigkeit der Tankanzeige zu betrachten. In diesem Beispiel wird sich voraussichtlich zunächst herausstellen, dass eine Manipulation des Parameters der Genauigkeit der Tankanzeige die Verkehrssicherheit nicht signifikant beeinflusst und somit verworfen werden kann. Anschließend wird festgestellt werden, dass die Bremskraft tatsächlich einen systematischen Einfluss auf die Sicherheit hat und ein höherer Wert dieses Parameters sehr häufig vorteilhaft ist. Das bedeutet, der resultierende Vorschlag wäre, den Parameter der Bremskraft zu erhöhen. Diese Form der Suche wird als GRADIENTENSUCHE bezeichnet und identifiziert die optimale Veränderung gegebener Parameter. Eine Gradientensuche, bei der wie im genannten Beispiel nur ein Parameter verändert werden kann oder soll, lässt sich meist auch als eine geordnete Suche darstellen. Die Suche im Beispiel etwa ordnet Pkw systematisch nach deren Bremskraft an und prüft anschließend im Rahmen des Autoadaptionsprozesses die Auswirkung einer veränderten Bremskraft auf die Verkehrssicherheit. Die Suchergebnisse werden in diesem Fall als durch die Größe des beeinflussbaren Parameters geordnet gedacht. Der Fokus einer Gradientensuche und einer geordneten Suche ist dennoch unterschiedlich, da im ersten Fall mathematisch die optimalen Parameter zur Erstellung einer Lösung gesucht werden und im zweiten Fall die Reihenfolge festgelegt wird, in der mögliche Lösungen betrachtetet werden sollen. Die dritte wesentliche Ausprägung neben der geordneten Suche und der Gradientensuche ist die STOCHASTISCHE SUCHE. Bei der stochastischen Suche soll eine VERTEILUNG gefunden werden, die optimal eine Menge von Rohdaten modelliert. Die Rohdaten werden als eine STICHPROBE für die zugrunde liegende Verteilung interpretiert und gemäß der folgenden Illustration wird eine Suchstrategie ausgewählt.

Abbildung 4: Überblick der Möglichkeiten stochastischer Suche



Parametrische Suchen und nichtparametrische Suchen wurden bereits in der Klassifizierung von Algorithmen über deren Eingabedaten kurz dargestellt. SEMIPARAMETRISCHE SUCHEN stellen verschiedene Mischformen dar, etwa wenn lokale Modelle erstellt werden, die zum Teil mit parametrischen und zum Teil mit nichtparametrischen Ansätzen bearbeitet werden.

2.2.3 Unterscheidung nach Verwendungszweck

Mitunter wird das maschinelle Lernens auch auf Basis des Verwendungszwecks in Teilbereiche unterteilt. Ein Beispiel ist die Unterscheidung zwischen den Typen des parametrischen, semiparametrischen und nichtparametrischen Lernens. Diese Unterscheidung verortet geordnete Suchen und Gradientensuchen je nach Einzelfall unter einem der drei Typen. Meist wird dabei davon ausgegangen, dass nichtparametrisches Lernen in erster Linie dann eingesetzt werden kann, wenn ein starker und stabiler Zusammenhang zwischen Eingaben und Ausgaben besteht und Änderungen dieses Zusammenhangs nur langsam auftreten. Solch ein Zusammenhang wird bei Gradientenverfahren benötigt, da diese das Vorliegen eines direkten Zusammenhangs der Parameter mit der Qualität der Ausgabe voraussetzen.

Eine zweite Unterscheidung nach Verwendungszweck teilt den Bereich des Data Mining in die Teilbereiche der Klassifikation, der Clusteranalyse und der Suche nach Assoziationsregeln ein. Klassifikation ist dabei, wie bereits angedeutet, gedacht als die Vorhersage der Eigenschaften von Rohdaten aus bereits eingeordneten Daten, die Clusteranalyse als die Einteilung von Rohdaten in Klassen ähnlicher Daten und die Suche nach Assoziations-

regeln als das Auffinden von Zusammenhängen zwischen häufig vorkommenden Daten.

2.2.4 Bewertung der Unterscheidungsmöglichkeiten

Die Betrachtung der oben genannten, bunt gemischten Ansätze der Informatik zur Strukturierung des maschinellen Lernens ist hilfreich, sowohl um die Grundbegriffe des maschinellen Lernens kennen zu lernen als auch um die verschiedenen und teilweise widerstrebenden Absichten der Informatik aufzuzeigen und damit ein interdisziplinäres Verständnis der Problematik zu entwickeln. Nichts desto trotz sind die dargestellten Perspektiven der Informatik zur Unterscheidung von Algorithmen aus technikphilosophischer Sicht mangels Systematik nicht befriedigend. Auch die Kombination der Kriterien ist auf dem Weg zu einem systematischeren Überblick über das maschinelle Lernen nicht hilfreich. Die Unterscheidungen nach Rückmeldungen und nach Suchstrategien sind zwar weitgehend unabhängig voneinander, ein Kreuzvergleich der beiden Unterscheidungsarten ergibt aber keine neuen Erkenntnisse. Zwar findet sich der typischere Einsatz einer parametrischen Suche im überwachten Lernen, da dort meist generell mehr Vorwissen besteht, aber auch im unüberwachten Lernen ist eine parametrische Suche denkbar. In die andere Richtung gedacht, kann bestärkendes Lernen sowohl bei einem geordneten Suchraum als auch bei einer stochastischen Suche eingesetzt werden. Gerade die Unterscheidung nach der Suchstrategie ist darüber hinaus kategorial nicht ganz einheitlich, da etwa die nichtparametrische Suche ein Bereich ist, der je nach Lesart des Begriffes sehr viel mehr enthält als nur stochastische Suchen.

Prinzipiell überrascht eine solche Gemengelage nicht, da die Definition von maschinellem Lernen sehr allgemein gehalten ist, sich also sehr viele Ansätze und Entwicklungsziele unter der Überschrift vereinen lassen und die Informatik als Disziplin nur ein eingeschränktes Interesse daran hat, den Bereich als Ganzen systematisch diskutieren zu können. Die genannten Unterscheidungen sind von Anwendungsgebieten und Rahmenbedingungen des Einsatzes von MLA motiviert und sind von den Spezifika des maschinellen Lernens erst einmal unabhängig. Zum Teil werden die Algorithmen nur noch über ihre Ergebnisse klassifiziert beziehungsweise typisiert und damit benennbar gemacht. Das ist insbesondere im maschinellen Lernen problematisch, da hier das Lernergebnis durchaus sinnvoll als Black Box

betrachtet werden kann und häufig keinen Rückschluss mehr auf den Autoadaptionsprozess zulässt. Andererseits hat sich überraschenderweise in der Informatik neben oder vor der genannten Unterscheidung des Gesamtbereiches in Teilbereiche eine andere Art durchgesetzt, maschinell lernende Algorithmen zu verorten. Einzelne Algorithmen werden nach der zugrunde liegenden Idee zu Klassen zusammengefasst und das allgemeine Funktionsprinzip hinter dem Algorithmus spielt dabei eine größere Rolle als fachliche Details. Diese Klassen sind die der LERNSTRATEGIEN.

2.3 KLASSIFIZIERUNG NACH LERNSTRATEGIEN

Die im Folgenden vorgenommene Klassifizierung von Gruppen lernender Algorithmen als Lernstrategien basiert auf dem Ansatz, maschinelles Lernen nicht vom Lernergebnis aus zu denken. Die Algorithmen sollen gerade nicht als Black Box betrachtet werden, die nach Ablauf eines opaken Autoadaptionsprozesses in der Lage sind Rohdaten Ausgabestrukturen zuzuordnen. Gleichzeitig soll die Klassifizierung nicht zu sehr auf die Details der konkreten Implementierung von MLA eingehen, denn formal entstehen im maschinellen Lernen ständig neue Algorithmen. Im Rahmen des Autoadaptionsprozesses verändert schon die Aufnahme der Eingabe- oder Sensordaten formal die Struktur des zugrunde liegenden Algorithmus. Die Klassifizierung wird stattdessen vorgenommen, indem Algorithmen der gleichen Lernstrategie zugeordnet werden, wenn die Selbstorganisationsprinzipien, die hinter den jeweiligen Autoadaptionsprozessen stehen, sich ähneln. Diese vom Algorithmus gedachte und über die Strategie argumentierende Klassifizierung wird auch innerhalb der Informatik in ähnlicher Weise vorgenommen. Die Lernstrategien sind meist gut voneinander abzugrenzen, allerdings gibt es Ausnahmen, beziehungsweise Grenzfälle, die im Einzelnen diskutiert werden, soweit sie aus interdisziplinärer Perspektive einen Mehrwert bieten. Insgesamt kann die Methodik per Konstruktion alle denkbaren Algorithmen abdecken und kann insbesondere auch mit Mischformen und neu entstehenden Strategien ohne größere Schwierigkeiten umgehen.

Die im maschinellen Lernen erzeugten Ausgabestrukturen sollen zwar nicht als Basis für die Klassifikation genutzt werden, spielen aber dennoch eine zentrale Rolle. Zwar können die erzeugten Ausgabestrukturen formal häufig als RECOMMENDER-SYSTEME und damit als Entscheidungsalgorithmen beschrieben werden, aber eine solche Interpretation erfolgt nicht

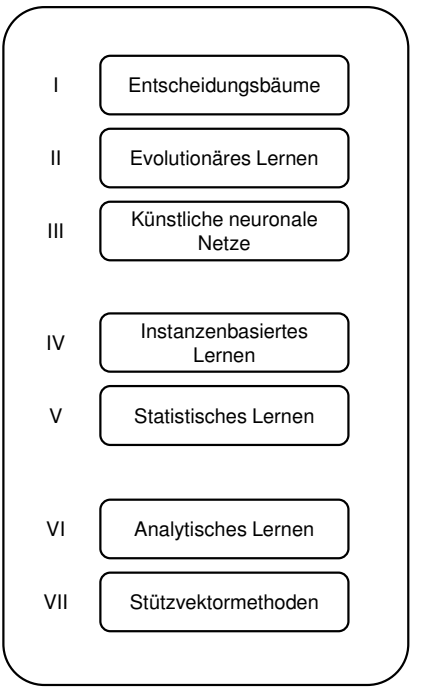
zwangsläufig und ist in den meisten Fällen nicht hilfreich. Die Ausgabestrukturen von MLA werden im Weiteren stattdessen als Strukturvorschläge betrachtet und bezeichnet. Unabhängig davon, inwiefern ein MLA einen Algorithmus erstellt oder nicht, erstellt das MLA formal nie eine wiederum autoadaptive Struktur, da ein Strukturvorschlag formal erst dann vorliegt, wenn der Autoadaptionsprozess beendet oder eingefroren wurde.

Zusammengefasst denkt die Klassifizierung von Lernstrategien das maschinelle Lernen vom Algorithmus aus und interessiert sich in erster Linie für den Autoadaptionsprozess und nur nachrangig für den resultierenden Strukturvorschlag.

2.3.1 Überblick der Lernstrategien

Die Darstellung der in Abbildung fünf skizzierten Klassifizierung maschinell lernender Algorithmen nach Lernstrategien stellt das zentrale Element des ersten Hauptteils dar.

Abbildung 5: Lernstrategien im maschinellen Lernen



Die Reihenfolge, in der die Lernstrategien dargestellt werden, begründet sich dabei wie folgt. Die erste dargestellte Klasse von lernenden Algorithmen sind die ENTSCHEIDUNGSBÄUME. Das Prinzip der hier verorteten lernenden Algorithmen ist einfach zu verstehen und den meisten Menschen aus anderen Kontexten als dem maschinellen Lernen bereits bekannt. Die Entscheidungsbäume eignen sich entsprechend gut als Einstiegsstrategie. Die zweite dargestellte Klasse ist das EVOLUTIONÄRE LERNEN. Das evolutionäre Lernen ist eine Zusammenfassung von drei stark verwandten Lernstrategien, deren Diskussion zwar nicht unproblematisch ist, die aber alle drei gut beschreibbar sind, da nur wenige formale Formulierungen und Erklärungen notwendig sind. Die dritte dargestellte Klasse sind die KÜNSTLICHEN NEURONALEN NETZE. Die hier betrachteten autoadaptiven Algorithmen lassen sich aus interdisziplinärer Perspektive ähnlich gut beschreiben wie evolutionäres Lernen – allerdings sind zur korrekten Darstellung künstlicher neuronaler Netze deutlich mehr formale Details notwendig. Die genannten drei Klassen erfordern zwar mitunter den Umgang mit formalen Konzepten der Informatik, allerdings wird der Zugang zu diesen Konzepten durch hilfreiche Intuitionen zu Bezeichnungen wie ›evolutionäres Lernen‹ erleichtert.

Die vierte und fünfte Klasse können zwar nicht auf eine entsprechende Intuition verweisen, nutzen aber eingängige und zum Teil allgemein bekannte mathematische Konzepte, die im Rahmen der Diskussion der Unterschiede zwischen Lernstrategien bereits sehr kompakt eingeführt wurden. Die vierte Klasse bildet das INSTANZENBASIERTE LERNEN. Das instanzenbasierte Lernen basiert stark auf der bereits kurz beleuchteten Clusteranalyse. Die fünfte Klasse des STATISTISCHEN LERNENS ist relativ eng mit dem instanzenbasierten Lernen verbunden und vereint wie schon das evolutionäre Lernen drei Lernstrategien, die stark verwandt sind.

Die sechste und siebte Klasse werden im Folgenden nur kurz skizziert. Einerseits sollte zu diesem Zeitpunkt schon ein recht gutes Verständnis für das maschinelle Lernen als Gebiet entstanden sein und andererseits spielen sie aus interdisziplinärer Sicht zunächst eine nachgeordnete Rolle[8]. Die Klasse des ANALYTISCHEN LERNENS basiert wesentlich auf der Idee, direkt logische Aussagen zu manipulieren und einen geordneten Suchraum sol-

8 Insbesondere spielen beide in der Diskussion des zweiten Hauptteils keine große Rolle.

cher Aussagen zu betrachten. Zwar bildet dieses Vorgehen einen interessanten Ansatz für die Realisierung maschinellen Lernens, die Lernstrategie spielt allerdings in der Praxis keine große Rolle. Die Klasse der STÜTZVEKTORMETHODEN schließlich stellt ein Beispiel für den oben genannten Fall dar, dass neu entstandene Algorithmen, die auf bereits bekannten Lernstrategien basieren, eine neue Lernstrategie entstehen lassen. Im Fall der Stützvektormethoden waren die Algorithmen mathematisch nicht neu, aber die systematische Umsetzung im maschinellen Lernen war es. Die Stützvektormethoden werden darüber hinaus ein besonders gutes Beispiel für Algorithmen darstellen, die im zweiten Hauptteil nicht von weiterem Interesse sind und können somit zur Abgrenzung genutzt werden.

Im praktischen Einsatz und bei der Erstellung von MLA wird zwar eine Vielzahl von Hybriden der oben beschriebenen Strategien eingesetzt, die These ist jedoch, dass eine Kombination verschiedener Lernansätze und Weiterentwicklungen in den allermeisten Fällen keine prinzipiell neuartigen Verhaltensweisen von und Interaktionsformen mit lernenden Maschinen entstehen lässt, die nicht nur eine Kombination beziehungsweise Überlagerung der im Weiteren beschriebenen Verhaltensweisen darstellt. Tritt doch der Fall auf, dass unerwartete Effekte bei der Konstruktion von Mischformen beobachtet werden, würde dies zu der Entstehung einer neuen Lernstrategie führen, die auf der Erzeugung beziehungsweise Nutzung des entsprechenden Phänomen beruhen würde. Unabhängig von neu entstehenden Strategien, decken die genannten Lernstrategien das aktuelle vorliegende maschinelle Lernen zu sehr großen Teilen ab. Weitere Lernstrategien werden zwar ständig entwickelt und verworfen, aber das Ziel des ersten Hauptteils liegt darin, ein interdisziplinäres Verständnis der etablierten Technik des maschinellen Lernens zu vermitteln. Hierfür ist ein gutes Verständnis der genannten Lernstrategien mehr als ausreichend.

Die nachfolgenden Darstellungen sollen auch dazu dienen, bei der Arbeit an technikphilosophischen Fragen schnell auf die hier geleistete Grundlagenarbeit zugreifen zu können. Aus diesem Grund und zur besseren Lesbarkeit werden mitunter bereits eingeführte Begriffe noch einmal kurz erläutert.

2.3.2 Lernen von Entscheidungsbäumen

Funktionsbeschreibung

Ein ENTSCHEIDUNGSBAUM ist eine Struktur, die EINGABEDATEN aufnimmt, die durch eine Menge von ATTRIBUTEN beziehungsweise Attributwerten vollständig beschreibbar sein müssen. Ein Entscheidungsbaum prüft für jede Eingabe eine Anzahl von Kriterien und führt anhand der Prüfungsergebnisse eine KLASSIFIZIERUNG der übergebenen Struktur durch. Diese Klassifizierung wird ausgegeben. Die gemeinsame Visualisierung aller Prüfungskriterien wird Entscheidungsbaum genannt. Entsprechend handelt es sich bei einem Entscheidungsbaum um einen KLASSIFIKATOR, bei dem die möglichen Klassen typischerweise vorgegeben sind und die Aufgabe nur darin besteht Prognosen abzugeben, welcher Klasse ein gegebenes Eingabedatum zugeordnet werden soll.

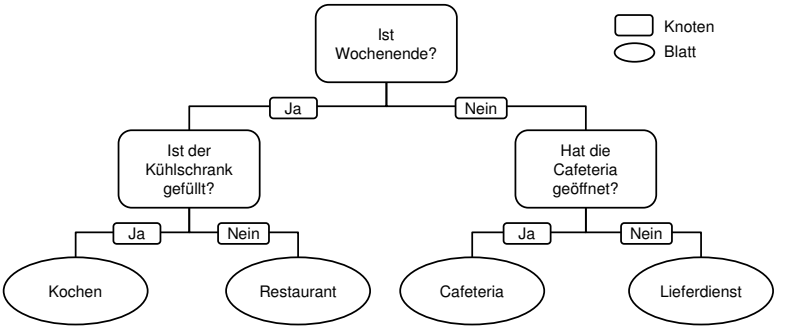
Die Betrachtung von Entscheidungsbäumen als Klasse von Algorithmen und damit als Lernstrategie bezieht sich auf die Konzeption eines Autoadaptionsprozesses, der mit dem Ziel gestartet wird, die Fähigkeit zur Klassifikation von Eingabedaten zu ermöglichen. Wenn der Entscheidungsbaum als Modell vorliegt, wird der Autoadaptionsprozess als abgeschlossen betrachtet. Wenn geplant ist, den Entscheidungsbaum möglicherweise zukünftig noch einmal zu adaptieren, wird der Lernvorgang als EINGEFRORENEN bezeichnet. Nicht der Einsatz eines Entscheidungsbaumes zur Klassifizierung von Eingabedaten, sondern die Modellbildung, die einem Entscheidungsbaum vorangeht, ist der Vorgang, der als maschinelles Lernen betrachtet wird. Die fertigen Entscheidungsbäume stellen dementsprechend das Ergebnis des Einsatzes eines auf einer gewissen Lernstrategie basierenden lernenden Algorithmus beziehungsweise MLA dar.

Beispiel für einen Entscheidungsbaum

Ein Entscheidungsbaum in seiner finalen Form kann ohne größere Schwierigkeiten in einer für den Nutzer direkt lesbaren Form dargestellt werden. Derartige Strukturvorschläge werden als SYMBOLISCHE SYSTEME bezeichnet. Ein SUBSYMBOLISCHES SYSTEM hingegen ist ein System oder Modell, dessen Funktion von einem menschlichen Betrachter nicht ohne größere

Schwierigkeiten erkannt werden kann[9]. Die folgende Illustration zeigt ein Beispiel für einen Entscheidungsbaum, der die Frage beantwortet, wo zu Mittag gegessen werden kann.

Abbildung 6: Entscheidungsbaum zur Wahl des Mittagessens



Die betrachteten Attribute der Eingabedaten sind hier der Wochentag, die Nutzbarkeit der Cafeteria und die Befüllung des Kühlschrankes. Die Eingabedaten können noch weit mehr Attribute aufweisen, die Suche der relevantesten Attribute ist fast immer Teil der Problemstellung. Die auf die Attribute bezogenen Prüfungskriterien, die der Entscheidungsbaum untersucht, sind auf KNOTEN festgehalten und die möglichen Prüfungsergebnisse entsprechen KANTEN, das heißt Verbindungen von Knoten zu anderen Knoten oder zu BLÄTTERN. Blätter stellen ›Endpunkte‹ des Entscheidungsbaums dar, an denen eine Entscheidung getroffen und damit eine Klassifizierung vorgenommen wird. Blätter weisen den auf ihnen noch betrachteten Eingabedaten Klassen zu. Diese Zuweisungen können STATISCH sein, indem wie im obigen Beispiel eine Klasse festgelegt wird, oder DYNAMISCH mittels Funktionen durchgeführt werden. Das Blatt links unten im Beispiel könnte etwa die zu kochende Menge abhängig von den bereits verspeisten Kilokalorien vorgeben. Diese Verwendung von FUNKTIONEN auf Blättern ließe sich im Prinzip auch durch weitere Knoten und Blätter ersetzen, allerdings würde die Übersichtlichkeit des Baumes sehr leiden, wenn tausende Blätter

9 Die Unterteilung in symbolische und subsymbolische Systeme ist eine weitere verbreitete Möglichkeit lernende Algorithmen zu unterscheiden. Jedes symbolische System kann jedoch durch eine Kodierung in ein subsymbolisches System umgeformt werden.

ergänzt würden, deren korrespondierende Klassen sich nur um eine Kilokalorie unterscheiden. Mit diesem Trick können auch Klassifizierungsprobleme bearbeitet werden, die eine Unterscheidung zwischen unendlich vielen Klassen erfordern.

Praktische Anwendungsbeispiele umfassten in der Vergangenheit beispielsweise die Bewertung des Risikos bei der Auswahl von Anwärtern für Kredite oder die Erstellung von medizinischen Diagnosen (Mitchell 1997).

Konstruktion eines Entscheidungsbaumes

Zu einer PROBLEMSTELLUNG in Form einer vorzunehmenden Klassifizierung sei eine Anzahl von TRAININGSDATEN gegeben, deren Klassenzugehörigkeit bereits bekannt ist – das heißt, eine Anzahl von INSTANZEN im eigentlichen Sinn. Die Betrachtung und Prüfung eines oder mehrerer Attribute einer Instanz wird als TEST bezeichnet. Jeder Test entspricht einer Klassifizierung, da die getesteten Daten auf mindestens zwei nachfolgende Knoten aufteilt werden. Ein Entscheidungsbaum kann diese Aufspaltung so intensiv betreiben und so viele Tests vorschreiben, dass jedes Blatt nur genau eine Trainingsinstanz beschreibt – das wird notwendig, wenn jede Instanz jeweils der einzige REPRÄSENTANT einer Klasse ist.

Die Entscheidungskriterien der Tests können STOCHASTISCH sein, das heißt, ein Prüfungsergebnis wird nur mit einer gewissen Wahrscheinlichkeit einem Datum zugeordnet[10]. Tests werden, wie im Beispiel zu sehen, als Knoten visualisiert und der Knoten des ersten durchzuführenden Tests wird als WURZELKNOTEN bezeichnet. Die Erstellung eines Entscheidungsbaumes folgt nun – beginnend mit der Betrachtung des Wurzelknotens – den folgenden Schritten.

A. Falls die am betrachteten Knoten noch verbliebenen Trainingsdaten sich vollständig aus Instanzen einer einzelnen Klasse zusammensetzen, wird der betrachtete Knoten zu einem BLATT und die Schritte **B** und **C** werden ausgelassen.

B. Für alle Tests, die für die Instanzenmenge am betrachteten Knoten durchgeführt werden könnten, wird geprüft, wie gut der Test die Instan-

10 Hierdurch entscheidet der Entscheidungsbaum bei wiederholter Eingabe des gleichen Datums unterschiedlich.

zen trennt. Die Kenngröße wird allgemein als QUALITÄT bezeichnet, kann jedoch unterschiedlich konstruiert sein. Der Test mit der höchsten Qualität wird dem Knoten zugeordnet. Der Knoten teilt damit die betrachteten Eingabedaten entsprechend den Testergebnissen in disjunkte Teilmengen ein.

C. Für jede dieser disjunkten Teilmengen wird ein neuer Knoten erzeugt, der nur die Instanzen der jeweiligen Gruppe testet. Diese neuen Knoten werden als KINDERKNOTEN bezeichnet und über eine Kante mit dem erzeugenden Knoten verbunden.

D. Der Prozess beginnt für einen noch nicht betrachteten Knoten erneut bei Schritt **A**. Wenn keine noch nicht betrachteten Knoten verblieben sind, ist die Erstellung des Baums abgeschlossen.

Sind alle Eingabedaten Instanzen einer einzigen Klasse, so besteht der Entscheidungsbaum nur aus einem Blatt und keinem Knoten. Wenn die Eingabedaten Instanzen aus zwei Klassen enthalten, kann der Entscheidungsbaum mit einem Knoten und zwei Blättern auskommen. Die Qualität des Tests wäre in diesem Fall maximal. Der Zusammenhang des Konzepts von Qualität bei der Erstellung von Entscheidungsbäumen wird später noch betrachtet. In der Praxis ist die Identifikation eines einzigen Tests, der alle Eingabedaten eindeutig in Klassen einordnet, meist nicht möglich und es muss eine Reihe von Tests durchgeführt werden um zwei Klassen von Instanzen voneinander zu TRENNEN. Ein Entscheidungsbaum darf nur endlich viele Knoten enthalten, dies muss unter anderem bei der Wahl des Qualitätskonzeptes für Schritt **B** sichergestellt werden, etwa indem jedes Attribut der Eingabedaten nur genau einmal im Rahmen eines Tests überprüft werden darf. Üblicherweise werden Entscheidungsbäume daher, trotz der Möglichkeit Blätter dynamischer Klassenzuordnung zu verwenden, nur zur Klassifizierung genutzt, wenn eine endliche Anzahl von Trainingsdaten vorliegt und nach endlich vielen Klassen getrennt werden soll.

Der Qualitätsbegriff bei Entscheidungsbäumen

Das Wesentliche an Entscheidungsbäumen ist aus Sicht der Lernstrategien des maschinellen Lernens nicht der Entscheidungsbaum selbst, sondern der Autoadaptionsprozess, der diesen Entscheidungsbaum entstehen lässt. Das wichtigste Konzept bei der Erstellung dieser Entscheidungsbäume wiede-

rum ist das Konzept der QUALITÄT eines Tests. Die Darstellung des Qualitätsbegriffes basiert unter anderem auf dem Verständnis von INFORMATION innerhalb des maschinellen Lernens und erfordert eine gewisse Vorarbeit. Innerhalb des maschinellen Lernens ist fast immer die syntaktische Ebene gemeint, wenn von Informationen die Rede ist.

> »Der Informationsgehalt einer Nachricht entspricht der Anzahl der Ja-/Nein-Fragen, die man bei einer idealen Fragestrategie braucht, um sie zu rekonstruieren.«
>
> (Wikipedia Contributors 2012, Information)

Der Begriff der Information ist im Kontext des maschinellen Lernens zwar klar bestimmt, dient jedoch in erster Linie der Bestimmung des wesentlicheren und innerhalb des maschinellen Lernens ebenso klar bestimmten Begriffs der ENTROPIE. Die Entropie misst die Durchmischung von Trainingsinstanzen, die positive und negative Beispiele einer Klassifizierung darstellen. Wenn alle Trainingsdaten der gleichen Klasse angehören, ist diese Durchmischung beziehungsweise die Entropie minimal. Wenn hingegen genau gleich viele Trainingsdaten den jeweils vorliegenden Klassen angehören, ist die Entropie maximal. Die Reduktion der Entropie in Datenbeständen ist häufig die Motivation zum Einsatz von MLA und das Konzept von Informationen spielt nur insofern eine Rolle, als es dem Informationsgewinn und damit der Messung der Entropie zugrunde liegt. Die Qualität eines Entscheidungsbaumes misst die Verbesserung der Entropie, die mit Hilfe eines speziellen Tests erzielt wird. Diese Verbesserung wird als INFORMATIONSGEWINN bezeichnet und gibt an, wie stark die durch den Test neu entstehenden Teilmengen mit Instanzen aus verschiedenen Klassen durchmischt sind, relativ zur Durchmischung der vor dem Test vorliegenden Menge von Instanzen. Angestrebt wird die Entstehung von Teilmengen, deren Elemente jeweils genau einer Klasse angehören, das heißt die Entstehung von Blättern. Blätter entstehen, wenn eine Durchmischung von null bei einer durch den Test neu entstehenden Teilmenge getesteter Instanzen festgestellt wird. Das Konzept des Informationsgewinns bringt die Gefahr mit sich, dass einzelne Knoten die betrachteten Daten FRAGMENTIEREN, das heißt, im Extremfall sämtlichen betrachteten Instanzen isolierte, individuelle Blätter zuordnen. Eine Weiterentwicklung des Qualitätskonzeptes zum Umgang mit diesem Problem ist das Konzept des GEWINNVER-

HÄLTNISSES. Hier wird der Grad, in dem die Instanzenmenge durch den Test fragmentiert wird, in das Qualitätskonzept integriert. Zusammengefasst stellt die Entropie die Kennzahl zur Messung der Fragmentierung einer Datenmenge dar und die Sicherstellung einer minimalen Entropie in der PARTITIONIERUNG einer Instanzenmenge wird als INFORMATIONSTRENNUNG bezeichnet. Eine andere Weiterentwicklung des Informationsgewinns berücksichtigt die Tatsache, dass gewisse Tests nur unter sehr hohen Kosten durchzuführen sind oder längere Zeit benötigen. Zur Berücksichtigung dieser Umstände kann bei der Bestimmung der Qualität eines Tests eine Strafe für teure Tests aufgenommen werden. Die an dieser Stelle eingeführten technischen Ausdrücke zur Arbeit mit dem Begriff der Information werden im Weiteren keine explizite Verwendung finden. Die Einführung dieser Ausdrücke ist dennoch hilfreich, da sie die Diskussion von Informationstechnik in Abschnitt 3.3 vorbereitet.

Vorteile und Nachteile von Entscheidungsbäumen

Im Folgenden werden die wesentlichsten Stärken und Schwächen von MLA beschrieben, die als Ausgabestruktur Entscheidungsbäume erstellen. Zwar ist der entstehende Entscheidungsbaum als Lernergebnis und damit als Strukturvorschlag des Autoadaptionsprozesses für das Verständnis ebenjenes Prozesses formal nicht von höchster Bedeutung, aber die Grundidee dieser Lernstrategie basiert darauf, dass Entscheidungsbäume manipuliert werden sollen. Entsprechend wird im Weiteren nicht explizit zwischen Vor- und Nachteilen des Autoadaptionsprozesses und des resultierenden Strukturvorschlages unterschieden. Tatsächlich gehen beide Bereiche bei der Entwicklung der konkreten Algorithmen ineinander über, da der Lernprozess, wie oben beschrieben, schrittweise Knoten und Blätter hinzufügt, wodurch er den entstehenden Entscheidungsbaum vergrößert. Das MLA muss somit in jedem Schritt mit der Struktur arbeiten, die auch das Endergebnis darstellt. Das heißt, eine schnelle Reaktionsgeschwindigkeit von Entscheidungsbäumen beschleunigt auch den Lernprozess. Dieser Zusammenhang ist für andere Lernstrategien, insbesondere die künstlichen neuronalen Netze, ebenfalls von großer Bedeutung. Die Rede von einem Autoadaptionsprozess findet hier ihre wesentlichste Motivation. Die iterative Adaption des Entscheidungsbaums ist die Adaption der Struktur, die aus Reizen gelernt hat. Das Verhältnis des Autoadaptionsprozesses und der

veränderten Struktur wird im Rahmen der Diskussion nichttrivialer Maschinen im zweiten Hauptteil weiter ausgeführt.

Gut geeignet für die Konstruktion eines Entscheidungsbaumes sind vor allem Problemstellungen, bei denen eine für die Nutzer schnell verständliche Darstellung der getrennten Klassen bevorzugt wird. Auch erfordert die Klassifizierung eines Eingabedatums nur eine vergleichsweise kurze LAUFZEIT, das heißt Bearbeitungsdauer der Klassifizierung durch das MLA. Die Nutzung von Entscheidungsbäumen in der Praxis ist entsprechend unproblematisch.

Die Entwicklung von Entscheidungsbäumen ist ROBUST gegen RAUSCHEN. Das bedeutet, fehlerhafte Messwerte und damit Trainingsdaten, die nur annähernd korrekte Aussagen machen, behindern den Autoadaptionsprozess nur geringfügig. Entscheidungsbäume können zudem auch dann gelernt werden, wenn Trainingsdaten unvollständig sind, das heißt, wenn Werte von Attributen fehlen, etwa weil Aufzeichnungen fehlen oder zu teuer waren[11].

Ein Hauptnachteil bei der Entwicklung von Entscheidungsbäumen liegt darin, dass der Lösungsraum UNVOLLSTÄNDIG DURCHSUCHT wird, wodurch Fehlklassifikationen entstehen können. Die Bezeichnung als unvollständige Suche bezieht sich darauf, dass durch die einmalige Entscheidung für einen speziellen Test für jeden Knoten und die darauf aufbauende Erweiterung des Modells all diejenigen Entscheidungsbäume nicht betrachtet werden, bei denen der jeweilige Test ein anderer wäre. Ein weiterer wesentlicher Nachteil liegt in der Gefahr einer ÜBERANPASSUNG an die Trainingsinstanzen. Von Überanpassung wird gesprochen, wenn ein MLA die Trainingsdaten zu genau berücksichtigt und beispielsweise Attribute, die keine Rolle spielen, in Tests miteinschließt. Wenn beispielsweise Sehenswürdigkeiten auf Basis von Fotos identifiziert werden sollen und auf allen Trainingsinstanzen, die den Eiffelturm zeigen, eine Wolke am Himmel ist, könnte der Entscheidungsbaum diese Eigenschaften als Attribut abprüfen, bevor er eine Abbildung als den Eiffelturm erkennt.

11 Formal gesprochen sind Entscheidungsbäume in der Lage sehr verschiedene Strukturvorschläge zu lernen. (Präzise: Sie können unter anderem alle Funktionen modellieren, die auf endlichen diskreten Mengen operieren, sind vollständig ausdrucksstark in der Klasse der aussagenlogischen Sprachen (Russell et al. 2007) und können zu Kausalsätzen umgeschrieben werden.)

Abgesehen von den beispielhaft aufgezählten Vor- und Nachteilen weisen Entscheidungsbäume einen prinzipiellen BIAS – eine konstruktionsbedingte systematische Verzerrung – auf. Dieser Bias wird innerhalb der Informatik vielfach als induktive Verzerrung bezeichnet, wenngleich der Begriff fast ausschließlich auf Englisch verwendet wird. Diese Bezeichnung ist jedoch irreführend, gemeint ist eine ABDUKTIVE VERZERRUNG bei der Qualitätsbewertung[12]. Die abduktive Verzerrung besteht in einer Bevorzugung von kleinen gegenüber großen Entscheidungsbäumen und damit kompakten gegenüber umfangreichen Strukturvorschlägen. Insbesondere werden Entscheidungsbäume, die einen hohen Informationsgewinn nahe dem Wurzelknoten aufweisen, bevorzugt. Dies führt systematisch zu weniger Überanpassung, allerdings gegebenenfalls auch zu theoretisch auf Basis der Trainingsinstanzen vermeidbaren Fehlklassifizierungen – Details zu dem Problem der unvollständigen Suche und der abduktiven Verzerrung werden in der Diskussion des Stutzens erläutert.

Stutzen als entscheidungsbaumspezifische Maßnahme gegen Überanpassung

Entscheidungsbäume können im finalen Zustand oder in Zwischenzuständen während des Autoadaptionsprozesses GESTUTZT werden, um eine Überanpassung zu vermeiden. Eine Stutzung von Entscheidungsbäumen wird häufig durchgeführt, indem die Kaskade von einem Knoten über all dessen Kinderknoten bis hin zu den Blättern – wenn man so will ein AST – durch ein Blatt ersetzt wird. Das neu erstellte Blatt kategorisiert Eingabedaten entsprechend der im gestutzten Ast vorrangigen Klassifikation. Häufig wird dabei überprüft und sichergestellt, dass der entstehende, verkleinerte Entscheidungsbaum die Trainingsdaten mindestens genauso präzise klassifiziert wie der ungestutze Entscheidungsbaum. Die Fähigkeit eines MLA zur präzisen Klassifizierung wird als Betrachtung der PERFORMANZ bezeichnet, eine Stutzung soll entsprechend keinen Performanzverlust auf den Trainingsdaten mit sich bringen[13].

12 Details zur Verwendung der Begriffe Induktion und Abduktion in der Informatik finden sich bei Kaminski und Harrach (Kaminski et Harrach 2010).

13 In der Informatik werden neben der Performanz noch andere Parameter eines Algorithmus optimiert. Beispiele sind der benötigte SPEICHERPLATZ oder die

Eine alternative Maßnahme, um gegen Überanpassung oder zu große Komplexität des Entscheidungsbaumes vorzugehen, ist die REGELSTUTZUNG. Hier wird genutzt, dass sich jedes Blatt und damit jede Klassifizierung als ein Kausalsatz darstellen lässt. Im obigen Beispiel wäre der Kausalsatz für das rechte untere Blatt ›wenn weder Samstag noch Sonntag ist und die Cafeteria geschlossen hat, dann wird beim Lieferdienst bestellt‹. Eine Regelstutzung formuliert für jedes Blatt den Weg vom Wurzelknoten als einen solchen Kausalsatz und entfernt anschließend eine der Voraussetzungen aus dem Kausalsatz. Eine Möglichkeit im Beispiel des rechten unteren Blattes wäre ›wenn die Cafeteria geschlossen hat, dann wird beim Lieferdienst bestellt‹. Hier ist gut zu erkennen, dass in diesem Fall eine Regelstutzung die Performanz des Baumes stark beschädigt. Dieser Effekt ist auch nicht überraschend, da der Baum schon sehr kurz war und kein Ast redundant erscheint.

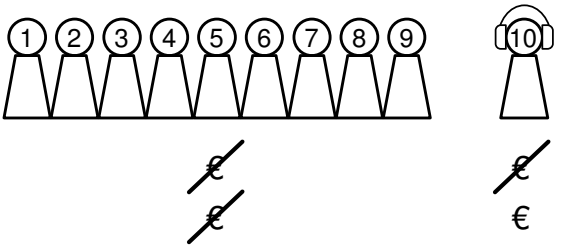
Generell ist bei der Reduktion der Analysetiefe von MLA zum Kampf gegen Überanpassung –insbesondere beim Stutzen von Entscheidungsbäumen – eine Dynamik in der Performanz zu beobachten. Auch wenn sich durch eine Stutzung die Performanz insgesamt nicht verschlechtert, kann dadurch dennoch eine erhebliche Anzahl von neuen Fehlklassifikationen erzeugt werden, solange die Anzahl der vermiedenen Fehlklassifikationen noch größer ist. Eine vom Nutzer hingenommene Erzeugung von zusätzlichen Fehlklassifikationen kann beispielsweise auftreten, wenn die Trainingsinstanzen widersprüchlich sind und ein teilweise irrtümlich entstandener Ast gestutzt wird. In diesem Fall wird statt des Astes ein Blatt mit der mehrheitlich richtigen Klasse eingefügt. Somit werden einige der Eingabedaten, die bisher richtig klassifiziert wurden, in der gestutzten Variante des Gesamtbaumes falsch zugeordnet. Eine Idee dahinter ist, dass häufig unsystematische Fehler oder Rauschen in den Trainingsdaten zu filigranen Verästelungen führen und sich die Nutzer nicht für eine Nachbildung des zufälligen Rauschens interessieren.

Der Vorgang des Stutzens und die Gefahr der Überanpassung im Allgemeinen sollen nachfolgend an einem Beispiel veranschaulicht werden. Die Beispielaufgabe sei eine Klassifikation von Bankkunden als kreditwürdig oder als nicht kreditwürdig. Es sei angenommen, dass in den Trainings-

LAUFZEIT. Diese Parameter sind interdisziplinär jedoch nur von geringem Interesse und werden im Weiteren nicht explizit betrachtet.

daten genau zehn Kunden enthalten sind, die parallel zwei Kredite abbezahlen. Es sei weiter angenommen, dass neun von diesen zehn Kunden mit beiden Krediten im Verzug sind und dass der zehnte Kunde bei genau einem der Kredite zahlungsfähig und außerdem schwerhörig ist.

Abbildung 7: Trainingsdaten des Beispiels zur Kreditwürdigkeit



Ein Entscheidungsbaum wird daraufhin wahrscheinlich bei der Frage nach der Vergabe eines zweiten Kredites den Test ›Ist der Kunde schwerhörig?‹ einfügen. Dies würde auf den Trainingsinstanzen zu einer Verbesserung der Performanz führen, jedoch erscheint dieses Kriterium dennoch die Verallgemeinerbarkeit des Baumes zu gefährden. In diesem Fall würde ein entsprechendes Stutzen des Entscheidungsbaumes zu einer Verschlechterung auf den Trainingsinstanzen führen, jedoch zu einer besseren Performanz auf den späteren Eingabedaten. Natürlich kann es Kunden geben, die aus speziellen Gründen im Verzug sind und dennoch kreditwürdig wären, aber die Hörfähigkeit der Person ist diesbezüglich sehr wahrscheinlich kein geeignetes Entscheidungskriterium. Grafisch bedeutet dies, dass aus dem Entscheidungsbaum A der gestutzte Entscheidungsbaum B wird.

Abbildung 8: Ungestutzter Entscheidungsbaum

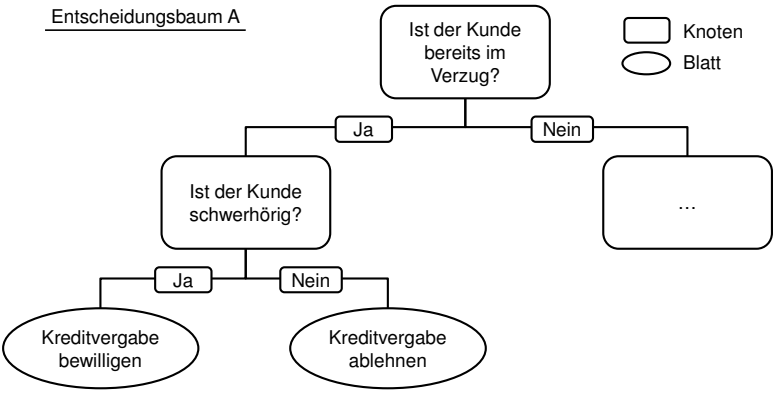
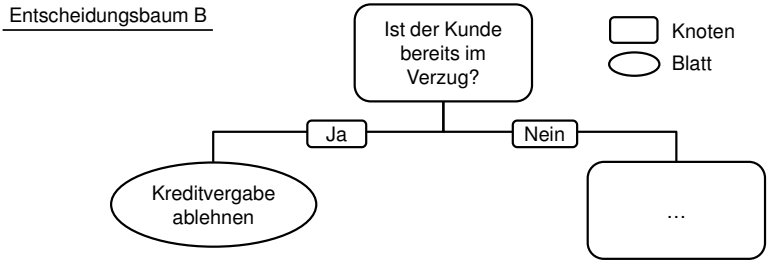


Abbildung 9: Gestutzter Entscheidungsbaum



Im Beispiel der Kreditvergabe war zu sehen, dass Entscheidungsbäume die Eigenschaft haben, dass Randphänomene, die nur wenige Instanzen betreffen, leicht falsch klassifiziert werden können. Eine Möglichkeit dies zu verbessern besteht in der Messung der Performanz auf TESTDATEN und VALIDIERUNGSDATEN. Die Trainingsinstanzen werden hier vor Beginn des Autoadaptionsprozesses in drei Teilmengen aufgeteilt[14]. Die Teilmenge der Testdaten wird indirekt für den Lernvorgang genutzt, etwa um zu prüfen, ob der Strukturvorschlag des MLA eine Überanpassung zeigt. Die Teilmenge der Validierungsdaten wird gar nicht als Rückmeldung innerhalb des Autoadaptionsprozesses verwendet, sondern dient dazu, den Struktur-

14 Die dritte Teilmenge bilden die Trainingsdaten selbst.

vorschlag als das Endergebnis des Autoadaptionsprozesses auf seine Funktionstüchtigkeit zu überprüfen.

Ergänzende Weiterentwicklungen von Entscheidungsbäumen

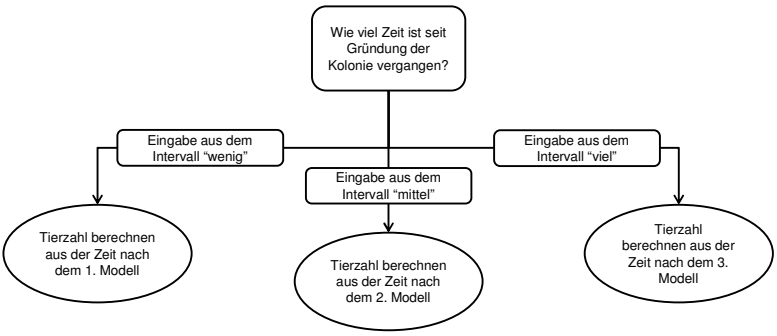
Die Performanz von Entscheidungsbäumen kann erhöht werden, indem ein Gremium verschiedener Entscheidungsbäume erstellt wird, das die Aussagen der beteiligten Entscheidungsbäume bündelt und das Resultat bei der Klassifizierung von neuen Eingabedaten nutzt. In diesem Fall spricht man von ENTSCHEIDUNGSWÄLDERN.

Abschließend soll noch einmal betont werden, dass Entscheidungsbäume überraschend aussagestark sind. Sie sind, wie oben angedeutet, in der Lage, alle Kausalsätze ausdrücken und können auch STETIGE Ein- und Ausgaben verarbeiten. Als stetig wird eine Eingabe bezeichnet, bei der der konkrete Wert aus einem Kontinuum von Werten – einem INTERVALL – stammen kann und nicht auf eine endliche Auswahl beschränkt ist. Die Menge der Vornamen aller lebenden Menschen etwa ist groß, bildet aber eine diskrete Eingabemenge, während die Menge aller Zahlen zwischen null und eins eine stetige und unendlich große[15] Eingabemenge darstellt. Entscheidungsbäume, die stetige Ausgabewerte bewältigen können, werden als REGRESSIONSBÄUME bezeichnet. Im Vorherigen wurde bereits angedeutet, dass Blätter mit Hilfe von Funktionen Klassifizierungen vornehmen können und dadurch stetige Ausgaben erzeugen können, ohne unendlich viele Blätter zu benötigen. Die endliche Anzahl von Blättern hat den Vorteil, dass die Erstellung von lokalen Modellen sehr intuitiv möglich ist. Das heißt, die Rohdaten werden getrennt und die Hoffnung ist, dass der entsprechende Test nur solche Daten derselben Kategorie zuordnet, die auch mit demselben lokalen Modell beschrieben werden können. Der Trick, der stetige Eingaben möglich macht, liegt darin, die Eingaben als REPRÄSENTANTEN von Intervallen zu betrachten. Ein Beispiel sei die Bestimmung der Vermehrungsrate einer bisher unbekannten Art von Ameisen. Zu diesem Zweck soll ein MLA konstruiert werden, das auf der adaptiven Erstellung eines Entscheidungsbaums beruht. Eine große Anzahl von Ameisenkolo-

15 Die Zahlen zwischen null und eins sind nicht beliebig groß, aber sie können eine beliebig große Zahl von Nachkommastellen aufweisen, und somit sind unendlich viele unterschiedliche Eingabedaten möglich.

nien wird gegründet und pro Kolonie wird in regelmäßigen Abständen die verstrichene Zeit in Verbindung mit der jeweiligen Anzahl der Tiere festgehalten. Das MLA soll nach Abschluss des Autoadaptionsprozesses einen Vorschlag für die Struktur der Vermehrung machen. Formal ist die Anzahl von Ameisen endlich, allerdings wird sie schnell sehr groß, das bedeutet, die Eingabe des Strukturvorschlages würde wahrscheinlich Intervalle nutzen. Die Vermehrungsrate beschleunigt sich wahrscheinlich mit zunehmender Anzahl der Tiere bis zu einem natürlichen Maximum der Produktion von Eiern einer Königin. Das heißt, die Verwendung von lokalen Modellen für unterschiedliche Zeitintervalle scheint erfolgsversprechend. Diese lokalen Modelle sind wahrscheinlich sehr viel einfacher darzustellen als ein GLOBALES MODELL, das alle Fälle erfasst. Der Strukturvorschlag könnte beispielsweise drei unterschiedliche Modelle für die Vermehrung erlernen, eines für die Zeit, in der die Königin noch jung ist und das Nest etabliert wird, eines für die Zeit, in der sie ungestört Eier legen kann und eines für die Zeit, in der die Königin alt ist und andere Königinnen schlüpfen.

Abbildung 10: Strukturvorschlag in Form eines Regressionsbaumes



Das obige Beispiel für einen Entscheidungsbaum ist ein untypisches Beispiel für einen Vertreter dieser Klasse von Lernstrategien. Dennoch ist es illustrativ, um zu zeigen, dass sich das Konzept eines MLA, das auf Entscheidungsbäumen basiert, schrittweise in den Vertreter einer anderen Lernstrategie umwandelt, wenn die eigentliche Idee und Intuition hinter der Lernstrategie keine Rolle mehr spielt. Im Fall der Ameisen liegt ein Großteil des Aufwands in der Bestimmung der lokalen Modelle und der Wahl der passenden Intervalle. Die Erstellung des Entscheidungsbaums ist nicht ganz so komplex, kann aber durchaus innerhalb einer größeren Zahl von

Modellen und Intervallen diejenigen identifizieren, die für die Tests der Knoten am geeignetsten sind. Natürlich gibt es weniger eindeutige Beispiele, bei denen ein recht komplexer Entscheidungsbaum in Kombination mit stetigen Ausgaben eingesetzt wird. In der Praxis werden sehr häufig mehrere Lernstrategien in Kombination eingesetzt, allerdings ist es auch in diesen Fällen wichtig und möglich zumindest eine Intuition zu erlangen, was welche Lernstrategie zum entstehenden MLA beiträgt. Die Hauptstärke von Entscheidungsbäumen in diesem Zusammenhang besteht darin, dass sie eine sehr gut nutzbare Schnittstelle zwischen Nutzer und dem Autoadaptionsprozess aufweisen.

2.3.3 Evolutionäres Lernen

Die zweite Klasse von Lernstrategien, die diskutiert werden soll, ist die des EVOLUTIONÄREN LERNENS. Wie bereits angedeutet wurde, setzt sich die Klasse des evolutionären Lernens aus drei stark verwandten Lernstrategien zusammen, deren Diskussion zu großen Teilen vereinheitlicht geführt werden kann. Alle drei nachgeordneten Lernstrategien sind an dieser Stelle gut beschreibbar, da im Rahmen der Darstellung der Entscheidungsbäume die meisten benötigten Grundbegriffe des maschinellen Lernens schon eingeführt wurden.

Motivation der Lernstrategien des evolutionären Lernens

Die drei Lernstrategien des evolutionären Lernens modellieren ihre Autoadaption anhand der Idee eines Evolutionsprozesses als Konzept zur Anpassung von POPULATIONEN an Umwelteinflüsse. Wohlgemerkt wird dabei keine Aussage darüber gemacht, wie Evolution tatsächlich verläuft, sondern verbreitete Ideen wie das Überleben des Stärksten werden genutzt. Die Identifizierung von informatikfernen Modellen und die mathematische Nutzbarmachung von Teilelementen in der Informatik sind wesentliche Konstruktionsmerkmale verschiedener Teilbereiche des maschinellen Lernens. Schon die Bezeichnung von Algorithmen als lernend war eine solche Begriffsübertragung, die allerdings das maschinelle Lernen nicht weiter beeinflusst. Zentral wird diese Vorgehensweise auch bei künstlichen neuronalen Netzen, die mit der Funktionalität von Neuronen im Gehirn ebenfalls einen natürlichen Prozess als Vorbild nehmen und diesen dann gerade *nicht*

im Detail nachbilden, sondern eine begrenzte Auswahl von Prozessbausteinen als Inspiration nutzen, um daraus einen genuin der Informatik entspringenden Ansatz zu entwerfen. Das bedeutet, evolutionäres Lernen modelliert keine faktisch ablaufenden evolutionären Prozesse, genau wie künstliche neuronale Netze keine Gehirne modellieren und statistische Lernverfahren keine statistischen Methoden einsetzen. Stattdessen formuliert evolutionäres Lernen mit Hilfe von statistischen Begriffen und Modellen, deren Definitionen von evolutionären Prozessen inspiriert sind, OPTIMIERUNGSAUFGABEN und andere Problemstellungen, die dann mittels MLA angenähert beziehungsweise bearbeitet werden.

Die Populationen des evolutionären Lernens bezeichnen verschiedene Suchräume, die das MLA im Laufe des Autoadaptionsprozesses bearbeitet. Die einzelnen Strukturvorschläge werden als INDIVIDUEN bezeichnet. Evolutionäres Lernen kann die Leistungsfähigkeit einzelner Individuen direkt vergleichen und nutzt diese Möglichkeit, um evolutionäre Fortschritte beziehungsweise evolutionäre Veränderungen direkt an den Individuen festzumachen. In Konsequenz werden die Individuen untersucht und die Population als Gesamtobjekt wird nicht mit anderen Populationen verglichen, sondern bezeichnet jeweils lediglich die Gesamtheit aller zu einem gewissen Stand des Autoadaptionsprozesses gerade betrachteten Individuen. Die Population verändert sich bei evolutionärem Lernen in jedem Adaptionsschritt und die in iterativer Abfolge entstehenden Populationen werden als GENERATIONEN von Individuen bezeichnet. Populationen werden nur mit nachfolgenden Populationen verglichen und das auch nur, um den Fortschritt in der Entwicklung der Individuen zu veranschaulichen. Mit verschiedenen Individuen wird im maschinellen Lernen umgegangen, wie bei einer klassischen Betrachtung evolutionärer Prozesse mit verschiedenen Populationen umgegangen würde. Die Begriffe sind im evolutionären Lernen jedoch nicht einfach vertauscht. Im Weiteren wird etwa dargestellt werden, wie Individuen im evolutionären Lernen gegeneinander antreten und dass sich einer der Gegner durchsetzt. Dieses Konzept lässt sich zwar prinzipiell, aber nicht ohne Weiteres auf Populationen übertragen. Wie bereits angedeutet wurde, stellt diese Verwendung der Begriffe in der Informatik kein Problem dar, da die MLA formal vollständig mit Mitteln der Informatik konstruiert und ausgewertet werden. Das maschinelle Lernen basiert meist nicht auf biologischen Modellen und es ist wichtig, dass Begriffe

wie Evolution im Sinne der Informatik gelesen werden müssen, um einen interdisziplinären Zugang zu erhalten.

Evolutionäre Strategien lernen, indem sie ziellos experimentieren und erfolgreiche Ergebnisse weiterverfolgen. Gelernte Muster liegen in Form von bislang erfolgreichen Strukturvorschlägen vor. Das Zusammenspiel von Systematik und Zufall besteht bei evolutionärem Lernen darin, dass die Erzeugung der Designs für mögliche Strukturvorschläge un- beziehungsweise zufallsgesteuert ist, anschließend jedoch systematisch die besten Strukturvorschläge ausgewählt werden. Evolutionäres Lernen findet zufällig mögliche Strukturvorschläge und vergleicht diese systematisch. Dies wird im zweiten Hauptteil ein wesentlicher Schritt weg von Lernstrategien sein, die vollständig systematisch und auf Vorwissen basierend vorgehen beziehungsweise optimieren.

Evolutionäres Lernen eignet sich für den Einsatz in sehr komplexen Umgebungen, deren Hintergründe nicht verstanden werden. Diese Komplexität kann bewältigt werden, da nicht versucht wird, die Zusammenhänge der Umwelt zu erklären, sondern das System durch eine große Anzahl von zufälligen Veränderungen anzupassen.

Einführungsbeispiel zu evolutionärem Lernen

Als kurzes Anwendungsbeispiel soll der Entwurf eines elektronischen Schaltkreises dienen (Koza et al. 1996):

Im Vorfeld des eigentlichen Entwurfs werden die Anforderungen an das fertige Produkt formuliert und als SPEZIFIKATIONEN festgehalten. Weiterhin wird eine Simulationssoftware bereitgestellt, um Schaltkreisentwürfe auf ihre Leistungsfähigkeit zu testen. Schließlich werden alle für die Konstruktion des Schaltkreises verfügbaren Bauteile im Rahmen der Simulationssoftware als simulierter Werkzeugkasten dargestellt und es wird eine Anzahl simpler Standard-Schaltkreise entworfen, deren Gesamtheit als ANFANGSPOPULATION betrachtet wird. Sowohl der simulierte Werkzeugkasten als auch die Standard-Schaltkreise sind hierbei unabhängig von den genauen Spezifikationen und können unverändert von einem anderen Schaltkreisdesign übernommen werden.

Nachdem diese Vorarbeit geleistet wurde, modifiziert ein evolutionär lernendes Artefakt die Individuen der Population – im ersten Schritt die Standard-Schaltkreise – indem zufällig Bauteile oder Verbindungen zwi-

schen Bauteilen ergänzt oder entfernt werden. Anschließend vergleicht die Simulationssoftware die Ausgabewerte und ggf. die Konstruktionskosten der entstandenen Schaltkreise mit den geforderten Spezifikationen. Die leistungsstärksten Individuen werden ermittelt und als zweite Generation betrachtet. Die entstehenden Schaltkreise sind dabei in der Mehrzahl nicht funktionstüchtig, allerdings steigt der Anteil an funktionstüchtigen Schaltkreisen mit zunehmender Zahl von Generationen deutlich an. Das MLA erstellt solange weitere Generationen, bis die Spezifikationen ausreichend gut erfüllt werden.

Definition evolutionären Lernens

Auch im Allgemeinen sind vor dem Einsatz eines auf evolutionärem Lernen basierenden MLA zur Bearbeitung einer Problemstellung einige Vorarbeiten zu erledigen. Es wird eine Anfangspopulation benötigt und die enthaltenen Individuen werden meist als eine Zusammenstellung von auf Vorwissen beziehungsweise Rahmenbedingungen basierenden, unveränderlichen Einzelteilen modelliert, um zu vermeiden, dass zu viele nutzlose Lösungen entstehen. Zu diesem Zweck wird häufig eine REPRÄSENTATION der Problemstellung erstellt, die implizit den Suchraum der sinnvollen und damit syntaktisch zulässigen Strukturvorschläge vorgibt, woraus sich wiederum eine Anfangspopulation von Strukturvorschlägen gewinnen lässt. Diese Repräsentation kann auch eine CODIERUNG sein, das heißt, die vorliegenden Rahmenbedingungen werden systematisch und wiederholbar in Eingabedaten übersetzt und so für das MLA registrierbar gemacht. Gleichzeitig unterstehen die aus der Codierung entstandenen Eingabedaten nach der Übergabe an das MLA nicht mehr den Gesetzmäßigkeiten, die außerhalb des MLA vorliegen. Der Verzicht auf eine Repräsentation im obigen Beispiel ist daran ersichtlich, dass die durch das MLA zurückgegebenen Strukturvorschläge nach einer DECODIERUNG, das heißt einer Rückübersetzung in Schaltkreise gemäß derselben Systematik wie zuvor, zum Großteil nicht funktionstüchtig sind. Eine Codierung ist eine formale Grammatik, mittels derer die Individuen zu beschreiben sind und entsprechend der alle Strukturvorschläge vom Nutzer interpretiert werden. Strukturvorschläge müssen entsprechend syntaktisch korrekt sein und die Codierung gibt damit den Autoadaptionsprozessen des evolutionären Lernens einen Rahmen. Die Codierung kann für einzelne Problemstellungen sehr unterschiedlich ausse-

hen und die weiter unten besprochenen Varianten evolutionären Lernens unterscheiden sich insbesondere in den Methoden zur Konstruktion solch einer Grammatik der Strukturvorschläge. In grober Analogie entspricht in der natürlichen Evolution die DNA solchen Codierungen, die nach Decodierung als Vorschlag für die Struktur eines Lebewesens interpretiert werden kann.

Im Autoadaptionsprozess des evolutionären Lernens werden die Anfangspopulation und jede darauffolgende Generation von Strukturvorschlägen durch an die Evolution angelehnte EVOLUTIONÄRE OPERATOREN so lange verändert, bis ein Strukturvorschlag entsteht, der nach einem vordefinierten Leistungsmaßstab hinreichend optimale Ergebnisse erzielt. Die typischen evolutionären Operatoren sind hierbei die zufällige MUTATION eines Strukturvorschlags oder die REKOMBINATION mehrerer Strukturvorschläge. Die Verwendung des Begriffs der Mutation bedeutet hier, dass ein kleiner Teil der Codierung zufällig abgeändert wird, während von Rekombination gesprochen wird, wenn syntaktisch vergleichbare Abschnitte – gegebenenfalls deutlich unterschiedlicher Länge – der Strukturvorschläge ausgetauscht werden. Die Bewertung der Individuen mittels des vordefinierten Leistungsmaßstabs wird als Einsatz der FITNESSFUNKTION und das Ergebnis der Bewertung als die FITNESS der Individuen bezeichnet. Die Fitnessfunktion bewertet üblicherweise die Performanz über den Testdaten, kann aber auch Parameter wie die Komplexität des betrachteten Strukturvorschlags berücksichtigen. Evolutionäres Lernen kann somit als Optimierung bezüglich der Fitness verstanden werden. Zusammengefasst werden die folgenden Prozessschritte durchlaufen:

A. Initialisierung
Der Prozess beginnt mit der Erstellung einer Codierung und der Zusammenstellung einer Anfangspopulation aus syntaktisch korrekt codierten Strukturvorschlägen.

B. Evolutionsschritt:
Die aktuelle Generation wird mittels evolutionärer Operatoren evolviert.

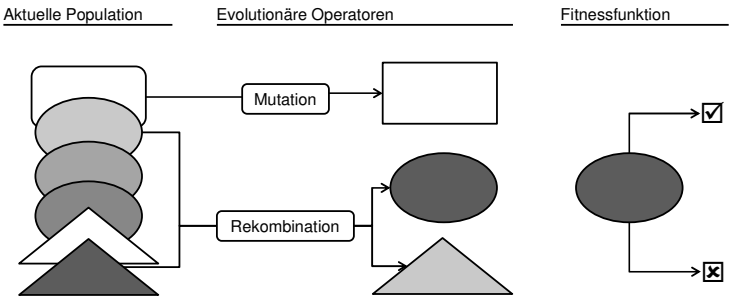
C. Selektionsschritt:
Die Strukturvorschläge der aktuellen Generation werden mittels einer vorgegebenen Fitnessfunktion beurteilt, und die fittesten Strukturvorschläge werden PROBABILISTISCH für die Erzeugung der nächsten Gene-

ration von Individuen, das heißt der nächsten Population, ausgewählt. Anschließend wird der nächste Evolutionsschritt **B** durchgeführt.

Probabilistisch bezeichnet die Zuordnung auf Basis einer Wahrscheinlichkeit und meint in der Praxis meist, dass die Wahrscheinlichkeit der Auswahl eines bestimmten Strukturvorschlags typischerweise dem Anteil der Fitness der entsprechenden Strukturvorschläge an der summierten Fitness der Population in der aktuellen Generation entspricht.

Sollte die Aufgabe darin bestehen spezielle geometrische Formen zu konstruieren, könnte eine Visualisierung des Evolutionsschrittes und Teilen des Selektionsschrittes wie folgt aussehen.

Abbildung 11: Beispiel für Evolutions- und Selektionsschritt



Hier wurde mit Hilfe der Rekombination als evolutionärem Operator eine dunkelgraue Ellipse erzeugt. Diese Ellipse wird anschließend auf ihre Fitness überprüft, und sollte das Ziel beispielsweise darin liegen einen schwarzen Kreis darzustellen, so wäre die entstandene Ellipse das fitteste Individuum der derzeitigen Generation. In der Visualisierung fehlt für den Selektionsschritt noch die Erstellung einer neuen Generation.

Die Definition der gemeinsamen Grundlage der drei Klassen evolutionären Lernens hat eine noch nicht beleuchtete Besonderheit in Hinblick auf die Frage, inwieweit ein MLA im Rahmen des Autoadaptionsprozesses Sensordaten berücksichtigt oder berücksichtigen kann. Formal bestand dieses Problem schon bei der Betrachtung der MLA, die auf die Konstruktion von Entscheidungsbäumen aus waren. Allerdings wurde die Betrachtung dort ausgespart, da der Begriff der Fitnessfunktion die Analyse dringlicher macht und gleichzeitig erleichtert. Wie schon bei der Erstellung von Ent-

scheidungsbäumen kann der gesamte Autoadaptionsvorgang ablaufen und einen Strukturvorschlag erzeugen, sobald die ursprünglichen Trainingsdaten übergeben wurden. Der resultierende Autoadaptionsprozess läuft bei der Konstruktion von Entscheidungsbäumen ohne spezielle Weiterentwicklungen immer gleich ab. Der Lernprozess bei evolutionärem Lernen kann jedoch eine autoadaptive Fitnessfunktion aufweisen. Tatsächlich ist diese Option ein Hauptgrund dafür, dass die Fitnessfunktion einen eigenen Namen erhält und als prominenter Teil der Lernstrategie betrachtet wird. Im Falle des Einsatzes einer Fitnessfunktion, die sich im Laufe des Autoadaptionsprozesses verändert, hängt der von einem evolutionär lernenden MLA erzeugte Strukturvorschlag gegebenenfalls noch von weiteren Einflüssen wie der Reihenfolge der Eingabe der Trainingsdaten oder der Uhrzeit zu Beginn des Prozesses ab[16]. MLA dieser Art könnten die Bewertung der Fitness einer Generation auch als Ausgabe an eine Testumgebung übergeben und als Eingabe die Bewertung der Fitness zurückerhalten. In diesem Fall wäre der Autoadaptionsprozess in gewisser Hinsicht immer nach einer Iteration beendet und andererseits würde er andauern und immer weitere Sensordaten aufnehmen, wenn sich die externen Anforderungen an die Fitness der Individuen häufig ändern. Wichtig ist hier zu verstehen, dass sowohl Entscheidungsbäume als auch evolutionäres Lernen am besten als einmal initiiertes abgeschlossenes System agieren können und die nachträgliche Integration von zusätzlichen Trainingsdaten zumindest kompliziert ist.

Die bei dieser Lernstrategie realisierte Suche unterscheidet sich durch ihre den Zufall nachbildenden Elemente deutlich von den Suchstrategien der übrigen MLA. Evolutionäres Lernen bekommt durch die Zufallselemente einen sehr unvorhersehbaren beziehungsweise unstetigen Charakter. Zwischen den Generationen können sehr unterschiedliche und zum Teil sehr extreme Veränderungen der Strukturvorschläge auftreten. Die Suche wird als STRAHLENSUCHE bezeichnet, bei der weder auf Basis eines groben Vorwissens gesucht, noch eine lokale Eigenschaft optimiert wird.

16 Ein Algorithmus kann keine echten Zufallsgrößen erzeugen. Zufallszahlengeneratoren nutzten häufig quasi-zufällige Parameter wie die Systemzeit um annähernd zufällige Ergebnisse zu erzeugen.

Vorteile und Nachteile evolutionären Lernens

Die Vorteile evolutionären Lernens entsprechen den Stärken, die auch der Evolution gemeinhin zugesprochen werden. MLA dieser Art sind eine robuste Methode der Anpassung an komplexe Systeme. Evolutionäres Lernen kann Suchräume betrachten, die komplexe, aufeinander reagierende Elemente beschreiben, bei denen die Auswirkung einzelner Komponenten des Strukturvorschlages auf dessen Gesamtfitness schwierig zu modellieren ist. Beispiele für solche Suchräume sind die Optimierung einer Robotersteuerung oder eines hochmodularen Computerprogramms. Auch bei extrem großen Suchräumen kann evolutionäres Lernen aufgrund der Unvorhersehbarkeit der Suchschritte gute Ergebnisse erzielen. Analog zur Evolution benötigt evolutionäres Lernen keine oder kaum externe Steuerung und kann autoadaptiv konstruiert werden. Das bedeutet, die Eigenschaften der evolutionären Operatoren und der meisten sonstigen Parameter der Lernstrategien können dynamisch angepasst werden. Der Verzicht auf eine externe Steuerung führt nicht zwangsläufig dazu, dass das System ziellos agiert, allerdings besteht sehr wohl die Möglichkeit auf ein explizites Ziel zu verzichten und lediglich eine rudimentäre Vorgabe für eine zwar ziellose, aber dennoch systematische Autoadaption zu machen. Wird diese Vorgabe zusätzlich autoadaptiv gestaltet, so können zuvor nicht bedachte Muster gefunden werden, allerdings senkt diese Vorgehensweise aufgrund der wenig vorstrukturierten Suche stark die Geschwindigkeit des MLA. Aus praktischer Sicht schließlich zeichnet sich evolutionäres Lernen dadurch aus, dass es sich einfach parallelisieren lässt und damit sehr effizient implementiert werden kann.

Von Nachteil bei evolutionärem Lernen ist vor allem, dass es relativ zu anderen Lernstrategien in unmodifizierter Form – analog zu biologischer Evolution – sehr langsam seine Performanz über den Trainingsdaten verbessert und dass es schwieriger ist, eine Erfolgsgarantie für eine zufallsgeprägte Suche zu erstellen beziehungsweise zu errechnen. Weiterhin ist die Wahl der Codierung der Strukturvorschläge von sehr großer Bedeutung, da die Suche syntaktische Einschränkungen bei der Formulierung der Anfangspopulation mitunter nicht überwinden kann.

Varianten evolutionären Lernens

Evolutionäres Lernen hängt, wie bereits angedeutet, sehr stark von der Form der gewählten Modellierung beziehungsweise Codierung der Problemstellung ab. Die Zahl der Abhängigkeiten ist jedoch noch deutlich größer, etwa üben auch Aspekte wie die Auswahl der einzusetzenden evolutionären Operatoren einen großen Einfluss aus, und sogar scheinbar nachrangige Parameter wie die Anwendungsreihenfolge spielen eine Rolle. Diese Vielzahl von Einflussfaktoren ist insofern bemerkenswert, als evolutionär lernende Artefakte durch Manipulation dieser Faktoren sehr unterschiedliche Vorgehensweisen aufweisen können. Die drei wichtigsten Formen evolutionären Lernens lassen sich darüber hinaus gut interdisziplinär betrachten. Unabhängig von den Spezifika evolutionären Lernens bietet sich so die Möglichkeit, ein Verständnis für das Zusammenspiel unterschiedlicher Varianten eines gemeinsamen Ansatzes maschinellen Lernens zu erlangen. Solch ein Verständnis ist auch in der weiteren Analyse des maschinellen Lernens hilfreich.

Die drei wichtigsten Konkretisierungen der Idee des evolutionären Lernens und die zugrundeliegenden Ideen werden im Weiteren kurz vorgestellt. Diese wesentlichen Varianten sind:

- Genetische Algorithmen
- Genetische Programmierung
- Evolutionsstrategien

Ergänzend zu den hier vorgestellten Varianten evolutionären Lernens können Hypothesen auch durch symbolische Repräsentationen beschrieben werden – wie etwa im Beispiel der geometrischen Figuren in Abbildung 11.

Genetische Algorithmen

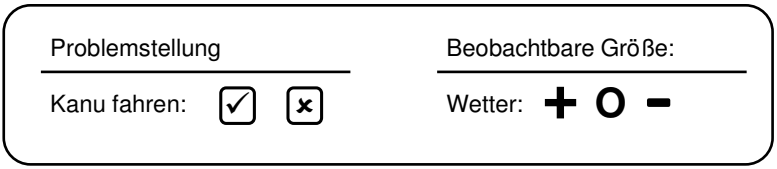
Codierung im Rahmen genetischer Algorithmen

Beim Einsatz GENETISCHER ALGORITHMEN werden Strukturvorschläge typischerweise als BITFOLGEN – Ketten von Nullen und Einsen – codiert, die im Kontext des Problems interpretiert werden müssen. Dies bedeutet, dass die Attribute, die bei der Codierung eines Strukturvorschlages von Bedeutung sind, auf eine festgelegte Weise oder an einer festgelegten Stelle in der Bit-

folge hinterlegt sind und dass diese Codierung in ihrer Gesamtheit den vorliegenden Strukturvorschlag beschreibt.

Der Rolle der Bitfolge würde bei einem Lebewesen in etwa die Rolle der DNA entsprechen. Die Codierung mittels Bitfolgen wird entsprechend als GENOTYP-PHÄNOTYP-ABBILDUNG von Bitfolgen auf die Menge der Strukturvorschläge betrachtet werden. Die wesentliche Idee hinter dieser Form der Codierung liegt darin, dass eine möglichst ATOMARE, das heißt minimal komplexe, Beschreibungssprache eine maximale Ausdrucksfähigkeit erzeugt (Goldberg 1990, S. 4ff). Bitfolgen eignen sich sehr gut für die Codierung von Kausalsätzen, da jedes Bit als Entscheidung für oder gegen etwas interpretiert werden kann. Ein konkretes Beispiel für die Codierung eines Strukturvorschlags ist etwa das Problem, zu entscheiden, ob eine Kanufahrt unternommen werden soll.

Abbildung 12: Beispielcodierung von ›Kanu fahren‹



In diesem Beispiel ist das Wetter das einzige Attribut und kann die drei Zustände gutes, normales und schlechtes Wetter annehmen. Ein Strukturvorschlag, der in jeder Situation eine Entscheidung ermöglicht, muss jeden möglichen Zustand des Attributes berücksichtigen. Ein möglicher Strukturvorschlag in Form eines Kausalsatzes wäre die Formulierung ›gutes und normales Wetter sind akzeptabel, schlechtes Wetter ist es nicht‹. Zu drei möglichen Zuständen muss jeweils eine Empfehlung ausgesprochen werden, deshalb könnte für die Codierung des Kausalsatzes ein dreistelliger Bitstring eingesetzt werden. Das genannte Beispiel ›ja-ja-nein‹ als ein möglicher Strukturvorschlag könnte mit ›110‹ in Form einer Bitfolge codiert werden. Wenn ein zweites Attribut betrachtet werden soll, könnte die Bitfolge einfach um so viele Stellen verlängert werden, wie das neue Attribut Zustände einnehmen kann. Die ersten drei Stellen der Bitfolge wären anschließend für die bisherige Codierung reserviert und die restlichen Stellen für die Codierung der Aussage des Strukturvorschlages bezüglich des zweiten Attributes. Wenn in einem Strukturvorschlag ein einzelnes Attribut keine Rolle spielt, können die reservierten Stellen pauschal mit einer eins be-

schrieben werden. Wenn etwa in obigem Beispiel das Wetter irrelevant ist und die Kanufahrt unabhängig davon sowieso stattfinden wird, entspräche das dem Strukturvorschlag ›111‹. Diese Vorgehensweise stellt sicher, dass die Bitfolgen aller Strukturvorschläge die gleiche Codierungslänge aufweisen. Dadurch wird es deutlich einfacher sicherzustellen, dass jede eingesetzte Bitfolge auch einen syntaktisch korrekten Strukturvorschlag darstellt[17].

Evolutionäre Operatoren und Selektion bei genetischen Algorithmen

Genetische Algorithmen nutzen sowohl Mutation als auch Rekombination als evolutionäre Operatoren, wobei der Schwerpunkt häufig auf dem Einsatz der Rekombination liegt. Mutationen werden realisiert, indem ein zufälliges Bit geändert wird. Rekombinationen werden durchgeführt, indem Teile von zwei Bitfolgen ausgetauscht werden. Es wird nach einem vorgegebenen oder zufälligen Muster entschieden, welche Teilstücke zwischen den beiden Bitfolgen ausgetauscht werden, und zwei neue Bitfolgen mit ausgetauschten Teilen werden erzeugt. Das vorgegebene Muster zu Rekombination wird MASKE genannt.

Genetische Algorithmen können in einem gewissen Rahmen ihre eigenen evolutionären Operatoren adaptieren, indem sie beispielsweise die codierte Maske explizit als einen zusätzlichen Teil der eigentlichen Bitfolge des Strukturvorschlages betrachten und evolvieren. Auf diese Weise können Entscheidungen über die Häufigkeit des Einsatzes von evolutionären Operatoren bei späteren Generationen getroffen werden. Darüber hinaus können die evolutionären Operatoren selbst verändert werden, etwa indem die Maske als Teil des Strukturvorschlags der letzten Generation ebenfalls mutiert. Dieses Vorgehen ermöglicht genetischen Algorithmen zumindest prinzipiell, mittels Autoadaption zu neuen Suchstrategien zu gelangen. Dies ist theoretisch auch bei den anderen Formen evolutionären Lernens umsetzbar, wird dort jedoch kaum genutzt.

Der Selektionsschritt bei genetischen Algorithmen entspricht dem Grundmuster für evolutionäres Lernen. Die Selektion wird auf Basis einer

17 Genetische Algorithmen können auch auf Basis von Eingaben arbeiten, die nicht in Form einer Bitfolge vorliegen. Entsprechende Varianten genetischer Algorithmen, die andere Formen einer Genotyp-Phänotyp Analogie aufweisen, werden in der Praxis auch erfolgreich umgesetzt (Salomon 1995).

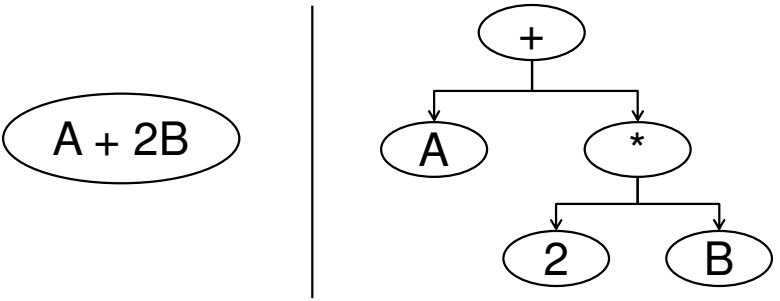
Fitnessfunktion vorgenommen und nur die fittesten der neu entstandenen Strukturvorschläge werden Teil der nächsten Generation von Individuen.

Genetische Programmierung

Codierung im Rahmen genetischer Programmierung

Der Ansatz der GENETISCHEN PROGRAMMIERUNG liegt darin, die Auswirkung evolutionärer Operatoren auf Algorithmen zu betrachten. Algorithmen können sehr unterschiedlich repräsentiert werden, werden im Kontext der genetischen Programmierung aber häufig als BÄUME dargestellt. Der sehr kurze Algorithmus zur Auswertung der Formel ›A + 2B‹ kann etwa in der folgenden Form als Baum dargestellt werden.

Abbildung 13: Auswertungsbaum / Parse Tree von ›A + 2B‹



Die Idee hinter der genetischen Programmierung ist, dass keine zusätzliche Codierung vorgenommen werden muss, falls die Individuen der aktuellen Population ein Zusammenwirken von Algorithmen darstellen. Auch die Repräsentation als Baum stellt keinen zusätzlichen Schritt dar, da Algorithmen von der ausführenden Hardware meist sowieso in solch einer Form abgearbeitet werden[18]. Auf Basis dieser Repräsentation können ganze Algorithmen evolviert werden. Wenn keine zusätzliche Codierung notwendig ist, weil die Strukturvorschläge bereits codiert vorliegen, unterscheiden die Einwirkungen von evolutionären Operatoren auf die Codierung sich nicht

18 Die Darstellung der Strukturvorschläge als Baum entspricht einem PARSE TREE oder ›Syntaxbaum‹ des gewählten Algorithmus, wie in der obigen Abbildung bereits angedeutet wurde.

von Einwirkungen direkt auf den Strukturvorschlag selbst. Die Unterscheidung zwischen GENOTYP und PHÄNOTYP[19] des Individuums, die bei genetischen Algorithmen zentral war, wird irrelevant. Die evolutionären Operatoren können stattdessen direkt auf den Phänotyp zugreifen. Dies hat den Nachteil, dass, auch wenn das Evolvieren eines Algorithmus prinzipiell jede denkbare oder sinnvolle Form erzeugen kann, dies in der Umsetzung sehr komplexe oder schlicht sehr lange Strukturvorschläge erfordern wird. Ein Beispiel hierfür aus dem Schachspiel ist der Versuch, bestimmte Felder mit Hilfe einer speziellen Figur zu erreichen. Ein Strukturvorschlag, der eine Springer-Zugfolge beschreibt, ist in den meisten Fällen deutlich länger als ein Strukturvorschlag für den Einsatz einer Dame. Analog kann im Rahmen der genetischen Programmierung auf sehr unterschiedliche Hilfsalgorithmen zurückgegriffen werden. Die Hauptschwierigkeit beim Einsatz genetischer Programmierung besteht entsprechend darin, diejenigen Hilfsalgorithmen zu identifizieren, die zur Bearbeitung der konkreten Problemstellung besonders geeignet sind. Die Identifizierung eines im jeweiligen Kontext gut einsetzbaren Algorithmus führt meist überhaupt erst zur Wahl der genetischen Programmierung als Lernstrategie. In Konsequenz verschiebt die genetische Programmierung das Problem der Wahl eines geeigneten Algorithmus zunächst nur. Allerdings bieten sich durch den bei der Verschiebung gewonnenen neuen Kontext auch neue Ansätze Strukturvorschläge zu erstellen.

Evolutionäre Operatoren und Selektion genetischer Programmierung

Die Verwendung von evolutionären Operatoren bei der genetischen Programmierung ist nicht eindeutig zu beschreiben. Sowohl die Rekombination von Teilbäumen in Form des Austauschs zweier Äste als auch die Mutation einzelner Knoten werden eingesetzt.

Die Selektion erfolgt bei genetischer Programmierung prinzipiell analog zu derjenigen bei genetischen Algorithmen, allerdings erfolgt die Auswahl häufig als TURNIERSELEKTION oder ROULETTESELEKTION. Die Turnierselektion lässt die für die Aufnahme in die nächste Generation in Frage kommenden Individuen in Form eines Turnieres gegeneinander antreten,

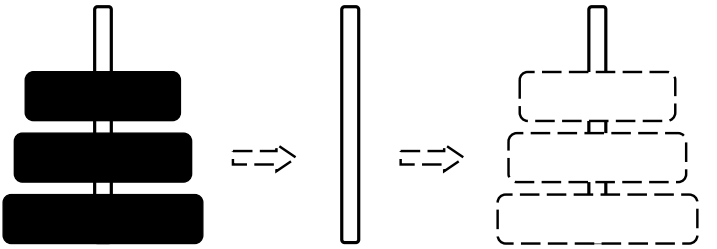
19 Der Genotyp ist die genetische Codierung, das heißt die Information der Gene, die eine biologische Zelle im Zellkern trägt. Der Phänotyp stellt die Realisierung dieser Codierung dar – etwa in Form einer Haarfarbe.

wobei jeweils das Individuum mit der niedrigeren Fitness ausscheidet. Die bestplatzierten Individuen werden anschließend in die nächste Generation aufgenommen. Wenn die Paarung von Individuen zufällig war, kann so durchaus das zweitbeste Individuum im Selektionsschritt ausgesondert werden und entsprechend ist eine gewisse Durchmischung jenseits der Fitness gewährleistet. Die Rouletteselektion ordnet jedem Individuum eine von dessen Fitness abhängige Wahrscheinlichkeit zu, mit der das Individuum in die nächste Generation aufgenommen wird. Die Wahrscheinlichkeit kann sich dabei an der absoluten Höhe der Fitness orientieren und Individuen mit sehr viel höherer Fitness auch sehr viel höhere Wahrscheinlichkeiten zuordnen, oder die Fitness relativ zu den anderen Individuen wird als Platzierung interpretiert. Im zweiten Fall erhält das Individuum mit der höchsten Fitness die größte Wahrscheinlichkeit und das Individuum mit der geringsten Fitness die niedrigste Wahrscheinlich zur Aufnahme in die nächste Generation. Beide Selektionsweisen sollen vermeiden, dass die genetische Programmierung zu einer reinen Optimierung der Fitness wird. Genetische Algorithmen setzen zu diesem Zweck autoadaptive genetische Operatoren ein, während genetische Programmierung einen zufälligen Aspekt in den Selektionsschritt aufnimmt. Dieser Versuch der aktiven Distanzierung von ZIELORIENTIERTEN OPTIMIERUNGSALGORITHMEN deutet an, dass die später im zweiten Hauptteil vorgeschlagene Unterscheidung optimierungsnäherer und -fernerer Ausprägungen maschinellen Lernens auch für die Informatik interessant sein kann.

Beispiele für genetische Programmierung

Ein Beispiel für genetische Programmierung bildet die Suche nach einem Strukturvorschlag zur Lösung des Spiels ›Turmbau zu Hanoi‹.

Abbildung 14: Aufbau und Zielzustand des Turmbaus von Hanoi



Die Aufgabenstellung beim Turmbau von Hanoi beginnt mit der Vorgabe von drei Pfählen, wobei auf einem Pfahl drei der Größe nach sortierte, unterschiedlich breite Scheiben platziert sind. Ziel ist es, diese Scheiben in der gleichen Anordnung auf einem anderen Pfahl zu platzieren. In jedem Zug darf immer nur genau eine Scheibe auf einmal bewegt werden und es dürfen auf allen Pfählen nur kleinere auf größeren Scheiben platziert werden. Die denkbaren Züge entsprechen bei dieser Aufgabe den Bewegungen einer speziellen Scheibe von einem bestimmten Pfahl zu einem anderen Pfahl und sind algorithmisch einfach beschreibbar. Strukturvorschläge setzen sich direkt aus einer Anreihung von Zügen zusammen, und sowohl die Zulässigkeit der Zugfolge als auch deren Funktionstüchtigkeit sind einfach zu simulieren und direkt durch den Nutzer überprüfbar. Wesentlich ist hier, dass die denkbaren Züge als fixe Komponenten des Strukturvorschlages betrachtet werden. Die Menge der denkbaren Züge darf nicht erweitert werden und nur die Reihenfolge der Züge darf durch evolutionäre Operatoren evolviert werden. Die Art und Weise, wie die Reihenfolge evolviert wird, darf wiederum selbst evolviert werden, das heißt, die evolutionären Operatoren können autoadaptiv sein. Gleichzeitig können durch die Fixierung der denkbaren Züge dennoch keine überraschenden Lösungen auftreten. Denkbar wären durchaus auch evolvierte Zugoptionen, etwa könnte die Hälfte der dritten Scheibe auf den zweiten Pfahl geschoben werden. Wenn in der Praxis alle undenkbaren Züge tatsächlich auch nicht umsetzbar sind, ist solch eine Vorgehensweise wenig hilfreich, aber in der Praxis treten selten unveränderliche Rahmenbedingungen auf, gegen die nicht verstoßen werden darf. Ein MLA kann hier helfen, die Menge der denkbaren Reaktionen im Kontext einer speziellen Aufgabe nur als einen Teil des Suchraums für den Strukturvorschlag zu betrachten.

Ein etwas komplexeres Beispiel für genetische Programmierung ist das eingangs beschriebene automatisierte Design von elektronischen Schaltkreisen (Koza et al. 1996), das in der Praxis mittels genetischer Programmierung realisiert werden kann. Die technischen Details einer solchen Realisierung stehen in der interdisziplinären Betrachtung des maschinellen Lernens nicht im Fokus, allerdings ist es hilfreich, eine Intuition der Größenverhältnisse der Parameter zu entwickeln. Ergänzend zur eingangs erfolgten Darstellung daher nachfolgend Größenordnungen der Parameter aus einer praktischen Anwendung:

- In jeder Generation bestand die Population aus über 500.000 Schaltkreisentwürfen.
- Die besten 10% jeder Generation wurden unverändert wiederverwendet, um mehr Mutationen und Rekombinationen zu erlauben, da bereits eine große Anzahl von performanten beziehungsweise fitten Individuen aus der aktuellen Generation übernommen und somit ein größeres Risiko eingegangen werden konnte.
- 1% der Individuen der nächsten Generation wurde durch Mutationen, der übrige Anteil durch Rekombinationen gewonnen.
- Der Anteil der Schaltkreisentwürfe, die im Rahmen der Simulationssoftware sinnvoll dargestellt werden konnten, begann bei weniger als 5% und stieg im Laufe der nächsten Generationen erst auf 15%, dann auf 25%. Im Schnitt über alle Generationen entstanden etwa zu 90% sinnvolle Ergebnisse.
- Nach ungefähr 140 Generationen entstand ein Schaltkreis mit den gewünschten Spezifikationen.

Der vergleichsweise seltene Einsatz von Mutationen überrascht auf den ersten Blick, allerdings gilt hier dieselbe implizite Annahme wie beim Turmbau von Hanoi. Die Schaltkreiselemente, die dem MLA als mögliche Komponenten für die Erstellung eines Strukturvorschlages genannt wurden, entsprechen den denkbaren Lösungen. Zwar ist beim Schaltkreisdesign klar, dass auch sehr nützliche Designs noch nicht entdeckt wurden, allerdings wird der Bereich von den Nutzern als gut verstanden wahrgenommen, und die Chancen auf halb-zufällige Entdeckungen durch ein MLA werden als fast vernachlässigbar erachtet. In anderen Kontexten kann sich diese Einschätzung natürlich drastisch ändern.

Evolutionsstrategien

Die Grundannahme und Voraussetzung für die Verwendung von Evolutionsstrategien[20] ist, dass im jeweiligen Kontext kleine Änderungen der Strukturvorschläge nur kleine Änderungen in deren Performanz und anderen wichtigen Eigenschaften herbeiführen. Dies entspricht der Annahme einer hinreichend STARKEN KAUSALITÄT als universellem Weltverhalten. Die zur Performanzmessung verwendete Fitnessfunktion wird bei Evolutionsstrategien als ZIELFUNKTION bezeichnet. Das ist insofern von Bedeutung, als der Begriff der Zielfunktion sich üblicherweise in der MATHEMATISCHEN OPTIMIERUNG wiederfindet, in der versucht wird eine Zielfunktion zu MAXIMIEREN, indem Parameter verändert werden, die die Zielfunktion beeinflussen.

Die Annahme einer starken Kausalität soll sicherstellen, dass beim Einsatz der evolutionären Operatoren Mutationen, die zu besonders großen Veränderungen im Strukturvorschlag oder dessen Eigenschaften führen, und Mutationen, die zu sehr kleinen Veränderungen führen, langfristig systematisch mit geringerer Wahrscheinlichkeit zu einem Fortschritt im Sinne der Zielfunktion führen als Mutationen, die eine zu bestimmende, optimale Größe von Veränderungen herbeiführen. Das Ausmaß der Veränderungen, die eine Mutation auslöst, wird als MUTATIONSSCHRITTWEITE bezeichnet. Ein Ziel von Evolutionsstrategien besteht darin MUTATIONSSCHRITTWEITENBAND zu bestimmen, das aussagt, in welchem Rahmen die Mutationsschrittweite in einem Evolutionsschritt minimal und maximal liegen sollte. Entsprechend ist ein zentrales Element von Evolutionsstrategien die Adaption der Mutationsschrittweite und daraus wiederum folgt, dass Evolutionsstrategien im Gegensatz zu anderen Formen evolutionären Lernens bereits in ihrer Grundform autoadaptive evolutionäre Operatoren einsetzen. Evolutionsstrategien besitzen darüber hinaus aufgrund der Voraussetzung einer starken Kausalität und der gezielten Steuerung der Mutationsschrittweite im Gegensatz zu den anderen Formen evolutionären Lernens ein mathematisches nutzbares Fundament. Auf dieser Basis können mathematische Analysen der KONVERGENZ des Autoadaptionsprozesses durchgeführt werden,

20 Der Begriff der Evolutionsstrategie wird mitunter in der Informatik unterschiedlich verwendet, allerdings ist die hier beschriebene Verwendung sehr häufig anzutreffen.

das heißt, Aussagen über die Existenz und Beschaffenheit eines Strukturvorschlages maximaler Fitness werden möglich.

Codierungen bei Evolutionsstrategien

In Rahmen von Evolutionsstrategien wird ein Individuum durch eine Menge von individuellen OBJEKTPARAMETERN mit zugewiesenen Zielfunktionswerten codiert. Realisiert wird diese Codierung der Strukturvorschläge durch einen VEKTOR, bei dem Nachkommastellen zugelassen sind. Vektoren sind mathematische Darstellungsformen für angeordnete Werte und werden in folgender Notation dargestellt.

Abbildung 15: Notationsbeispiel eines zufälligen Vektors

$$\begin{pmatrix} 2{,}5 \\ 1 \\ 0 \\ 3{,}4 \end{pmatrix}$$

Die Verwendung von Nachkommastellen ermöglicht unendlich viele verschiedene Eingabemöglichkeiten, etwa 2,1 oder 2,11 oder 2,111 und so fort. Im Gegensatz dazu codierten etwa die Bitfolgen genetischer Algorithmen typischerweise eine endliche Zahl von möglichen Zuständen.

Evolutionäre Operatoren und Selektion bei Evolutionsstrategien

Wie die genetische Programmierung versuchen auch Evolutionsstrategien eine komplexe oder undurchsichtige Codierung zu vermeiden und sprechen nicht von einer Unterscheidung zwischen einem Genotyp und einem Phänotyp der Strukturvorschläge. Allerdings verwenden auch Evolutionsstrategien Begriffe der Biologie, um dem Verständnis der Algorithmen zuträgliche Assoziationen zu erzeugen. Evolutionsstrategien sprechen von selbstadaptiven[21] und fixen STRATEGIEPARAMETERN, die im selbstadaptiven Fall als ENDOGEN und anderenfalls EXOGEN bezeichnet werden. Die Unterscheidung soll andeuten, dass exogene Strategieparameter dem Kontext der Aufgabe entstammen, die vom jeweiligen MLA bearbeitet werden soll, und

21 ›Selbstadaptiv‹ ist dabei synonym zu der Rede von Autoadaptivität in dieser Arbeit zu verstehen.

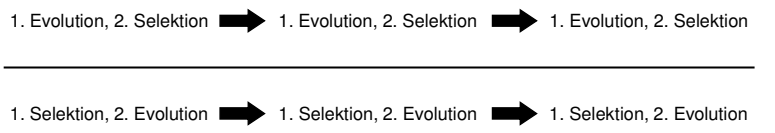
nicht evolviert werden können. Die Strategieparameter beziehen sich auf die Strategie bei dem Einsatz eines evolutionären Operators und sind im Einzelnen:

- Die Größe der zu evolvierenden Population
- Die Anzahl der Elternindividuen, die MULTI-REKOMBINATIV Nachkommen generieren
- Die Anzahl der von jeder Gruppe von Eltern erzeugten Nachkommen
- Die Entscheidung, ob Elternindividuen bei der nächsten Selektion mitberücksichtigt werden

Die Anzahl der Elternindividuen wird als MISCHUNGSZAHL bezeichnet und liegt beim Menschen bei zwei Eltern. Evolutionsstrategien verwenden ebenso wie das übrige evolutionäre Lernen Mutation und Rekombination als evolutionäre Operatoren, denn wenn die Mischungszahl eins beträgt, entspricht dies einer Mutation, da genau ein ELTER evolviert wird und eine festzulegende Anzahl Nachkommen generiert wird. Durch die Formalisierung des Evolutionsschrittes ist ein großes Spektrum von evolutionären Operatoren darstellbar, die prinzipiell auch bei anderen Formen evolutionären Lernens Verwendung finden könnten. Hierin findet sich auch die Motivation der Rede von evolutionären Operatoren, die bisher nur Mutationen und Rekombinationen unter einem Begriff zusammenfassen konnte. Jetzt sind Mutationen nur spezielle PARAMETRISIERUNGEN allgemeiner Strategieparameter. Die zentrale Rolle der Suche nach dem Mutationsschrittweitenband führt bei Evolutionsstrategien im Vergleich zu anderen Formen evolutionären Lernens zu einem Schwerpunkt auf der Mutation gegenüber der Rekombination. Dieser Schwerpunkt ist relativ, da die Mutation häufig deutlich mehr Relevanz erhält als bei anderen Formen evolutionären Lernens, allerdings dadurch innerhalb der Evolutionsstrategie nicht zwangsläufig häufiger eingesetzt wird als die Rekombination. Insbesondere gibt es sehr viel mehr Möglichkeiten zur Rekombination als zur Mutation, da der entsprechende Strategieparameter nur in genau einem Fall eine Mutation erzeugt und in allen anderen Fällen Rekombinationen erstellt – auch und gerade bei sonst identischen Strategieparametern. Ein Ausdruck der Betonung von Mutationen ist, dass Evolutionsstrategien mitunter systematisch erst Rekombinationen durchführen und Mutationen dadurch auf einer größeren Menge von Individuen durchgeführt werden können.

Evolutionsstrategien bestimmen durch eine Parameterwahl, ob Elternindividuen in der Selektion berücksichtigt werden sollen, dadurch entsteht eine Verbindung des Evolutionsschrittes mit dem Selektionsschritt. Die Selektion findet als abschließender Teil des Evolutionsschrittes statt und die Wahl, welche Individuen der letzten Generation evolviert werden, wird vor Durchführung der Selektion getroffen. Andere Formen evolutionären Lernens trennen den Evolutionsschritt und den Selektionsschritt strikt. Lernstrategien, bei denen nur diejenigen Individuen weiter betrachtet werden, die eine hohe Fitness aufweisen, können so gedacht werden, dass sie jeden Autoadaptionszyklus mit dem Selektionsschritt beginnen. Diese Denkweise bestärkt die Wahrnehmung, dass zuerst die Fitness der letzten Generation bewertet wird, bevor festgelegt wird, wie die neue Generation sich zusammensetzt, dass also Evolutionsschritt und Selektionsschritt strikt getrennt sind. Diese unterschiedlichen Interpretationen lassen sich wie folgt visualisieren.

Abbildung 16: Unterschiedliche Autoadaptionszyklen



Schon bei der Definition evolutionären Lernens deuteten sich diese beiden Interpretationsmöglichkeiten an, dort wurde eine einmalige Initialisierung durchgeführt, die als eine erste Selektion interpretiert werden kann.

Problembehandlungen und Weiterentwicklungen evolutionären Lernens

Zur Vermeidung von Überanpassung können innerhalb eines Strukturvorschlages analog zum Stutzen von Entscheidungsbäumen zufällig gewählte Einschränkungen bezüglich eines Attributes getilgt oder gleich alle Anforderungen an ein bestimmtes Attribut aus dem Strukturvorschlag entfernt werden.

Evolutionäres Lernen tendiert zu CROWDING, womit das Auftreten einer Gruppe von Individuen bezeichnet wird, die sich untereinander sehr ähneln und ein höheres Maß an Fitness aufweisen als die übrige Population. Crowding ist selbstverstärkend, da in den nächsten Evolutionsschritten wiederum

sehr viele Mutationen und Rekombinationen der fittesten Individuen entstehen. Eine Maßnahme zur Vermeidung dieses Phänomens ist der Einsatz von Turnier- oder Rouletteselektion, eine andere Maßnahme besteht darin zu fordern, dass jedes Individuum nur einmal als Elter zum Einsatz kommt oder dass die Anzahl an Nachkommen pro Individuum begrenzt wird. Weiterhin kann die gemeinsame Elternschaft von sehr ähnlichen Individuen verboten oder erzwungen werden um Crowding zu erzeugen oder zu vermeiden.

Eine dritte typische Weiterentwicklung evolutionären Lernens besteht in der Nutzung des BALDWIN EFFEKTS. Der Baldwin Effekt beschreibt, dass Individuen in einer sich verändernden Umwelt einen evolutionären Vorteil besitzen, falls sie in der Lage sind, unabhängig von der Entwicklung der Population, zu der sie gehören, lernen zu können, das heißt in der Lage sind, mittels lokaler Autoadaptionsprozesse individuell ihre Fitness zu erhöhen. Individuen, denen individuelles Lernen erlaubt ist, müssen weniger gut an spezifische Situationen angepasst sein. Die Nutzung dieses Effektes kann im Rahmen der Wahl der Strategieparameter realisiert werden, indem evolutionäre Operatoren gezielt auf Teilpopulationen angewendet werden, die etwa bezüglich der Fitnessfunktion eine gemeinsame Schwäche zeigen. Wenn Schwächen von Teilpopulationen auf diese Weise gezielt reduziert werden, kann die Population insgesamt gegebenenfalls in größeren Schritten mutieren und das Mutationsschrittweitenband erweitert sich oder kann sogar hin zu größeren Schrittweiten verschoben werden.

Viele Ideen und Problemlösungen des evolutionären Lernens sind aus der Biologie motiviert und an real existierende Phänomene angelehnt. Das soll nicht darüber hinwegtäuschen, dass die Vorgehensweise häufig diejenige einer mathematischen Optimierung ist und der Zufallsfaktor und die Ziellosigkeit der natürlichen Evolution keine unmittelbare Entsprechung haben. Wichtig für die Diskussion des zweiten Hauptteils wird aber sein, dass innerhalb der Informatik im Zusammenhang mit maschinellem Lernen Maßnahmen entwickelt werden, mit Hilfe derer solch eine zufällige Ziellosigkeit nachempfunden und in MLA erzeugt werden kann.

2.3.4 Lernen von künstlichen neuronalen Netzen – KNN

Motivation

KÜNSTLICHE NEURONALE NETZE – kurz KNN – bilden ein Feld des maschinellen Lernens, das sich für den Entwurf eines Autoadaptionsprozesses die Prozesse und Strukturen innerhalb des menschlichen Gehirns zum Vorbild genommen hat. Analog zu der Motivation des evolutionären Lernens bezieht sich auch bei KNN die Motivation auf vereinfachende Aussagen zur Funktionsweise des Gehirns. Ein Beispiel für solch eine Aussage ist die 100-Schritt-Regel.

> »Ein Mensch kann einen ihm bekannten Gegenstand oder eine bekannte Person innerhalb von 0,1 Sekunden erkennen. Dabei sind bei einer angenommenen Verarbeitungszeit einer Nervenzelle von 1 Millisekunde maximal 100 sequentielle Verarbeitungsschritte im Gehirn des Menschen nötig.«
>
> (Wikipedia Contributors 2012, 100-Schritt-Regel)

Diese Leistungsparameter werden von technischen Systemen zur Objekterkennung noch nicht erreicht. Das Gehirn scheint im Gegensatz zu modernen Rechnern zu einer massiven und funktionellen PARALLELVERARBEITUNG[22] in der Lage zu sein.

Der Fokus beim Versuch der Übertragung dieser Fähigkeit auf MLA liegt nicht auf einer präzisen Modellierung der extrem komplizierten biochemischen Vorgänge des Gehirns. Stattdessen wird die Betrachtung auf einen verhältnismäßig gut verstandenen Teilbereich dieser Prozesse beschränkt, auf die Funktionsweise spezieller NERVENZELLEN im Gehirn, der NEURONEN. Künstliche neuronale Netze basieren, genau wie der Name es andeutet, auf der Betrachtung von vernetzten künstlichen Neuronen. Weitere Eigenschaften des Gehirns, wie eine hohe Parallelität von Prozessen, werden von KNN genau dann genutzt, wenn diese sich gut in den Metho-

22 Parallelität bezieht sich hierbei auf die KONNEKTIONISTISCHE Idee der Darstellung eines Systems durch die massive Parallelisierung der Arbeitsschritte einfacher, vernetzter Einheiten und die damit verbundenen Möglichkeiten zu VERTEILTEN BERECHNUNGEN (Wikipedia Contributors 2012, Konnektionismus).

den der Informatik nutzen lassen. Andere Eigenschaften wie hormonelle Abhängigkeiten werden nicht betrachtet.

Kurzgefasst liegt die Motivation für die Nutzung von KNN in der Hoffnung, die Rolle der Neuronen für die biologischen Vorgänge im Gehirn imitieren und mittels künstlicher Neuronen die Stärken neuronaler Strukturen auf MLA übertragen zu können. Autoadaptionsprozesse und die resultierenden Strukturvorschläge auf Basis von KNN können auch tatsächlich trotz des stark vereinfachenden Vorgehens erfolgreich einige Stärken des menschlichen Gehirns reproduzieren, etwa eine große Unempfindlichkeit gegenüber verfälschten und unvollständigen Eingaben.

Abgrenzung zu biologischen Gehirnen

Die hier vorgenommenen Charakterisierungen maschineller Lernstrategien bewegen sich auf der Betrachtungsebene konzeptioneller Ideen und basieren auf der Betrachtung der Ideen und Motivationen hinter der Entwicklung lernender Algorithmen. Die Entwicklung eines interdisziplinären Verständnisses der Lernstrategien und damit des maschinellen Lernens als Technikbereich hängt dementsprechend stark davon ab, dass die Grenzen der zugrunde liegenden Metaphern klar dargestellt werden. Dies stellt beim evolutionären Lernen ein relativ kleines Problem dar, weil die verkürzte Verwendung von Begriffen aus dem Kontext der Evolution ein häufig anzutreffendes Phänomen darstellt und die Begriffe automatisch mit einer gewissen Skepsis betrachtet werden. Künstliche neuronale Netze erfordern dieselbe Form von Skepsis und um dies zu begründen folgt ein kurzer Abriss eines wesentlichen Standpunktes innerhalb der Neuroanatomie bezüglich der Lernvorgänge im menschlichen Gehirn. Das Ziel hierbei ist, eine neutrale Betrachtung künstlicher neuronaler Netze zu ermöglichen und nicht eine Diskussion der Funktionsweise eines biologischen Gehirns vorzubereiten. Aus Sicht der Neuroanatomie ist einer der typischsten Fehler bei der Rede über das menschliche Gehirn, dass dessen Funktionsweise als mit der eines KNN vergleichbar verstanden wird (Teuchert-Noodt 2011). Diese These soll im Weiteren, aufbauend auf dem Standpunkt von Teuchert-Noodt, kurz begründet werden.

Die erste falsche Grundannahme liegt für Teuchert-Noodt darin, anzunehmen, dass Menschen als Kleinkinder in einer Art von NULLZUSTAND ihr lebenslanges Lernen beginnen und dass Menschen Zusammenhänge ken-

nenlernen, um sie dann durch eine Form von Wiederholung zu erlernen. In diesem Zusammenhang wird häufig die HEBB'SCHE LERNREGEL »what fires together, wires together« genannt. Diese Aussage ist inhaltlich richtig, wird aber für das Verständnis von Lernprozessen deutlich überbewertet. Das menschliche Gehirn ist sehr stark vorstrukturiert. Selbst die Reihenfolge von möglichen Inhalten, die vom menschlichen Gehirn in den einzelnen Altersstufen der kindlichen Entwicklung erlernt werden können, ist stark vorgegeben.

Eine zweite falsche Intuition, die durch den Vergleich mit KNN vermittelt wird, ist diejenige, dass verstanden wird oder modellierbar ist, wie das menschliche Gehirn arbeitet. Dies ist schlicht noch nicht der Fall und die vorliegenden Erkenntnisse sind nicht ohne größere und noch ausstehende Anstrengungen in andere Wissenschaftsbereiche wie die Informatik zu übertragen. Im Gehirn liegen mit Neuronen und GLIAZELLEN mindestens zwei unterschiedliche Formen von Nervenzellen vor, die eine zentrale Rolle spielen und in komplexen Abhängigkeiten zueinander stehen (Wikipedia Contributors 2012, Neuronales Netz). Die Funktionsweise dieser Gliazellen hat in der Arbeit mit KNN keine Entsprechung. Weiterhin sind die chemischen beziehungsweise hormonellen Abhängigkeiten innerhalb des Gehirns noch weitgehend unverstanden und insbesondere noch nicht mit Mitteln der Informatik modellierbar. Ein Grund für die Fokussierung auf Neuronen in der Diskussion der Funktionsweise des Gehirns könnte aus Sicht Teuchert-Noodts darin liegen, dass die Entdeckung von Nervenzellen in der Form von Neuronen zu genau prognostiziert wurde. Es lagen konkurrierende Theorien über die prinzipiellen Abläufe im Gehirn vor und eine dieser Theorien postulierte das Bestehen einer Zelle, die aufgebaut sein sollte, wie ein Neuron tatsächlich aufgebaut ist. Entsprechend war nach dem experimentellen Nachweis beziehungsweise der Entdeckung der Neuronen mittels verbesserter Technik die Überzeugung groß, dass die Theorie, die diese Zellen prognostiziert hatte, genau zutreffend sei. Dieser Erfolg hat die große Popularität der Neuronen mitbegründet und eine übergroße Emphase der Bedeutung dieser Art der Nervenzelle begünstigt, da Neuronen sofort intensiv untersucht werden konnten und inzwischen einen vergleichsweise gut dokumentierten Baustein des menschlichen Gehirns darstellen.

KNN sind ohne umfangreiche Weiterentwicklungen nicht dazu geeignet Gehirne beziehungsweise Gehirnprozesse zu simulieren. Zwar eignen sich KNN für den Einsatz in Kontexten, in denen ein Überfluss an Sensordaten

vorliegt und bei denen wenig Vorwissen besteht, aber das Gehirn ist schlicht zu komplex, um es ohne massive Vorstrukturierungen der einzusetzenden KNN funktionell anzunähern. Vereinfachende Theorien wie die Lokalisierung, die es erlauben würden, ein Gehirn mittels lokaler Modelle anzunähern, scheinen nicht einsetzbar zu sein. Zwar gibt es auch Erfolge im Versuch, Hirnprozesse mit Hilfe der Informatik zu verstehen, etwa bei der Diagnose von posttraumatischen Belastungsstörungen bei Soldaten (Hayes et al. 2011). Allerdings basieren diese Methoden, unabhängig davon, wie viel Potenzial ihnen überhaupt zugebilligt werden kann[23], nur sehr nachrangig auf maschinellem Lernen oder speziell auf KNN. Der Hauptvorteil des Einsatzes von KNN als MLA liegt – wie noch diskutiert wird – darin, dass KNN ohne umfangreiche Vorstrukturierung durch die Entwickler oder Nutzer eingesetzt werden können. Eine Simulation von Gehirnprozessen kann auf dieser Stärke nicht aufbauen.

Insgesamt sind künstliche Neuronen mathematische Modelle, die auf der Funktionsweise von natürlichen Neuronen basieren, aber so abgewandelt wurden, dass sie sich gut in der Informatik und dort im maschinellen Lernen einsetzen lassen. Zwar lassen sich einige Stärken und Schwächen von neuronalen Netzen auch bei künstlichen neuronalen Netzen beobachten, allerdings geben die neuronalen Netze nur einen Indikator dafür ab, in welchen Bereichen der KNN eine genauere Analyse gegebenenfalls einen Mehrwert ergeben würde.

Exemplarische Einsatzgebiete von KNN

Einer der bekanntesten und erfolgreichsten Einsätze künstlicher neuronaler Netze ist die Entwicklung von Algorithmen, die in der Lage sind, Backgammon zu spielen. Backgammon ist ein würfelbasiertes 2-Personen-Brettspiel, bei dem nach jedem Würfelwurf für den aktuellen Spieler eine begrenzte Anzahl von Zügen möglich ist. KNN können im Backgammon sehr erfolgreich als Lernstrategie eingesetzt werden.

> »[KNN] excel at strategic and positional judgment, using their knowledge to make fine distinctions between plays. They are less

23 Eine Darstellung der engen Grenzen der Beobachtbarkeit von Hirnprozessen gibt Hasler (Hasler 2011, S. 39ff).

> skilled in ›technical" positions, such as bearing in against an anchor, which humans solve by calculation of the probabilities.
> This is the opposite of the situation in many other games, in which computers calculate tactics well but fall short in strategic understanding. As a result there's been a lot of interest in applying temporal difference learning with neural nets to other games, and to mundane tasks too.«
>
> (Scott 2001)

Die Aussage von Scott zeigt eine für Algorithmen kontraintuitive Schwäche: sie sind menschlichen Spielern in berechenbaren Situationen unterlegen. Auch beim Backgammon treten solche Situationen auf, etwa im Endspiel, in dem die Spieler häufig keine Möglichkeit mehr haben die Spielsteine des Gegners zu bedrohen oder dessen Zugplanung zu beeinflussen. KNN sind schlechter als andere MLA für Kontexte geeignet, in denen Vorwissen wie die Wahrscheinlichkeiten von Würfelergebnissen dem MLA mittels einer Vorstrukturierung vorgegeben werden können. KNN können in solchen Stellungen nicht analytisch die Würfelwahrscheinlichkeiten errechnen und einen optimalen Zug identifizieren. Eine in diesen Situationen von menschlichen Spielern umgesetzte, stellungsbezogene Spielweise beruht auf klaren Berechnungen oder zumindest mathematischen Abschätzungen der Nützlichkeit eines Zuges, basierend auf mathematischem Vorwissen. Dennoch sind KNN sehr erfolgreich im Backgammon eingesetzt worden, da dort die präzise Bewertung der aktuellen Spielposition – der STELLUNG – aufgrund fehlenden theoretischen Wissens vergleichsweise schwierig ist. Unabhängig von der Frage des fehlenden Vorwissens ist es bei der Nutzung eines KNN besonders gut möglich, das MLA im Rahmen des Autoadaptionsvorgangs gegen eine Kopie von sich selbst spielen zu lassen, ohne dass der Lernvorgang darunter leidet. Auf diese Weise können sehr viele simulierte Spiele in sehr kurzer Zeit durchgeführt werden und der Zufallseffekt des Würfelwurfes wirkt dabei einer Überanpassung entgegen. Der Autoadaptionsprozess benötigt daher im Prinzip keine Trainingsdaten und kann größtenteils automatisch ablaufen. Die Eingabedaten eines MLA sind in diesem Fall die Ergebnisse der Würfelwürfe und die Züge des anderen MLA. Insgesamt bewegt sich die Spielstärke der besten künstlichen neuronalen Netze im Backgammon auf Weltklasseniveau und einige der entsprechenden Programme sind kostenlos im Internet verfügbar. Die ver-

fügbaren KNN sind dann bereits mit mehreren Millionen Partien trainiert worden und stellen schon den aus dem Autoadaptionsprozess entstandenen Strukturvorschlag dar, wodurch sie auch auf rechenschwachen Computern eingesetzt werden können[24].

Ein zweites Anwendungsbeispiel kommt aus der Bilderkennung (Russell et al. 2007, S. 914ff) und soll dazu dienen die Leistungsfähigkeit von KNN zu illustrieren. In dieser Anwendung wurde eine Datenbank aus 60.000 handschriftlichen Ziffernproben zugrunde gelegt und die Aufgabe bestand darin die Ziffern zu erkennen. KNN konnten ihre Fehlerrate im Laufe einiger Weiterentwicklungen von 1,6% über 0,9% auf schließlich 0,7% verbessern. Hier wurde sehr viel menschliches Vorwissen in den Autoadaptionsprozess eingebracht und die KNN wurden gezielt für den spezifischen Kontext vorstrukturiert. Die Fehlerrate eines Menschen bei der Ziffernerkennung liegt im genannten Beispiel geschätzt bei 0,2%, allerdings wurde für eine vergleichbare Datenbasis des United States Postal Service heuristisch eine Fehlerrate von 2,5% für den Menschen ermittelt. In jedem Fall geben die Größenordnungen der Fehlerraten ein sehr hilfreiches Gefühl für die Leistungsfähigkeit von KNN[25].

Funktionsbeschreibung künstlicher neuronaler Netze

Die Teilfunktionalität des Gehirns, die in KNN nachgebildet werden soll, wird, wie bereits angedeutet, durch die erwähnte Hebb'sche Lernregel beschrieben. Diese 1946 vom Psychologen Donald Hebb aufgestellte These beschäftigt sich mit Strukturen verbundener Neuronen, das heißt, mit neuronalen Netzen, und besagt sinngemäß, dass ein neuronales Netz lernt, indem bei gleichzeitiger Reizung zweier Neuronen die Stärke ihrer Verbindung vergrößert wird. Die relativ bekannte Kurzfassung dieser Lernregel ist die Formulierung »what fires together, wires together«. Der Hauptgrund,

24 Eine gute lesbare Einführung in die Konstruktion eines Backgammon-Programms auf Basis von KNN – inklusive einer nützlichen Visualisierung der Grenzen der Leistungssteigerung durch immer weitere Trainingspartien – findet sich bei Tsinteris (Tsinteris 2012).

25 Einige gut verständliche, Animationen der Vorgehensweise beziehungsweise der Fähigkeiten von KNN im Hinblick auf Zahlenerkennung finden sich bei LeCun (LeCun 2011).

aus dem diese Lernregel als Grundlage einer Lernstrategie eingesetzt wird, findet sich in der folgenden Perspektive.

> »Viele der Modelle, die diskutiert wurden, beschäftigen sich mit der Frage, welche logische Struktur ein System besitzen muss, um eine Eigenschaft X darzustellen… Ein alternativer Weg, auf diese Frage zu schauen, ist folgender: Was für ein System kann die Eigenschaft X (im Sinne einer Evolution) hervorbringen? Ich glaube, wir können in einer Zahl von interessanten Fällen zeigen, dass die zweite Frage gelöst werden kann, ohne die Antwort zur ersten zu kennen. (Rosenblatt 1962)«
>
> (Görz et al. 2003, S. 11)

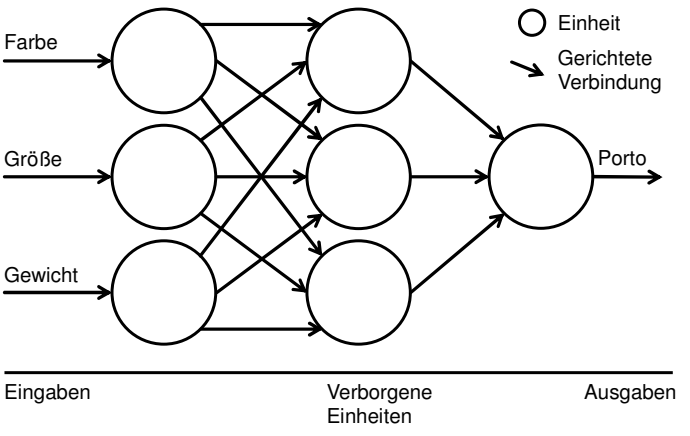
KNN können mit Hilfe eines Autoadaptionsprozesses, basierend auf künstlichen Neuronen und der Hebb'schen Lernregel, erwünschte Eigenschaften hervorbringen, ohne dass im Vorhinein bekannt ist, welche Struktur für die Realisierung dieser Eigenschaft notwendig ist. Daraus folgt sofort, dass KNN ihre Struktur in noch größerem Umfang als andere MLA autoadaptiv anpassen müssen. Die entstehenden Netze stellen Strukturvorschläge dar, denen keine geschickte Codierung des Kontextes und kein formales Vorwissen zugrunde liegen, sondern die als Gesamtstruktur entsprechend der Hebb'schen Lernregel systematisch auf Eingaben reagieren. KNN kommen damit der ursprünglichen in der Einleitung dargestellten Idee eines assoziativ lernenden Algorithmus sehr nahe.

Für den Einsatz von KNN als Lernstrategie ergibt sich daraus, dass zur Durchführung des Lernvorgangs – analog zur Verwendung von evolutionärem Lernen– nur sehr wenige Parameter oder gar analytische Hintergrundinformationen identifiziert oder quantifiziert werden müssen. Lediglich die Auswahl der relevanten Eingabegrößen ist notwendig, und im Gegensatz zu evolutionärem Lernen muss darüber hinaus kein Aufwand in die Erstellung einer Codierung investiert werden. Gerade weil dem Autoadaptionsprozess eines KNN keine interpretierbare Codierung zugrunde liegt, können aus dem Strukturvorschlag eines KNN nicht ohne Weiteres die Faktoren, die zu diesem Ergebnis geführt haben, abgelesen werden. Die Autoadaption eines KNN resultiert in der Aneignung einer Fähigkeit und nicht in der Darstellung, wie diese Fähigkeit erlernt werden kann.

Die Komponenten künstlicher neuronaler Netze

Ein künstliches neuronales Netz setzt sich aus EINHEITEN genannten Knoten und GERICHTETEN VERBINDUNGEN zwischen diesen Knoten zusammen. Die Einheiten sind nicht notwendigerweise mit allen anderen Einheiten verbunden, insbesondere liegt nicht immer eine Verbindung in beide Richtungen vor. Die Daten, die ein KNN erhält und weiterverarbeitet, werden SIGNALE genannt. Dieser Begriffsbildung soll hier gefolgt werden, um die Intuition einer potenziellen Inhaltslosigkeit und der Konzeptlosigkeit eines Signals zu stärken. Ein KNN ist in der Lage Eingabesignale aufzunehmen, sie zwischen den Einheiten weiterzuleiten, dabei zu modifizieren und schließlich Ausgabesignale zu erzeugen. Die Stärke einer gerichteten Verbindung zwischen zwei Knoten wird deren GEWICHT genannt. Der Autoadaptionsprozess eines KNN entspricht der Adaption der Gewichte der Verbindungen zwischen Einheiten, wobei sich prinzipiell jedes Gewicht in jedem Adaptionsschritt ändern kann und jede solche Änderung die Reaktionsmuster des gesamten KNN beeinflussen kann. KNN verzichten im Rahmen des Autoadaptionsprozesses auf die Manipulation interpretierbarer Symbole. Strukturvorschläge setzen sich nicht aus codierten Regeln zusammen, sondern aus einer Anordnung von Knoten, Verbindungen und Verbindungsgewichten. Die nachfolgende Abbildung zeigt eine exemplarische Visualisierung einer solchen Anordnung mit drei Eingabesignalen A, B und C, sowie einem Ausgabesignal D.

Abbildung 17: Vollständig verbundenes KNN



Als erstes soll die Rolle der Einheiten genauer betrachtet werden. Diese fungieren als KÜNSTLICHE NEURONEN, das heißt, sie orientieren sich in ihrer Funktionsweise an den Neuronen im menschlichen Gehirn.

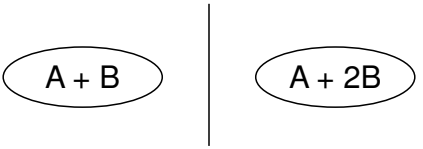
> »Die ›Schaltungstechnik‹ von Neuronen kennt üblicherweise mehrere Eingangsverbindungen sowie eine Ausgangsverbindung. Wenn die Summe der Eingangsreize einen gewissen Schwellenwert überschreitet […] ›feuert‹ das Neuron […] das Ausgangssignal des Neurons.«
>
> (Wikipedia Contributors 2012, Neuronales Netz)

KNN konstruieren künstliche NEURONEN, indem sie Einheiten einsetzen, die in der Lage sind, aus einem oder mehreren Eingabesignalen ein oder mitunter auch mehrere Ausgabesignale zu erzeugen. Die Erzeugung eines Ausgabesignals in einer Einheit wird entsprechend als AKTIVIERUNG dieser Einheit bezeichnet. Aktivierungen können sowohl der Signalweiterleitung an eine andere Einheit als auch der Ausgabe an den Nutzer des KNN dienen. Einheiten, deren Aktivierungsfunktion ein Eingabesignal des Nutzers aufnimmt oder ein Ausgabesignal an den Nutzer abgibt, werden als EINGABE- respektive AUSGABEEINHEITEN bezeichnet. Die verbliebenen Einheiten werden unter dem Begriff VERBORGENE EINHEITEN zusammengefasst. Im obigen Beispiel sind in der mittleren Spalte drei verborgene Einheiten zu sehen.

Die Systematik, nach der eine Einheit Aktivierungen vornimmt, wird als AKTIVIERUNGSFUNKTION bezeichnet. Eine Aktivierungsfunktion muss nur eine Anforderung erfüllen, sie muss die Entscheidung über die Aktivierung der zugeordneten Einheit nach einem systematischen Kriterium treffen, das die jeweiligen Eingabesignale berücksichtigt. Unterschiedliche Einheiten können individuelle Aktivierungsfunktionen aufweisen, und eine Anpassung dieser Funktionen im Rahmen des Autoadaptionsprozesses eines KNN ist zwar unüblich, kann aber durchaus vorgenommen werden. Eine einfache Aktivierungsfunktion besteht darin, die Stärke der Eingabesignale zu summieren und die jeweilige Einheit zu aktivieren, wenn diese Summe einen gewissen SCHWELLENWERT überschreitet. Eine leichte Weiterentwicklung dieser Aktivierungsfunktion besteht darin, die Eingabesignale nicht länger gleichberechtigt zu summieren, sondern jedes Signal in seiner Wichtigkeit für die Summe einzuschätzen und mit einem GEWICHT

zu belegen. Im der folgenden Visualisierung wurde einmal eine normale Summe gebildet und in zweiten Fall dem Signal B ein doppelt so großes Gewicht beigemessen wie dem Signal A.

Abbildung 18: Möglichkeiten einer Signalgewichtung



Veränderungen in der Signalstärke von Signal B haben durch die Wahl dieses Gewichtes einen doppelt so großen Einfluss auf die Erreichung des Schwellenwertes wie Veränderungen von A. Jeder gerichteten Verbindung eines KNN, außer den Ausgabesignalen an die Nutzer, ist eine Gewichtung zugeordnet. Dieses VERBINDUNGSGEWICHT stellt den Einfluss des transportierten Signals auf die angesteuerte Einheit dar. Die Anzahl der Einheiten und die Definition der Aktivierungsfunktionen, sowie die Anzahl und Orientierung der gerichteten Verbindungen eines KNN werden meist vor Beginn des Autoadaptionsprozesses fixiert. Die Adaptivität eines KNN liegt in diesem Fall ausschließlich in der Wahl der Verbindungsgewichte, das heißt, die Verbindungsgewichte stellen die manipulierbaren Parameter des KNN dar. Der Raum aller möglichen Verbindungsgewichte stellt für die KNN den Suchraum des Autoadaptionsprozesses dar und wird als GEWICHTUNGSRAUM bezeichnet. Die Verbindungsgewichte ändern sich während des Autoadaptionsprozesses mit der Betrachtung jedes Eingabedatums. Ein KNN kann mit zufälligen Gewichten initialisiert werden, optional kann jedoch auch analytisches Hintergrundwissen zu Abhängigkeiten zwischen den Attributen der Eingabedaten bei der Initialisierung der Gewichte berücksichtigt werden. Einerseits kann, wie im Beispiel der Ziffernerkennung, die Leistung eines KNN verbessert werden, wenn Hintergrundwissen eingesetzt wird, andererseits besteht der wesentliche Punkt gerade darin, dass diese Möglichkeit optional ist. Die Eingabedaten eines KNN können in nahezu jeder Art und Weise übergeben werden, es ist nicht erforderlich eine konsistente oder alle Teilaspekte erfassende Codierung zu erstellen. Die Eingabedaten dürfen und werden in der Praxis nicht direkt vergleichbar, fragmentarisch und mitunter sogar widersprüchlich sein. Das Lernen mittels künstlicher neuronaler Netze ist dementsprechend weitgehend unemp-

findlich auch gegenüber starkem Rauschen. Es genügt prinzipiell all diejenigen Aspekte der Trainingsdaten, die als potenziell relevant für den zu erstellenden Strukturvorschlag eingestuft werden, zu erfassen und dem KNN zu übergeben.

Der Autoadaptionsprozess bei KNN

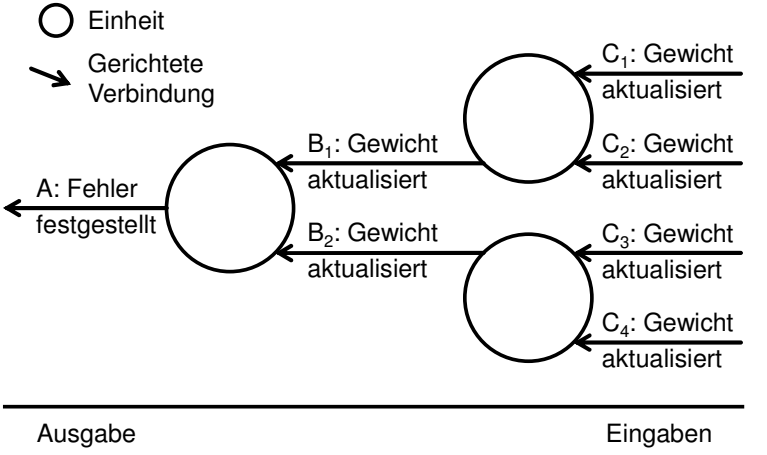
Der Autoadaptionsprozess bei KNN durchläuft Zyklen, die als EPOCHEN bezeichnet werden. Innerhalb einer Epoche besteht der Prozess aus den folgenden Schritten.

A. Die Eingabesignale werden von den Eingabeeinheiten aufgenommen. Die Eingabeeinheiten initiieren einen durch die Aktivierungsfunktionen und die Gewichte der beteiligten Einheiten geleiteten SIGNALFLUSS.
B. Der Signalfluss setzt sich entlang der gerichteten Verbindungen durch das KNN fort.
C. Die Ausgabeeinheiten geben die resultierenden Signale aus.
D. Eine AKTUALISIERUNGSREGEL tritt in Kraft und nimmt eine Anpassung der Gewichte des KNN vor.

Die Aktivierungsregel kann sehr unterschiedlich ausfallen und sowohl überwachtes als auch unüberwachtes Lernen realisieren. Wenn ein KNN überwachtes Lernen realisieren soll, wird ihm ein Trainingsdatum übergeben und das Ausgabesignal, das das KNN im seinem aktuellen Zustand erzeugt, wird mit dem Wert verglichen, den das Trainingsdatum vorgibt. Die Abweichung wird als FEHLER bezeichnet und die Gewichte des KNN werden so angepasst, dass der Fehler der Ausgabeeinheiten minimal oder zumindest kleiner als zuvor wird. Diese Vorgehensweise birgt die Herausforderung, dass Fehler verborgener Einheiten nicht direkt messbar sind. Zwar können verborgene Einheiten in den meisten KNN beobachtet werden, aber das Verborgene an ihnen ist ihre Funktion beziehungsweise ihre Relevanz für das Gesamtnetz. Die Trainingsdaten machen natürlich keine Vorgaben für den Zustand von verborgenen Einheiten. Ein Ansatz mit diesem Problem umzugehen besteht darin, eine Einheit als für einen Teil der Fehler aller ihr nachfolgenden Einheiten anzusehen. Auf diesem Weg können Fehler von Ausgabeeinheiten auf die mit ihnen verbundenen verborgenen Einheiten übertragen werden, wodurch ein epochenabhängiger Sollwert für die

entsprechende verborgene Einheit erstellt werden kann. Dieser Ansatz zur Analyse verbogener Einheiten wird als BACKPROPAGATION bezeichnet und soll im Weiteren visualisiert und detaillierter dargestellt werden. Ein Verständnis, inwiefern die verborgenen Einheiten eines KNN verborgen sind beziehungsweise analysiert werden können, verlangt zumindest grundlegende interdisziplinäre Kenntnisse bezüglich dieser Aktualisierungsregel. Zur besseren Übersicht wurde das KNN in der Visualisierung gegenüber der vorherigen Abbildung spiegelverkehrt dargestellt, da im ersten Schritt die Ausgabeeinheit betrachtet wird und dann entgegen dem Signalfluss die übrigen Einheiten analysiert werden.

Abbildung 19: Backpropagation als Aktualisierungsregel



Die Aktualisierungsregel zur Adaption der Gewichte verborgener Einheiten basiert auf den folgenden Schritten.

A. Beim Vergleich eines Ausgabesignals mit den Vorgaben eines Trainingsdatums wird ein Fehler festgestellt.

B. Die Gewichte der eingehenden Verbindungen werden so abgeändert, dass der Fehler in der Ausgabeeinheit verkleinert wird.

C. Die Verbindungen, deren Gewichte gerade verändert wurden, entstammen bestimmten Einheiten – im obigen Beispiel den zwei rechten Einheiten. Der Vergleich der gewünschten neuen Ausgabesignale mit den bisherigen Ausgabesignalen definiert einen Fehler. Die Gewichte der

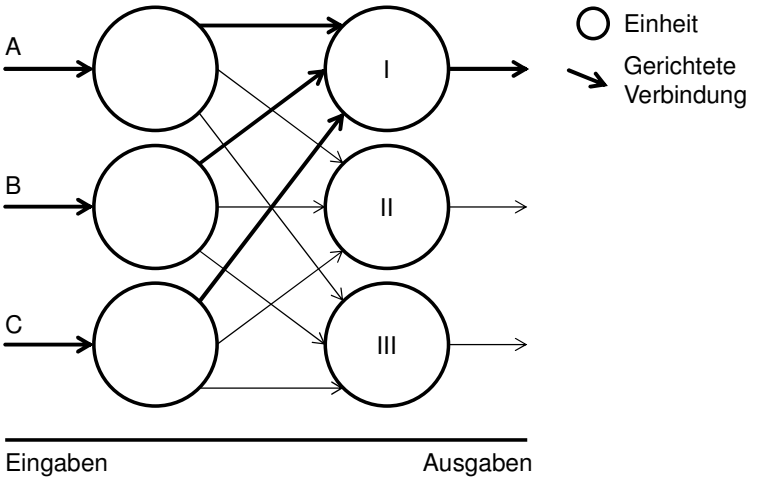
eingehenden Verbindungen werden wiederum so abgeändert, dass dieser Fehler verkleinert wird.

D. Schritt **C** wird wiederholt, bis die Eingabeeinheiten erreicht werden.

Die genannten Schritte können und werden in der Praxis im Rahmen des Einsatzes einer Aktualisierungsregel sehr häufig durchlaufen. Das bedeutet, innerhalb einer Epoche wird die Aktualisierungsregel formal nur einmal angewendet, aber dennoch können sehr viele Gewichtsaktualisierungen vorgenommen werden. Ein Beispiel ist, dass der beschriebene Prozess zur Adaption der Gewichte solange durchlaufen wird, bis die zu korrigierenden Fehler eine festgesetzte Grenze unterschreiten.

Wenn im Zusammenhang mit KNN von unüberwachtem Lernen gesprochen wird, dann wird darunter verstanden, dass das KNN Signale klassifizieren kann, ohne dass die möglichen Klassen im Vorhinein bekannt sind. Ein erstes Beispiel hierfür sind SELBSTORGANISIERENDE MERKMALSKARTEN oder kurz selbstorganisierte Karten. Hierbei wird jede Eingabeeinheit mit allen Nicht-Eingabeeinheiten des KNN verbunden, wie etwa im folgenden Beispiel.

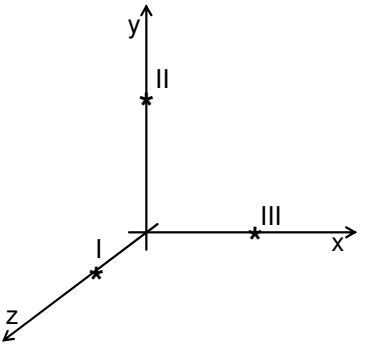
Abbildung 20: Selbstorganisierende Karte aktiviert Kategorie I



Das Konzept einer selbstorganisierenden Karte ist, dass für jedes Eingabesignal diejenigen Einheiten identifiziert werden, deren Verbindungsgewich-

te die größte Ähnlichkeit mit diesem Eingabesignal aufweisen. Im obigen Beispiel entspricht das aktuell betrachtete Trainingsdatum den Eingangssignalen A, B und C und die Verbindungsgewichte der Einheit I sollen am ehesten mit den Eingangssignalen übereinstimmen. Nachdem auf diese Weise eine Einheit identifiziert wurde, besteht die Aktualisierungsregel selbstorganisierender Karten darin, dass die Verbindungsgewichte der beschriebenen Einheit dem Eingabesignal noch weiter angenähert werden. Wenn immer wieder dasselbe Trainingsdatum eingelesen wird, wird irgendwann eine Einheit mit genau den zu den Eingangssignalen passenden Verbindungsgewichten vorliegen. Im obigen Beispiel wurde der Umgang mit drei Signalen dargestellt. Die Verbindungsgewichte einer Einheit können in diesem Beispiel als die Koordinaten eines Punktes im dreidimensionalen Raum interpretiert werden. Im Beispiel lagen drei Einheiten vor, die jeweils drei Verbindungsgewichte aufweisen. Das bedeutet, den Einheiten I, II und III können Punkte im Raum zugewiesen werden. Wenn beispielsweise angenommen wird, dass jede Ausgabeeinheit nur genau ein Verbindungsgewicht ungleich null besitzt, so liegen die drei Einheiten I, II und III in einem Koordinatensystem auf den Achsen[26].

Abbildung 21: Verbindungsgewichte als Werte der Raumachsen

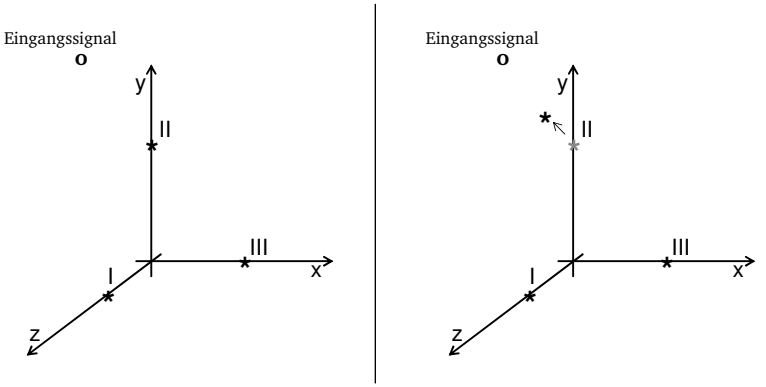


Wenn jetzt ein Eingangssignal an das MLA übergeben wird, können die drei Komponenten **A**, **B** und **C** der Eingabe wiederum als Koordinaten im

26 Zum Beispiel hat die Einheit II eine Ausprägung von null in die Breite und in die Tiefe. Diese Einheit besitzt lediglich für die Höhe einen (in diesem Fall positiven) Wert.

dreidimensionalen Raum interpretiert werden. Die Annäherung der dem Eingabesignal ähnlichsten Ausgabeeinheit könnte beispielsweise wie folgt aussehen.

Abbildung 22: Adaption von Verbindungsgewichten



Die Visualisierung ist etwas irreführend, da sich in der Praxis bei einer selbstorganisierenden Karte die Gewichte sehr vieler Einheiten ›in Richtung‹ des Eingabesignals bewegen, die nächstgelegenen jedoch stärker als die weiter entfernten Einheiten. So bilden sich nach einer gewissen Zeit die Muster, die in den Rohdaten vorliegen, im KNN nach. Die Funktionsweise selbstorganisierter Karten liefert eine sehr gute Intuition, wie ein MLA in der Lage sein kann, nach Durchführung eines völlig ungesteuerten Autoadaptionsprozesses Strukturen vorzuschlagen, die einen Bezug zu den Eingabedaten haben.

Lernen mittels KNN hat generell die Schwäche, dass Eingabesignale, die keine Ähnlichkeit zu den zuvor verarbeiteten Trainingsdaten aufweisen, ein unberechenbares Verhalten des KNN hervorrufen. Die ADAPTIVE RESONANZTHEORIE ist ein Ansatz des unüberwachten Lernens, der dieses Problem zumindest für den Zeitraum des Lernvorgangs behebt. Allerdings reagieren auch die auf diese Weise erzeugten Strukturvorschläge – die Ergebnisse des Autoadaptionsprozesses – nicht systematisch auf neuartige Eingabesignale.

Im Rahmen der adaptiven Resonanztheorie wird analog zu den selbstorganisierenden Karten bei der Aktualisierungsregel die Ähnlichkeit der Eingabesignale mit den Gewichten der Einheiten des KNN festgestellt und

die ähnlichste Einheit dem Eingabesignal noch weiter angenähert. Allerdings werden nur die Gewichte der ähnlichsten Einheit adaptiert, die adaptive Resonanztheorie entspricht diesbezüglich genau dem obigen Beispiel. Über die Anpassung der ähnlichsten Einheit hinaus besitzt die adaptive Resonanztheorie, im Gegensatz zu den selbstorganisierenden Karten, einen zusätzlichen WACHSAMKEITSPARAMETER. Dieser Wachsamkeitsparameter entscheidet, ob die ähnlichste Einheit ausreichend große Übereinstimmungen aufweist oder ob mit dem Eingabesignal etwas komplett Neues vom MLA registriert wurde, für das im KNN noch keine Entsprechung besteht. Wenn die Ähnlichkeit als ausreichend groß betrachtet wird, wird das KNN als IN RESONANZ befindlich bezeichnet. In dem Fall, dass die Ähnlichkeit nicht ausreichend groß ist, wird eine zusätzliche Einheit erzeugt und die Gewichte der zusätzlichen Einheit werden entsprechend dem als neu beurteilten Eingabesignal eingerichtet. Wenn Teile eines KNN sich zu einem Eingabesignal in Resonanz befinden, können Einheiten eindeutig Klassen von Eingabesignalen zugeordnet werden. Das bedeutet, dass die Unterscheidung der Klassen von Eingabesignalen im entstandenen Strukturvorschlag eine strukturelle Entsprechung aufweist. Eine solche LOKALE REPRÄSENTATION hat zwei Stärken: zum einen kann ein solches KNN nach Abschluss des Autoadaptionsvorgangs aufgrund seiner hohen Parallelität sehr schnell ausgewertet werden und zum anderen besitzt es eine hohe Fehlertoleranz gegenüber Ausfällen einzelner Einheiten oder Verbindungen. Eine direkt damit zusammenhängende Schwäche besteht jedoch in dem vergleichsweise großen Zeitaufwand, der nötig ist um die Trainingsdaten zu lernen. Wenn aufgrund von Vorwissen die Anzahl der zu identifizierenden Eingabesignale bekannt ist oder abgeschätzt werden kann, wird in der Praxis häufig ein KNN mit einem entsprechend großen Reservoir an speziell ausgezeichneten Einheiten erzeugt, die nur adaptiert werden dürfen, wenn ein neues Eingabesignal erkannt wurde. Diese Vorstrukturierung hat den Vorteil, dass keine neuen Einheiten erzeugt werden müssen und dennoch eine lokale Repräsentation möglich ist. Die Idee hinter dieser Maßnahme ist auch über die adaptive Resonanztheorie hinweg von Bedeutung, da sie erklärt, wie prinzipiell im Vorfeld Einfluss auf den Aufbau eines KNN genommen werden kann. Darüber hinaus deutet sich hier an, dass und wie ein KNN ohne Vorstrukturierungen oder Steuerung in Reaktion auf die Eingabesignale systematisch wachsen und schrumpfen kann.

DARSTELLUNGSKRAFT von KNN

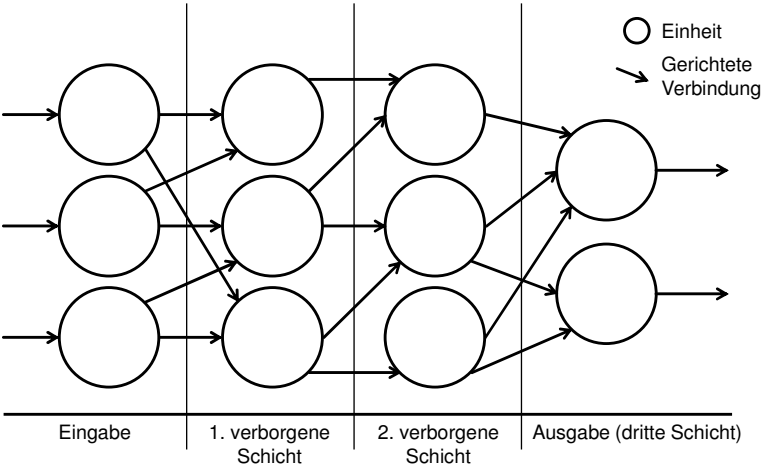
Die Motivation zur Erstellung eines KNN lag unter anderem in der Idee, mittels eines Autoadaptionsprozesses Strukturvorschläge zu erstellen, die gewisse Eigenschaften aufweisen oder Funktionen erfüllen, ohne dass im Vorfeld klar sein muss, welche Voraussetzungen die Strukturvorschläge erfüllen müssen, um ebendies leisten zu können. Zwar verringert eine solche Vorgehensweise das notwendige Vorwissen, sie entbindet jedoch nicht von der Betrachtung, welche Eigenschaften und Funktionen ein KNN prinzipiell ausbilden kann. Kenntnisse über die Potenziale und Grenzen des Autoadaptionsprozesses sind notwendig um KNN konzeptionieren zu können, insbesondere um die Aktivierungsfunktionen der Einheiten festzulegen. Ein Beispiel für die Grenzen des Autoadaptionsprozesses bilden Aktivierungsfunktionen, die – wie im obigen Beispiel – eine Summe mit gewichteten Summanden bilden und prüfen, ob ein Schwellenwert überschritten wurde. Solche Aktivierungsfunktionen erlauben dem entstehenden KNN unabhängig vom übrigen Autoadaptionsprozess nur die Darstellung von mathematisch äußerst einfachen Funktionen und Eigenschaften. Zwar können komplexere Zusammenhänge näherungsweise durch einfachere Funktionen und Eigenschaften beschrieben werden, aber die Frage, welche Funktionen von welchen KNN prinzipiell darstellbar sind, ist dennoch von großer Bedeutung. Zur Beantwortung dieser Frage sollen zunächst zwei Klassen von KNN unterschieden werden: KNN, bei denen keine Rückkopplung erlaubt ist, die AZYKLISCHEN KNN, und diejenigen Netze, bei denen Rückkopplungen zugelassen sind, die REKURRENTEN KNN.

Zuerst soll eine Darstellung der deutlich weniger komplexen azyklischen KNN vorgenommen werden. Die Ausgaben dieser KNN hängen nur von den Eingabesignalen und dem Zustand der Gewichte ab. Sie besitzen neben den Verbindungsgewichten keine veränderlichen Parameter und abgesehen von der zwischenzeitlichen Adaption der Gewichte gemäß der Aktualisierungsregel reagieren sie auf die gleichen Eingabesignale immer auf die gleiche Weise. Jedes azyklische KNN kann formal in SCHICHTEN angeordnet dargestellt werden[27], wobei Einheiten jeder Schicht Signale nur aus

27 Gegebenenfalls müssen dabei für Einheiten, die mit einer entfernten Schicht kommunizieren, in den dazwischenliegenden Schichten Einheiten mit fixen Gewichten eingefügt werden, die lediglich das Signal weiterleiten.

der jeweils vorhergehenden Schicht empfangen und Signale nur in die folgende Schicht senden. Eingabeeinheiten werden dabei nicht als eigene Schicht betrachtet, da sie Eingabesignale ohne Gewichtung aufnehmen, diese aufspalten oder vervielfältigen und an andere Einheiten weiterleiten. Schichten werden deshalb in VERBORGENE SCHICHTEN und die AUSGABESCHICHT unterteilt.

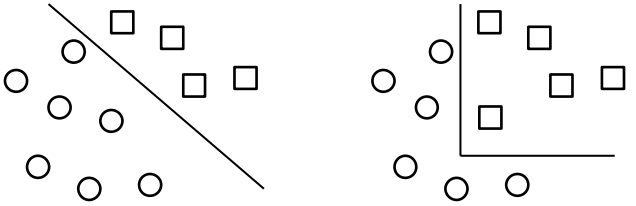
Abbildung 23: Übersicht der Schichten eines dreischichtigen KNN



Die Darstellungskraft von KNN wird beurteilt, indem betrachtet wird, welche mathematischen Funktionen und Operationen die Netze darstellen können, da die Umwandlung von Eingabesignalen in Ausgabesignale formal unabhängig von der konkreten Codierung ein mathematischer Vorgang ist.

Das einfachste azyklische Netz wird als EINLAGIGES PERZEPTRON bezeichnet und besitzt nur eine Schicht, die dadurch gleichzeitig die Ausgabeschicht ist. Ein einlagiges Perzeptron kann den Raum der Eingabedaten LINEAR in zwei Teile teilen, wenn als Aktivierungsfunktion, wie oben beschrieben, eine gewichtete Summe verwendet wird.

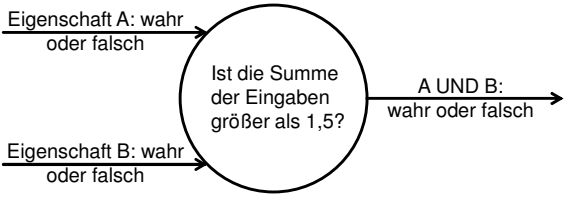
Abbildung 24: Lineare und nicht-lineare Trennung



Ein ZWEILAGIGES PERZEPTRON, das heißt ein KNN mit einer verborgenen Schicht, kann die meisten MATHEMATISCHEN FUNKTIONEN beliebig genau beschreiben[28], wenn etwas komplexere Aktivierungsfunktionen eingesetzt werden. Bereits der Einsatz einer gewichteten Summe als Aktivierungsfunktion erlaubt es einem zweilagigen Perzeptron, rein mittels der Adaption ihrer Gewichte die grundlegenden LOGISCHEN FUNKTIONEN UND, ODER und NICHT abzubilden. Nachfolgend wird zur Anschauung eine Einheit visualisiert, die ein UND abbildet. Die zwei Eingabesignale können dabei das Vorliegen zweier Eigenschaften codieren, wodurch die Einheit überprüft, ob die Eigenschaften gleichzeitig auftreten. Die zwei Eingabesignale werden so codiert, dass sie jeweils entweder eine 0 oder eine 1 übermitteln und die Entscheidung über die Aktivierung der Einheit wird getroffen, indem eine Summe der Eingabesignale mit dem Schwellenwert 1,5 verglichen wird.

28 Mathematische Funktionen beschreiben jegliche Formen von eindeutigen Zuordnungen zwischen einem Eingabe- und genau einem Ausgabewert. Bestimmte, besonders einfache (präzise: beschränkte und stetige) mathematische Funktionen können von KNN aus zwei Schichten beliebig genau beschrieben werden, wobei die verborgene Schicht aus SIGMOIDEN EINHEITEN zusammengesetzt sein muss. Der Grund ist, dass einfache (präzise: stetige) Funktionen als stückweise linear betrachtet werden können und die Zerlegung der anzunähernden Funktion je nach Forderung an die Genauigkeit immer kleiner gewählt werden kann.

Abbildung 25: Ein logisches UND aus Verbindungsgewichten



Die beiden Gewichte der Summe sind dabei auf 1 festgelegt. Eine Senkung des Schwellenwertes auf ½ oder eine Erhöhung der beiden Gewichte auf 2 kann diese Einheit zu einem ODER[29] machen, da dann in beiden Fällen bereits ein einzelnes Eingangssignal ausreichen würde, um den Schwellenwert zu überschreiten und die Aktivierung der Einheit auszulösen. Die Ausbildung eines interdisziplinären Verständnisses von MLA setzt nicht voraus, KNN auf der Ebene der Wahl von Schwellenwerten zu verstehen. Das genannte Beispiel ist dennoch von einiger Bedeutung, weil es einen Eindruck vermittelt, wie im Rahmen eines einfachen Teilschrittes eines Autoadaptionsprozesses ohne einen Steuerungseingriff aus einem UND ein ODER werden kann. Dies ist ein verhältnismäßig konkretes Beispiel dafür, was bei MLA unter Selbstorganisation verstanden werden kann.

Ein drei- beziehungsweise mehrlagiges Perzeptron schließlich kann – bei Verwendung der angedeuteten, etwas komplexeren Aktivierungsfunktion – bereits alle mathematischen Funktionen von praktischer Relevanz beliebig genau annähern[30].

Die Fähigkeit mehrschichtiger Netze, automatisch mathematische Funktionen mittels verborgener Schichten erstellen zu können, ermöglicht einen beträchtlichen Grad von Flexibilität bei der Suche nach Strukturvorschlägen, da diese nicht im Vorfeld vom Nutzer vorgegeben werden müssen. Entsprechend können KNN Strukturen vorschlagen, die dem Nutzer völlig unbekannt sind – und gegebenenfalls auch nach Aufbau beziehungs-

29 Ein mathematisches ODER entspricht einem einschließenden ODER, das heißt einem ›Und-oder‹ und gerade keinem Entweder-oder.

30 Insbesondere können unstetige Funktionen angenähert werden. Dies wird plausibel, wenn klar ist, dass zwei Schichten sigmoider Einheiten bereits alle stetigen Funktionen abbilden können und dass unstetige Funktionen sich durch Linearkombinationen von lokal definierten stetigen Funktionen darstellen lassen.

weise Adaption der verborgenen Schichten unbekannt bleiben, da verborgene Schichten nicht ohne Weiteres verständlich oder gar selbsterklärend sind. Hinzu kommt der systematische abduktive Bias azyklischer KNN, der in etwa darin besteht, dass angenommen wird, dass eine gleichförmige Annäherung an die gesuchte Struktur möglich ist. Meist wird diese Annahme jedoch vom Nutzer des MLA geteilt, wodurch der Bias an Bedeutung verliert. Mitunter können darüber hinaus bis zu einem gewissen Grad mögliche ›Bedeutungen‹ für die verborgenen Schichten gefunden werden, etwa wenn eine lokale Repräsentation vorliegt oder eine Analyse zeigt, dass eine spezielle verborgene Einheit einer Bilderkennung die Eigenschaft ›links ist es sehr hell‹ codiert.

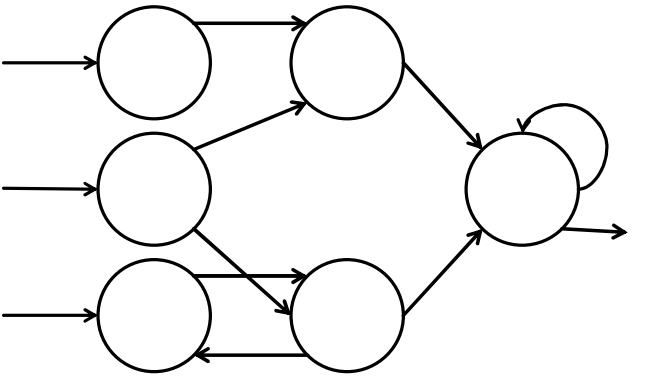
Während allgemeine Aussagen der oben dargestellten Form über die Darstellungskraft von azyklischen KNN möglich sind, gilt dies nur sehr eingeschränkt für Aussagen bezüglich konkreter Funktionen. Im Einzelfall ist es sehr kompliziert, analytisch für eine Funktion oder eine Funktionenklasse zu bestimmen, wie viele verborgene Einheiten und Verbindungen genau benötigt werden, um die Funktion annähern oder abbilden zu können. Diese Schwäche spielt jedoch keine Rolle, wenn Suchräume betrachtet werden, über die im Vorfeld sehr wenig bekannt ist. In solchen Fällen ist unabhängig von der eingesetzten Lernstrategie unbekannt, wie das MLA genau vorstrukturiert werden muss. Varianten evolutionären Lernens, die in der Lage sind ihre Strukturvorschläge sehr stark zu verändern und KNN mit ihrer sehr großen Darstellungskraft eignen sich besonders gut für Einsätze in solchen Kontexten.

Alternativ zu azyklischen KNN kann die Erzeugung von Schleifen beziehungsweise Rückkopplungen im Rahmen des Autoadaptionsprozesses auch erlaubt sein. Ein solches REKURRENTES KNN gibt Teile seiner Ausgabesignale als Eingabesignale an sich selbst weiter. Die Gewichte dieser Art von KNN bilden ein DYNAMISCHES SYSTEM, dessen unterschiedliche Reaktionsweisen sich meist mittels einer gewissen Zahl systematisch unterschiedlicher Zustände beschreiben lassen. Zu jedem Zeitpunkt kann sich das KNN entweder in einem chaotischen Zustand, einem stabilen Zustand oder einem schwingenden Zustand befinden. Ein chaotischer Zustand beschreibt eine zufällig erscheinende Reaktionsweise, ein stabiler Zustand eine Reaktionsweise analog zu einem azyklischen KNN und ein schwingender Zustand entspricht einem KNN, das sich wie ein Pendel zwischen mindestens zwei unterschiedlichen Reaktionsweisen hin und her bewegt. Die

Antwort des KNN ist entsprechend abhängig von seinem derzeitigen Zustand, der wiederum von früheren Eingaben abhängig ist. Die Unterscheidung zwischen der Änderung eines Zustandes und einer Autoadaption kann verglichen werden mit der Nutzung eines Lichtschalters. Ein Lichtschalter hat meist zwei Zustände und abhängig von diesen Zuständen reagiert er auf eine Betätigung mit dem Ein- oder dem Ausschalten des Lichtes. Diese Zustände können jedoch nicht sinnvoll als das Ergebnis eines systematischen Autoadaptionsprozesses beschrieben werden.

Die Veränderung eines rekurrenten KNN unterscheidet sich auch insofern von den Adaptionen anderer KNN oder auf anderen Lernstrategien basierenden MLA, als der Autoadaptionsprozess zunächst ein Eingabedatum registriert und meist in einem gesonderten Schritt eine Aktualisierung oder Adaption vornimmt. Rekurrente KNN verändern sich mitunter schon bei der Registrierung von Eingabesignalen. Im Falle des Vorliegens eines schwingenden oder chaotischen Zustandes kann diese Veränderung der Antwort des dynamischen Systems – des Strukturvorschlags – sogar beliebig lange andauern. Denkbar wäre etwa ein KNN, das in Reaktion auf ein Eingabesignal seinen Zustand verändert, ein Ausgabesignal erzeugt und dieses wieder als Eingabesignal aufnimmt. Die Abhängigkeit von vergangenen Eingaben tritt dementsprechend nicht nur als systematische oder zumindest bewertete Anpassung im Rahmen des Autoadaptionsprozesses auf.

Abbildung 26: Rekursive Verbindung in einem rekurrenten KNN



Rekursive Netze sind als dynamische Systeme schwieriger mittels einzelner Trainingsdaten zu formen als azyklische Netze und der Autoadaptionsprozess verläuft weniger systematisch, allerdings können manche Abhängig-

keiten innerhalb der betrachteten Rohdaten von rekursiven Netzen besser modelliert werden. Zur Erzeugung eines systematischeren Verhaltens können rekursive Netze durch azyklische Netze dargestellt werden, indem das betrachtete rekursive Netz vervielfältigt wird und rekursive Verbindungen jeweils in die nächste Kopie des Netzes eingehen. Der mit dieser Maßnahme verbundene Aufwand führt dazu, dass versucht wird, das Problem zu umgehen und nicht zu lösen. Ein Beispiel für solch einen Umweg besteht darin, einen evolutionären Algorithmus einzusetzen, der KNN codiert. Das Problem hierbei ist, dass in diesem Fall zwar formal KNN verwendet werden, allerdings nicht als Grundlage für eine Lernstrategie, sondern lediglich als Darstellungsform eines rekurrenten Zusammenhangs. Die Kenntnis weiterer Details zu rekurrenten KNN ist für ein interdisziplinäres Verständnis nicht vonnöten. Allerdings ist es hilfreich zu wissen, dass auch die hier nur angedeuteten rekurrenten KNN noch weit von der Komplexität biologischer neuronaler Netze entfernt sind. Biologische neuronale Netze haben beispielsweise die zusätzliche Anforderung, dass Neuronen ein Membranpotenzial besitzen, das ausreichend groß sein muss, wenn das entsprechende Neuron in der Lage sein soll zu feuern. Entsprechend spielt der genaue Zeitpunkt des Feuerns eines Neurons in biologischen neuronalen Netzen eine bedeutende Rolle. Diese und andere Eigenschaften lassen sich abstrakt als Erweiterung in die Idee von dynamischen Systemen aufnehmen und werden im Rahmen von komplexeren KNN modelliert.

Stutzen der Netzstruktur gegen Überanpassung

In der bisherigen Betrachtung wurde der Lernvorgang eines KNN meist mit der Modifikation seiner Gewichte identifiziert. Dies lässt sich erweitern, indem das KNN den eigenen Aufbau als Netz ebenfalls adaptiert. Die Neuschaffung oder die Entfernung von Verbindungen zwischen Knoten und von Einheiten kann in den Autoadaptionsprozess aufgenommen werden anstatt beides vor Beginn des Prozesses zu fixieren. Diese Vorgehensweise wurde bei der Darstellung der adaptiven Resonanztheorie bereits angedeutet. Dort bestand die Möglichkeit, lokale Repräsentationen zu realisieren, indem die entsprechenden Einheiten neu erstellt werden oder im Vorfeld als solche ausgewählt werden (Fahlman 1991). Wenn ein KNN im Rahmen des Autoadaptionsprozesses erweitert werden soll, kann eine verborgene Einheit ergänzt und deren Gewichte heuristisch so eingestellt werden, dass der

Fehler des erweiterten Netzes minimal ist. Anschließend werden die Gewichte der neu ergänzten Einheit fixiert, während die Gewichte des ursprünglichen Netzes adaptiert werden. Dies wird mehrfach wiederholt. Eine solche Erweiterung von KNN ermöglicht eine sehr schnelle Adaption an neuartige Signale. Allerdings liegt eine naheliegende Gefahr in der übermäßigen Ergänzung verborgener Einheiten und damit einer Überanpassung an die Trainingsdaten. Als Gegenmaßnahme lassen sich die Anzahl und Art der Verknüpfungen zwischen den Einheiten und die Anzahl der Einheiten auch wieder reduzieren. Die Vorgehensweise ähnelt dabei dem Stutzen von Entscheidungsbäumen. Ein vollständig vernetztes KNN wird erstellt und es werden Verbindungen zwischen Einheiten oder ganze Areale identifiziert, deren Relevanz für das Gesamtnetz fraglich ist. Ein Grund kann sein, dass sich die Gewichte im entsprechenden Gebiet während des gesamten Autoadaptionsprozesses kaum verändert haben. Die Auswirkung des Teilnetzes auf die Performanz des Gesamtnetzes kann überprüft werden, indem ein zweites KNN erstellt wird, dem die identifizierten Verbindungen oder Areale fehlen. Dieses zweite KNN kann anschließend mit dem ursprünglichen KNN auf Performanz oder bezüglich anderer Kriterien wie der Antwortgeschwindigkeit verglichen werden[31]. Ein Hauptziel, das mit der Stutzung von KNN verfolgt wird, ist die Vermeidung oder Reduzierung von Überanpassungen. Ein großes KNN kann genau wie ein großer Entscheidungsbaum alle Trainingsinstanzen reproduzieren, indem es in den verborgenen Schichten eine Art Nachschlagetabelle anlegt. Kurz gesagt verringert sich die Tendenz zur Überanpassung mit sinkender Anzahl von verborgenen Einheiten. Eine Möglichkeit sehr große KNN zu vermeiden, anstatt sie zu reduzieren, besteht darin, aus mehreren unabhängigen kleinen Netzen große Netze zusammen zu setzen. Jedes der betrachteten kleinen Netze muss in diesem Fall bereits isoliert möglichst viele Trainingsdaten erklären können. Anschließend werden solange kleine Netze, die disjunkte Mengen von Trainingsdaten erklären, zusammengeschaltet, bis alle Trainingsdaten abgedeckt sind.

Insgesamt kann der Autoadaptionsprozess eines KNN, wie schon derjenige bei evolutionärem Lernen, sehr weitreichende Veränderungen des

31 Dieser Vergleich unterschiedlich aufgebauter KNN kann so intensiv betrieben werden, dass für einen bestimmten Kontext heuristisch eine optimale Anzahl von Schichten und der Knoten pro Schicht bestimmt werden können.

Strukturvorschlages bewirken. Im Gegensatz zu evolutionärem Lernen wird bei KNN allerdings keinerlei Wert auf eine Codierung gelegt und eine prinzipielle Unverständlichkeit der Prozesse wird hingenommen, um eine maximale Darstellungskraft zu gewinnen. Ein interdisziplinäres Verständnis der Funktionsweise von künstlichen neuronalen Netzen hängt stark von der Vermeidung von Denkfehlern ab, daher wurde für die Darstellung der Motivation und der Funktionsbeschreibung in dieser kurzen Diskussion der Lernstrategie ein vergleichsweise großer Aufwand betrieben. KNN sind jedoch auch für die meisten technikphilosophischen Diskussionen eines MLA oder eines ›selbstorganisierten Algorithmus‹ von Bedeutung und können häufig als Beleg oder Gegenbeispiel für eine These eingesetzt werden.

Zusammenfassung des Zweckes von KNN

Zusammengefasst lässt sich sagen, dass künstliche neuronale Netze einen allgemeinen, praktischen Ansatz darstellen, um auf Basis von Messwerten Strukturvorschläge zu erstellen, die Funktionen abbilden, die mit einer großen Zahl von Eingangsgrößen agieren. Zum Einsatz kommende Algorithmen wie die BACKPROPAGATION benutzen iterative Methoden, um die Parameter von künstlichen neuronalen Netzen so einzustellen, dass diese möglichst performant auf einer Menge von Trainingsdaten sind, die in Form von Eingabe-Ausgabe-Paaren vorliegen. Autoadaptionsprozesse auf Basis von KNN sind unempfindlich gegenüber Rauschen und werden erfolgreich für die Bearbeitung einer Vielzahl von Problemen eingesetzt[32].

Trainingsdaten, die an ein KNN übergeben werden und die eine Modifikation der Verbindungsgewichte bewirken sollen, sind typischerweise durch eine große Anzahl von mit Zahlenwerten versehenen Attributen codiert. Messwerte zu diesen Attributen können dem KNN in nahezu jeder Art und Weise übergeben werden. Es ist nicht erforderlich, eine konsistente oder alle Teilaspekte der Problemumgebung erfassende Codierung zu erstellen, die Codierung darf und wird typischerweise fragmentarischen Charakter haben. Die Attribute müssen weder unabhängig noch korreliert sein, sie müssen nicht einmal vergleichbar oder widerspruchsfrei sein. Das be-

32 Eine sehr gut zugängliche und frei verfügbare Einführung in die technischen Details und die Möglichkeiten zur Implementierung von KNN findet sich bei Kriesel (Kriesel 2007).

deutet, eine Messung muss lediglich all diejenigen Aspekte der Trainingsinstanzen, die als potenziell relevant für das unbekannte und zu lernende Konzept eingestuft werden, als Zahlenwerte erfassen und dem KNN anschließend in beliebiger – wenn auch über alle Instanzen konstanter – Reihenfolge übergeben. Diese Robustheit gegenüber der Übergabereihenfolge von Messwerten veranschaulicht die hohe Fehlertoleranz von KNN gegenüber verrauschten Eingabewerten, aufgrund derer KNN sich besonders gut zur Darstellung und Verarbeitung von Unschärfe sowie Rauschen eignen. Die Fähigkeit künstlicher neuronaler Netze, auch in Kontexten einsetzbar zu sein, die dem Nutzer fremd oder völlig unbekannt sind, wird im zweiten Hauptteil eine zentrale Rolle spielen.

2.3.5 Instanzenbasiertes Lernen

Motivation

Die zentrale Motivation des INSTANZENBASIERTEN oder DESKRIPTIVEN LERNENS besteht darin, die Trainingsdaten beziehungsweise Instanzen den Strukturvorschlag direkt und möglichst stark beeinflussen zu lassen. Die Trainingsdaten sollen möglichst unmittelbaren Einfluss auf die Klassifizierung neuer Eingabedaten haben. Insbesondere sollen die Trainingsdaten gerade *nicht* nur dazu genutzt werden, einen Strukturvorschlag in Form eines Baumes oder eines Netzes zu erstellen. Nach Abschluss der Erstellung eines Entscheidungsbaumes besitzen die Trainingsdaten etwa allenfalls noch eine implizite Bedeutung. Die wichtigste Vorannahme des instanzenbasierten Lernens muss dementsprechend sein, dass neue Ereignisse sehr wahrscheinlich den bereits bekannten Ereignissen ähneln. Dies kann gesteigert werden bis hin zu der Annahme, dass grundsätzlich keine Überraschungen auftreten und alle neu registrierten Eingabedaten immer bereits zuvor registrierten Eingabedaten ähneln. Diese Annahme ist verwandt zum Bias bei KNN, dort war die implizite Grundannahme, dass die Strukturunterschiede innerhalb der Rohdaten nicht extrem groß sind. Beide Annahmen stellen Schwächen der jeweiligen Lernstrategien dar, mit denen jedoch genau gegensätzlich umgegangen wird. Während KNN die Annahme als impliziten systematischen Fehler ignorieren, wird sie beim instanzenbasierten Lernen zum zentralen Merkmal der Lernstrategie.

Die Idee der Klassifizierung von Eingabedaten mittels eines direkten Bezugs zu den Trainingsdaten war ebenfalls im Rahmen der Darstellung KNN bereits aufgetaucht. Dort wurden beim Konzept der adaptiven Resonanztheorie lokale Repräsentationen von Eingabedaten erzeugt. Die adaptive Resonanztheorie hatte dabei in erster Linie versucht, für jedes neu hinzugekommene Eingabedatum einen Abgleich mit den bisherigen Klassen oder klassifizierten Instanzen vorzunehmen, um die gebildeten Klassen den betrachteten Instanzen anzunähern. Diese Vorgehensweise steht beim instanzenbasierten Lernen im Fokus. Die Diskussion instanzenbasierter Lernstrategien kann dementsprechend genutzt werden, um andere Lernstrategien, die diesen Anspruch nicht explizit formulieren, anders und mitunter besser zu verstehen.

Ein Problem der Kategorisierung dieser Klasse von MLA und des nachfolgenden STATISTISCHEN LERNENS liegt darin, dass sich der Fokus von der Beschreibung einer Strategie zur Erstellung eines Autoadaptionsprozesses entfernt. Stattdessen steht tendenziell eine Strategie zur Erstellung von Strategien im Vordergrund. Dieses Problem ist jedoch nicht sehr schwerwiegend, da auch evolutionäres Lernen eine Gruppe von Ansätzen umfasste, die stark verwandt waren. Die Methoden, mit denen instanzenbasiertes Lernen realisiert wird, werden im Weiteren allerdings nicht so detailliert betrachtet, wie die genetischen Algorithmen, die genetische Programmierung und die Evolutionsstrategien als Varianten evolutionären Lernens. Der Grund hierfür ist, dass die eingesetzten Methoden aus mathematischer Sicht relativ geradlinig sind und in erster Linie versuchen, die konzeptionellen Vorgaben des instanzenbasierten Lernens möglichst gut zu realisieren. Eine Diskussion dieser Methoden ist für ein interdisziplinäres Verständnis nicht zwingend notwendig, da sie kaum einen Mehrwert gegenüber der noch folgenden Betrachtung der STÜTZVEKTORMETHODEN bietet.

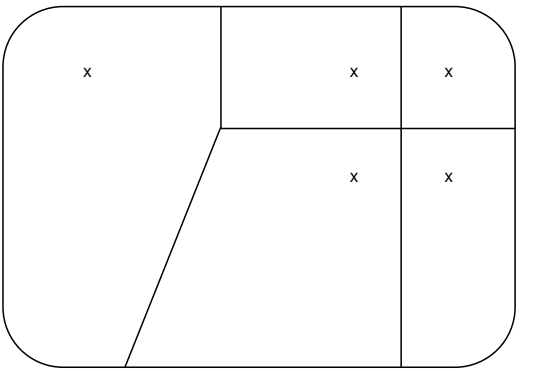
Einführungsbeispiele

Ein Beispiel für die Umsetzung von menschlichem instanzenbasiertem Lernen liegt vor, wenn begonnen wird, ein Puzzle zusammenzusetzen, insbesondere wenn die puzzelnde Person das Bild noch nicht genau betrachtet hat. Üblicherweise wird damit begonnen, einen Puzzlestein zu einem ähnlichen Puzzlestein oder einer zufällig schon bestehenden Häufung von Steinen des entsprechenden Musters zu legen. Dieses Vorgehen wird wieder-

holt, bis die Steine grob in Kategorien wie Himmel, Rand und so weiter eingeteilt wurden. Die Annahme, dass neue Instanzen aussehen wie bisherige Instanzen, ist in diesem Fall gerechtfertigt, da nur die wenigsten Puzzles Steine enthalten, die keinem anderen Stein ähnlich sehen und jeweils die gesamte Restmenge an Steinen als Referenzgruppe für den isolierten Stein dient. Ein anderes Beispiel für manuelles instanzenbasiertes Lernen ist das Bleigießen. Hier handelt es sich um ein besonders reines Beispiel, da die konstituierende Vorannahme des Bleigießens genau darin besteht, dass die entstehenden Gebilde bereits bekannten Gebilden ähneln[33].

Eine grafische Visualisierung für die instanzenbasierte Zerlegung einer Fläche in Teilflächen ist ein VORONOI-DIAGRAMM. Die Teilflächen werden in diesem Fall entsprechend der Lage der Trainingsdaten bestimmt. Das jeweils nächstgelegene Trainingsdatum bestimmt die Teilfläche, der ein Punkt zugeordnet wird.

Abbildung 27: Voronoi Diagramm mit fünf Teilflächen



Voronoi-Diagramme veranschaulichen das Resultat der Regel ›jede Instanz soll so klassifiziert werden, wie die ihr nächstgelegene Trainingsinstanz‹. Die Erstellung des Voronoi-Diagramms selbst ist jedoch gerade kein instanzenbasiertes Lernen, da das Diagramm eine Aussage über den gesamten Hypothesenraum trifft und die Arbeit nach der Erstellung des Diagramms

33 Formalere Beispiele und ein technischer Ansatz das instanzenbasierte Lernen insgesamt zu beschreiben, finden sich in einem Text zum »Fallbasierten Problemlösen in Expertensystemen. Begriffliche und inhaltliche Betrachtungen« (Althoff et Weß 1991).

bereits getan ist. Das Diagramm visualisiert lediglich, welches Resultat sich aus instanzenbasiertem Lernen ergibt.

Funktionsbeschreibung

Autoadaptionsprozesse finden im Rahmen von instanzenbasiertem Lernen für eine neu zu klassifizierende Instanz dasjenige bereits klassifizierte Trainingsdatum, das der neuen Instanz am ähnlichsten ist und ordnen beide einer gemeinsamen Klasse zu. Wenn Ähnlichkeit als eine Entfernung interpretiert wird, wird der Raum der Eingabedaten wie im Beispiel des Voronoi-Diagramms in Umgebungen um die Trainingsdaten unterteilt. Instanzenbasiertes Lernen beschreibt lokale Zusammenhänge, die gegebenenfalls nicht für den gesamten Raum der Eingabedaten eine Aussagekraft haben.

Instanzenbasierte Lernstrategien werden als FAULE Lernstrategien bezeichnet, da das Maß zur Bestimmung der Ähnlichkeit vom Nutzer vorgegeben wird und spezielle Trainingsinstanzen erst dann ausgewertet beziehungsweise berücksichtigt werden, wenn eine Eingabe erfolgt, die dies erfordert. Der Autoadaptionsprozess besteht initial nur darin, eine Datenbank der Trainingsdaten zu erstellen. Diese Datenbank stellt in Kombination mit dem Ähnlichkeitsmaß formal den Strukturvorschlag dar. Den Strukturvorschlägen anderer Lernvorgänge vergleichbar wäre die Betrachtung von Visualisierungen wie dem Voronoi-Diagramm.

Definition

MLA, die instanzenbasiertes Lernen realisieren, basieren auf zwei Grundsätzen:

- Entscheidungen über die Klassifikation von Eingabedaten werden vorgenommen, wenn konkrete Daten vorliegen.
- Eingabedaten werden auf Ähnlichkeit zu Trainingsdaten überprüft und werden auf Basis des ähnlichsten Teils der Trainingsdaten klassifiziert.

In diesen Grundsätzen ist implizit die Aussage enthalten, dass die weniger ähnlichen Trainingsinstanzen bei der Klassifizierung eines Eingabedatums ignoriert werden und dass keine generalisierenden Hypothesen über die Trainingsdaten aufgestellt werden sollen. Formal wird bei instanzenbasier-

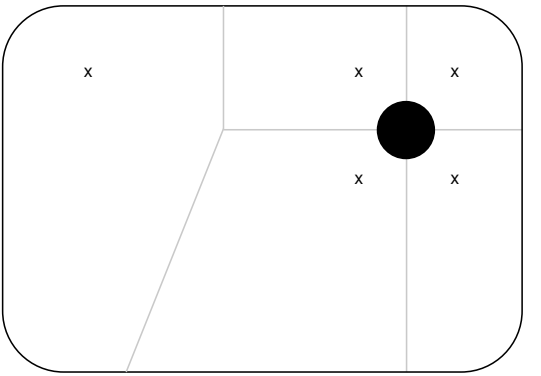
tem Lernen darüber hinaus gefordert, dass Eingabedaten in einer speziellen mathematischen Codierung vorliegen[34]. Das Voronoi-Diagramm etwa bezog sich auf eine zulässige zweidimensionale Darstellung der Eingabedaten. Die mathematischen Forderungen sind so formuliert, dass eine mathematisch sinnvolle Definition des Abstands der Daten möglich ist und dieser als Maß der Ähnlichkeit verwendet werden kann. Der Abstandsbegriff kann durchaus sehr komplex definiert sein. Im Weiteren wird auf diese Forderung verzichtet, da es für ein interdisziplinäres Verständnis nur einen geringen Mehrwert liefert, zu diskutieren, welche Rohdaten sich auf diese spezielle Weise codieren lassen. Die weiteren Betrachtungen beschäftigen sich dementsprechend formal mit dem FALLBASIERTEN LERNEN. Das fallbasierte Lernen ist ein Spezialfall des instanzenbasierten Lernens, bei dem die Rohdaten relativ frei codiert sein können. Ein Beispiel für fallbasiertes Lernen ist die Erstellung von medizinischen Diagnosen auf Basis der Zuordnung von aufgetretenen Symptomen zu ähnlichen, bereits klassifizierten Symptomen. Zur Bearbeitung von Problemen, die sich gut anschaulich codieren lassen, sich aber gegen eine formale, mathematische Codierung sperren, sind sehr verschiedene Ansätze denkbar. In allen Einzelfällen fließt allerdings sehr viel fallspezifisches Vorwissen in die Konstruktion des Algorithmus ein. Das Ausmaß dieser Anpassungen an die Spezialfälle kann so weit gehen, dass nicht mehr ohne Weiteres davon gesprochen werden kann, dass eine bestimmte Lernstrategie angewandt wird. Im Weiteren wird dennoch nicht zwischen instanzen- und fallbasierten Lernstrategien unterschieden, da die Strategie hinter fallbasiertem Lernen mit derjenigen von instanzenbasiertem Lernen übereinstimmt. Wichtig ist jedoch festzuhalten, dass instanzenbasiertes Lernen durch die Forderung einer bestimmten mathematischen Codierung die Verwendung bestimmter mathematischer Optimierungsverfahren ermöglichen will.

Instanzenbasiertes Lernen kann zwei unterschiedliche Ziele verfolgen, die unterschiedliche Konsequenzen für die Lernstrategie haben. Zum einen kann die Klassifizierung von Eingabedaten gemäß dem ähnlichsten Trainingsdatum im Vordergrund stehen. Die alternative Zielvorstellung besteht darin, lokale Hypothesen über die Eigenschaften der Rohdaten zu suchen. Die Rohdaten werden zu diesem Zweck in NACHBARSCHAFTEN zerlegt gedacht, wobei nicht ein Trainingsdatum das Zentrum einer Nachbarschaft

34 Als REELLWERTIGE PUNKTE im N-DIMENSIONALEN EUKLIDISCHEN RAUM.

darstellt, sondern genau die gegenteilige Idee umgesetzt wird. Für einen Bereich, in dem nicht unmittelbar ein Trainingsdatum vorliegt, werden die nächstgelegenen Trainingsdaten als Nachbarn und der Bereich selbst als eine Nachbarschaft bezeichnet. In der nachfolgenden Abbildung wurde das obige Beispiel um eine Nachbarschaft mit vier Nachbarn erweitert.

Abbildung 28: Voronoi Diagramm mit einer Nachbarschaft



Die Daten in dieser Nachbarschaft werden als von den vier Nachbarn beeinflusst gedacht. Die Suche nach möglichen Strukturen einer Nachbarschaft bleibt den Grundsätzen des instanzenbasierten Lernens treu. Zum einen sollen neue Instanzen ausschließlich unter Nutzung bereits bekannter Instanzen klassifiziert werden, die Ausschließlichkeit bezieht sich insbesondere darauf, dass auf die Formulierung oder Nutzung globaler Hypothesen explizit verzichtet wird. An diese Forderung anknüpfend, soll zum anderen die Komplexität der Strukturvorschläge dynamisch mit Anzahl der Trainingsdaten wachsen – im Gegensatz zur Parametrisierung einer vorgegebenen Menge von Modellen für eine beliebig große Datenmenge. Gleichzeitig wird bei der Betrachtung von Nachbarschaften zwar eine über einen einzelnen Datenpunkt hinausgehende Aussagekraft angestrebt, aber nur insofern als keine Trainingsdaten dem widersprechen. Die speziellen Parameter bei der Erstellung von lokalen Strukturvorschlägen für Nachbarschaften liegen in der Anzahl der zu berücksichtigenden Nachbarn sowie der Wahl und Gewichtung der zu berücksichtigenden Eigenschaften der Trainingsdaten. Die Suche nach lokalen Strukturvorschlägen für Nachbarschaften steht ganz im Sinne des instanzenbasierten Lernens im Gegensatz zum GLOBALEN oder PRÄDIKATIVEN HYPOTHESENLERNEN anderer Lernstrategien. Die

Vorgehensweise wird in Abgrenzung von diesem Vorgehen als LOKALES oder DESKRIPTIVES HYPOTHESENLERNEN bezeichnet. Auch das lokale Hypothesenlernen ist jedoch wie bereits angedeutet nicht ausschließlich lokal, da zumindest der Abstands- oder Ähnlichkeitsbegriff auf alle Rohdaten anwendbar sein muss.

Varianten instanzenbasiertes Lernens

Typische Anwendungssituationen für instanzenbasiertes Lernen sind solche, in denen die Nachbarschaften nur lokal definiert werden, sich aber immer mehr verfestigen und entgegen den Grundsätzen des instanzenbasierten Lernens nicht für jedes zusätzliche Eingabedatum völlig neu berechnet werden sollen. Eine gezielte Verfestigung von Nachbarschaften tritt ebenfalls vergleichsweise häufig auf, etwa wenn eine bestehende und feste Rohdatenmenge weiter analysiert werden soll. In diesem Fall liegt der Fokus nicht darin, Voraussagen für neue Instanzen zu ermöglichen, sondern die bereits vorliegende Rohdatenmenge besser zu strukturieren. Die beiden in diesem Kontext eingesetzten Varianten instanzenbasierten Lernens sind die der Subgrupppenentdeckung und der Clusteranalyse.

Der Fokus der SUBGRUPPPENENTDECKUNG liegt darauf, besonders interessante Teile der Rohdaten zu identifizieren, wenn die Gesamtmenge an Rohdaten sehr komplex oder chaotisch erscheint. Zu diesem Zweck wird eine Zielfunktion festgelegt, um den Interessantheitsgrad von Gruppen von Rohdaten zu bewerten. Üblicherweise wird die Interessantheit von Rohdaten dabei über eine signifikante Abweichung von einem erwarteten Wert gemessen. Eine solche Abweichung zu messen ist insbesondere dann möglich, wenn die VERTEILUNG der Rohdaten zumindest näherungsweise bekannt ist. Als SUBGRUPPEN werden anschließend entweder diejenigen Gruppierungen von Rohdaten, die einen gewissen Mindestgrad an Interessantheit aufweisen, oder die qualitativ besten Gruppen bezeichnet[35]. Zur Veranschaulichung sei angenommen, ein Finanzdienstleister würde planen, zielgruppenorientierte Werbung für die Anschaffung von Kreditkarten zu

35 Weiterhin werden bei einer Subgrupppenentdeckung für instanzenbasiertes Lernen vergleichsweise komplexe Codierungen für Strukturvorschläge zugelassen – etwa indem zeitliche Zusammenhänge zwischen einzelnen Instanzen mitbetrachtet werden.

entwerfen. Von besonderem Interesse sind in diesem Kontext Kundengruppen, die einen strukturellen Zusammenhang aufweisen und unter denen gleichzeitig der Anteil an Kreditkartennutzern relativ zum Gesamtdurchschnitt noch besonders gering ist. Der Finanzdienstleister kann jetzt seine Kundendaten sichten, um entsprechende Subgruppen zu identifizieren und gruppenspezifische Werbemaßnahmen zu initiieren. Bekannt ist in diesem Beispiel lediglich die Zielfunktion. Der Finanzdienstleister muss keinerlei Vorwissen darüber besitzen, welche Gemeinsamkeiten die einzelnen Subgruppen aufweisen. Es muss nicht einmal sichergestellt sein, dass es solche Subgruppen überhaupt gibt. Dieses Beispiel der Subgruppenentdeckung verdeutlicht, dass die Nutzer eines MLA mitunter lediglich vage Wünsche formulieren und das MLA die Aufgabe hat, mittels der Strukturvorschläge Inspirationen für Maßnahmen zur Erfüllung der Wünsche zu liefern. Das MLA kann in dem genannten Beispiel ohne Weiteres bei dem Versuch scheitern, interessante Subgruppen zu entdecken.

Im Rahmen der CLUSTERANALYSE werden die Rohdaten so in CLUSTER genannte Teilmengen aufgeteilt, dass Rohdaten innerhalb eines Clusters einander möglichst ähnlich und den Rohdaten außerhalb des Clusters möglichst unähnlich sind. In der Mehrzahl der Fälle ist die Aufteilung in Cluster überschneidungsfrei und bezieht alle Rohdaten mit ein. Zwar wird meist eine Menge von Rohdaten in Cluster aufgeteilt, allerdings kann auch eine AGGLOMERATIVE CLUSTERANALYSE durchgeführt werden, bei der Einzeldaten zu kleinen Clustern zusammengefasst werden, die wiederum zu größeren Clustern zusammengefasst werden[36]. Die benötigte Ähnlichkeitsfunktion der Clusteranalyse basiert wie häufig bei instanzenbasiertem Lernen auf einem Abstandsbegriff. Die Ähnlichkeit kann jedoch auch durchaus als eine semantische Ähnlichkeit definiert sein. Dieser Begriff von Ähnlichkeit ist formal vergleichbar mit dem Konzept des Interessantheitsgrades der Sub-

36 Die Aufteilung einer Gesamtheit von Rohdaten hat den Nachteil, dass der Idee des langsamen Anwachsens der Komplexität der Strukturvorschläge entgegengewirkt wird. Im Extremfall führt dieses Vorgehen dazu, dass eine globale Aussage über die Struktur der Gesamtheit der Rohdaten gesucht wird und die Cluster einzelne Aspekte dieser Struktur darstellen sollen. Zwar stellt dieser Extremfall kein instanzenbasiertes Lernen mehr dar, aber die Ausrichtung der Clusteranalyse wird in der Praxis je nach Kontext und Ausmaß der Verschiedenheit der vermuteten Cluster vorgenommen.

gruppenentdeckung, verzichtet aber gezielt auf eine Hierarchie, da es gerade keine Instanz geben soll, die ein hohes Maß an Ähnlichkeit zu allen anderen Instanzen gleichermaßen besitzt. Jenseits der Definition einer Ähnlichkeitsfunktion, die auch bei der Grundform instanzenbasierten Lernens benötigt wird, sind bei der Clusteranalyse keine Vorgaben notwendig. Die Anzahl der Cluster kann prinzipiell im Vorhinein festgelegt werden, eine solche Vorgabe ist jedoch nicht notwendig. Vorwissen oder Vermutungen bezüglich der Strukturen innerhalb der Rohdaten können stattdessen in der Wahl der Mittel zur Umsetzung einer Clusteranalyse genutzt werden. Die im Weiteren noch diskutierte statistische Clusteranalyse unterteilt oder gruppiert Eingabedaten auf Basis der Annahme gewisser Verteilungen zur Schätzung VERBORGENER PARAMETER, das heißt von Parametern, die nicht direkt gemessen werden können. Andere Methoden der Clusteranalyse sind die im Rahmen der Betrachtung von KNN bereits dargestellte adaptive Resonanztheorie und die selbstorganisierenden Merkmalskarten. Diese Beispiele zeigen das Bestehen von Übergangsbereichen zwischen den Lernstrategien an, allerdings verletzen die MLA in diesen Übergangsbereichen meist immer stärker die Grundideen der einen Lernstrategie, wenn sie sich der Vorgehensweise einer anderen Strategie annähern. Die Nutzung von Vorwissen und die gezielte Erstellung desselben führt häufig dazu, dieses als Grundlage des MLA zu verwenden und die Lernstrategien allenfalls als Startpunkt bei der individuellen Gestaltung eines Autoadaptionsprozesses zu betrachten. Wichtig ist festzuhalten, dass auf Vorwissen basierende MLA häufig kaum noch Aspekte aufweisen, die in der Diskussion der Selbstorganisation von Algorithmen eine Rolle spielen.

Darstellungskraft

Die meisten Einsätze von MLA, die auf der Idee des fallbasierten Lernens aufbauen, nutzen in der Umsetzung eine weitere Lernstrategie – die allerdings stark angepasst wird. Der hybride Charakter dieser Teilklasse des instanzenbasierten Lernens ist so ausgeprägt, dass das fallbasierte Lernen als NACHGEORDNETE LERNSTRATEGIE betrachtet werden kann. Nach Wahl einer Lernstrategie, etwa der KNN, kann entschieden werden, ob ein INSTANZENBASIERTER oder KONZEPTORIENTIERTER Autoadaptionsprozess eingesetzt werden soll. Häufig wird auch innerhalb einer anderen Lernstrategie ein Autoadaptionsprozess auf Basis der Grundsätze des instanzenbasierten

Lernens durchgeführt, wenn das Ziel darin besteht, unüberwachtes Lernen zu realisieren.

Instanzenbasierte Lernstrategien erstellen keinen globalen Klassifikator, daher treten auch nicht die damit einhergehenden Einschränkungen bei der Erstellung eines Strukturvorschlages auf. Natürliche Vorgänge und Zusammenhänge sind häufig extrem komplex und lassen sich allenfalls lokal durch einen handhabbaren Strukturvorschlag darstellen. Die Aufteilung in Nachbarschaften erlaubt es instanzenbasierten Lernstrategien, auch chaotisch erscheinende Systeme zu untersuchen und zumindest für Teilsysteme Strukturvorschläge zu erstellen. Entsprechend eignen sich Situationen, in denen kein oder kaum Vorwissen besteht, vergleichsweise gut für instanzenbasierte Lernstrategien, da diese auch dann Muster oder Strukturen finden können, wenn tatsächlich keine globale Struktur vorliegt und der Nutzer im Vorfeld keinen Anhaltspunkt hat, welche Teile der Rohdaten interessante Subgruppen sein könnten. Der korrespondiere Nachteil liegt darin, dass ein Strukturvorschlag, der auf der Suche nach Ähnlichkeiten zwischen Trainingsdaten basiert, keine verborgenen Parameter darstellen kann, da diese sich nicht aus der Registrierung der Eingabedaten ergeben. Dieser Verzicht auf eine Verallgemeinerung der Strukturen der Trainingsinstanzen bedingt, dass Überanpassung ein zentrales Problem des instanzenbasierten Lernens darstellt.

Adaptive Struktur gegen Überanpassung

Überanpassung scheint notwendig gefordert zu sein, wenn alle neuen Instanzen analog zu bereits bestehenden Instanzen klassifiziert werden sollen und gleichzeitig keine neuen Klassen oder etwa Cluster erzeugt werden dürfen. Wenn die Trainingsinstanzen oder Cluster miteinander bezüglich der Einordnung einer neuen Instanz in Konkurrenz stehen, das heißt, wenn KOMPETITIVES Lernen oder WETTBEWERBSLERNEN eingesetzt wird, oder wenn für jede neue Instanz nur genau ein Cluster aktualisiert wird, verschärft sich dieses Problem noch. Im Gegenzug kann die Gefahr einer Überanpassung auch reduziert werden, indem diese Form des Lernens nicht eingesetzt wird. Die Alternative besteht darin, neue Instanzen gemäß einer ganzen Nachbarschaft von Trainingsdaten beziehungsweise Clustern zu klassifizieren und die Einflüsse der unterschiedlichen Trainingsdaten gewichtet zu berücksichtigen. So kann eine Vielzahl von Nachbarn berück-

sichtigt und etwa in Abhängigkeit von ihrer Entfernung gewichtet zur Einordnung der neuen Instanz eingesetzt werden. Einer Überanpassung an eklatante Messfehler oder andere AUSREIßER in den Rohdaten kann vorgebeugt werden, indem die Umgebungen der Ausreißer bei Überschreitung einer gewissen Entfernung zu den übrigen Trainingsdaten als eigene Nachbarschaften beziehungsweise sehr kleine Einzelcluster interpretiert werden. Die Messung der Performanz auf TESTDATEN und VALIDIERUNGSDATEN zur Reduzierung der Überanpassung bietet sich bei instanzenbasierten Lernstrategien besonders an, da im Vorfeld des Autoadaptionsprozesses ohne Probleme Trainingsdaten zu Testdaten umgewidmet werden können. Die Trainingsdaten werden im Autoadaptionsprozess ohnehin erst genutzt, wenn ein neues Eingabedatum eingeordnet werden soll. Testdaten können bei instanzenbasierten Lernstrategien auch genutzt werden, um optimale Werte für Verfahrensparameter wie die Anzahl der Cluster zu bestimmen oder um diejenigen Attribute der Rohdaten zu identifizieren, die im jeweils verwendeten Abstandsbegriff eine besonders große oder kleine Rolle spielen sollen. Eine gezielte Vernachlässigung einzelner Attribute der Rohdaten ist mitunter sehr zentral für den Autoadaptionsprozess. Dies gilt insbesondere, wenn die Rohdaten nicht gut verstanden werden, der Nutzer also im Vorfeld nicht einschätzen kann, welche Attribute überhaupt von Bedeutung sein könnten und dementsprechend eine Vielzahl von Messwerten an das MLA übergibt.

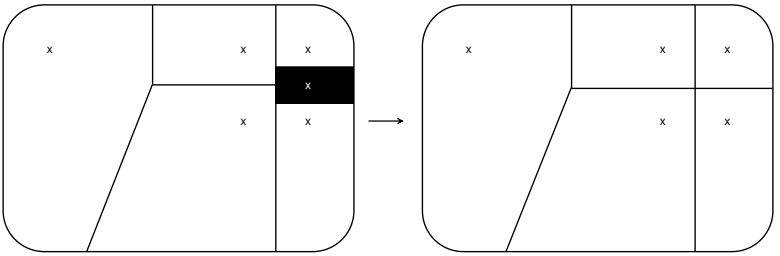
Beispiele für Adaptionen instanzenbasierter Lernstrategien

Im bisher Beschriebenen wurden bereits Veränderungen instanzenbasierter Lernstrategien angedeutet, die in Übergangsbereiche zu anderen Lernstrategien führen. Natürlich kann auch eine Vielzahl von anderen Veränderungen vorgenommen werden, allerdings haben diese meist mit den schon genannten Vorgehensweisen gemein, dass sie den Autoadaptionsprozess von den Grundansätzen des instanzenbasierten Lernens entfernen. Nachfolgend wird eine kleine Auswahl an Weiterentwicklungen dargestellt, die einen Eindruck geben soll, auf welche Weise ergänzende Selbstorganisationsprinzipien in den Autoadaptionsprozess integriert werden können.

Mitunter sollen einzelne oder ganze Gruppen von Trainingsdaten bewusst nicht in den Lernvorgang einbezogen werden, etwa wenn Messgenauigkeiten vorliegen beziehungsweise vermutet werden. Dies führt zu

Veränderungen, die wieder gut in einem Voronoi-Diagramm visualisiert werden können.

Abbildung 29: Vereinfachung eines Voronoi Diagramms



Alternativ kann auch aus mehreren Trainingsdaten ein künstliches Datum konstruiert werden, allerdings widersprechen beide Maßnahmen gleichermaßen dem grundlegenden Ziel instanzenbasierten Lernens, dass keine Trainingsdaten vernachlässigt werden sollen.

Die Subgruppenentdeckung kann verbessert werden, indem bereits gefundene Subgruppen später im Autoadaptionsprozess noch einmal untersucht werden – meist auf zeitliche Zusammenhänge. Ähnliche Subgruppen können anschließend gruppiert werden und der Grad der Interessantheit einer Subgruppe kann noch einmal eingeschätzt werden, diesmal im Vergleich mit ähnlichen Subgruppen und nicht gegenüber der Gesamtmenge an Rohdaten. Wenn eine Subgruppe bezüglich einer anderen keinen großen Informationsgewinn zeigt, kann eine der beiden Gruppen verworfen werden. Der Nutzer wird in diesem Fall lediglich in Form eines Zusatzes zum Strukturvorschlag der verbliebenen Subgruppe über das Verwerfen informiert. Eine solche Vorgehensweise sorgt dafür, dass formal überraschende oder unvorhergesehene, aber dennoch immer wieder vorkommende Strukturen dem Nutzer nicht beliebig oft in ähnlicher Form als neue Strukturvorschläge präsentiert werden. In der weiteren Diskussion wird es von Bedeutung sein, eine grobe Vorstellung zu besitzen, wie MLA eigenständig die Interessantheit von Strukturen bewerten. Die Bewertung identifizierter Subgruppen kann als Beispiel für solch einen Vorgang dienen und ist dementsprechend auch über den Kontext des instanzenbasierten Lernens hinaus von Interesse.

Clusteranalysen schließlich müssen keine überschneidungsfreien Cluster generieren und müssen den Raum auch nicht vollständig in Cluster auf-

teilen. Beispielsweise sind bei HIERARCHISCHEN CLUSTERANALYSEN ineinander verschachtelte Cluster zugelassen. Solche vielfach ineinander verschachtelten Cluster können vom Nutzer als Baumstrukturen interpretiert werden und so entsteht ein Hybridfeld zu den Entscheidungsbäumen oder zumindest eine neue Perspektive auf diese Lernstrategie.

2.3.6 Statistisches Lernen

Motivation

Statistische Lernstrategien basieren auf der Arbeit mit und der Manipulation von Wahrscheinlichkeiten. Im Kontext des maschinellen Lernens bedeutet das, dass der Autoadaptionsprozess eine Struktur vorschlagen soll, die mit besonders großer Wahrscheinlichkeit auch über die Trainingsdaten hinaus von Interesse ist. Die Wahrscheinlichkeitsrechnung und die mathematische Stochastik stellen hierzu ein breites Sortiment an sehr stark ausgearbeiteten Begriffsdefinitionen und Werkzeugen zur Verfügung. Die Ziele dieser mathematischen Theoriebildung liegen in der Prüfung einer Vermutung bezüglich einer unzugänglichen Gesamtmenge von Rohdaten anhand einer REPRÄSENTATIVEN STICHPROBE. Diese Beschreibung entspricht sehr genau den Anforderungen des maschinellen Lernens. Dementsprechend liegt es nahe, aus wahrscheinlichkeitstheoretischen Werkzeugen statistische Lernstrategien gewinnen zu wollen, die belastbare Aussagen über die Wahrscheinlichkeit der Richtigkeit von Hypothesen im Kontext des maschinellen Lernens machen können.

Analog zu der Betrachtung instanzenbasierter Lernstrategien kann die Diskussion statistischer Lernstrategien dazu beitragen das Verständnis anderer Lernstrategien, die Wahrscheinlichkeiten nicht explizit aber sehr wohl implizit verwenden, zu erweitern. Die später beschriebenen Bayes'schen Lernstrategien ähneln etwa in der Praxis den Ideen des instanzenbasierten Lernens. Entsprechend werden die Begriffe Subgruppenerkennung und Clusteranalyse wieder auftauchen und es wird die Perspektive des statistischen Lernens auf hybride Autoadaptionsprozesse dargestellt werden.

Einführungsbeispiel

Das meistgenutzte Beispiel für ein ZUFALLSEXPERIMENT ist der Münzwurf[37]. Statistisches Lernen besteht in dieser Situation lediglich darin, die Münze zu beobachten und nach jedem Wurf neu die Wahrscheinlichkeit zu bestimmen, mit der bisher Kopf geworfen wurde. Der Strukturvorschlag würde dennoch nicht ausschließlich darin bestehen, eine Wahrscheinlichkeit anzugeben, sondern würde zusätzlich eine VERTEILUNG prognostizieren und eine Verlässlichkeit der Verteilung angeben. Die Verteilung eines Münzwurfes ist zwar denkbar einfach, aber es wäre aus Perspektive des MLA prinzipiell durchaus möglich, dass zukünftig nicht nur Kopf und Zahl als Eingabedaten registriert werden. Relevant ist die Vorstellung einer auf dem Rand zum Liegen kommenden Münze, weil MLA mögliche implizite Annahmen der Nutzer gerade vermeiden sollen und können und häufig eine Wahrscheinlichkeit dafür abschätzen können, dass konkrete Strukturvorschläge auch bei künftigen Eingabedaten zutreffen werden. Eine Münze wird zwar in der Praxis nicht auf dem Rand landen, aber sie kann durchaus systematisch unterschiedlich oft Kopf und Zahl zeigen. Lediglich eine steigende Zahl von Münzwürfen kann die Verlässlichkeit einer Aussage über die Verteilung der Ergebnisse erhöhen. Allerdings ist auch die Wiederholung eines Zufallsexperimentes nicht unbedingt eine Verbesserung. In dem Fall, dass die Erstellung einer Wahlprognose auf Basis einer Telefonbefragung angestrebt wird, wird die Verlässlichkeit der Ergebnisse sinken, wenn dieselben Personen wieder und wieder angerufen werden.

Funktionsbeschreibung

Statistische Lernstrategien suchen Wege zur Entscheidungsfindung unter expliziter Berücksichtigung und Berechnung oder SCHÄTZUNG der Ungewissheit von Faktoren wie dem Vorwissen oder der Datenerhebung. Statistische Lernstrategien suchen typischerweise Strukturen in Datenbanken und dabei häufig Abhängigkeiten zwischen den Attributen der Rohdaten. Statistische Lernstrategien werden in unterschiedlichen Funktionen eingesetzt. Sie können verborgene Variablen oder Strukturen wie VERTEILUNGEN su-

37 Ein technischeres Beispiel zur Versicherung von Automobilen findet sich bei Dugas (Dugas et al. 2003).

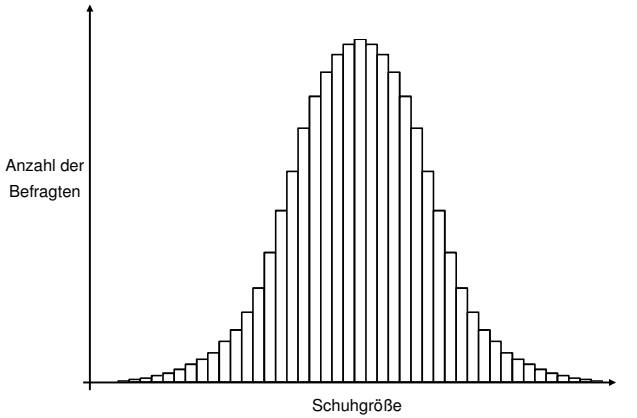
chen oder die Parameter von bereits bekannten oder vermuteten Verteilungen bestimmen. Rohdaten werden dabei immer als Stichproben interpretiert, die genommen wurden, um eine Vermutung zu bestärken oder zu schwächen.

Statistische Lernstrategien kommen in der Praxis des maschinellen Lernens sehr häufig zum Einsatz und können in Szenarien, in denen andere Lernstrategien besser geeignet erscheinen, als Maßstab dienen, an dem die Performanz anderer Lernstrategien gemessen werden kann.

Definition

Eine STATISTIK ist eine Reihe auf Basis einer STICHPROBE zu einem ZUFALLSEXPERIMENT berechneter beziehungsweise geschätzter Werte. Der wichtigste Begriff in diesem Zusammenhang ist derjenige der Verteilung. Dieser Begriff wurde in Abschnitt 2.2.1 bereits motiviert und verwendet, soll hier aber noch einmal etwas detaillierter dargestellt werden. Eine Verteilung ist ein Graph, der die einem Histogramm zugrunde liegenden Gesetzmäßigkeiten abbildet und für jedes mögliche Ergebnis des Zufallsexperiments die entsprechende Wahrscheinlichkeit angibt. Die Verteilung ignoriert die Schwankungen, die in der Realität entstehen, wenn ein Zufallsexperiment durchgeführt wird. Die wichtigste Verteilung in der Praxis und in der mathematischen Theorie ist die NORMALVERTEILUNG. Diese Verteilung beschreibt jedes Zufallsexperiment, das von vielen unabhängigen Faktoren beeinflusst wird und für das eine große Stichprobe genommen wurde. Nachfolgend ist das Histogramm für eine fiktive Befragung zur Ermittlung der Schuhgröße deutscher Männer abgebildet, die einem normalverteilten Zufallsexperiment entspricht.

Abbildung 30: Histogramm in Form einer Normalverteilung



Diese fiktive Umfrage ergibt genau die Form einer Normalverteilung, allerdings ist es wichtig zu verstehen, dass eine andere Verteilung angesichts konkreter Trainingsdaten durchaus wahrscheinlicher sein kann als die tatsächlich vorliegende Struktur. Eine Stichprobe ist nicht zwangsläufig repräsentativ für den Kontext, aus dem sie entstammt.

Die Erstellung eines Strukturvorschlages im Rahmen eines Autoadaptionsprozesses entspricht einer Entscheidung auf Basis einer Statistik.[38] Solche Entscheidungsfindungen sind mittels parametrischer, semiparametrischer und nichtparametrischer Ansätze möglich. Die Differenzierung dieser Ansätze wurde im Überblick zur Unterscheidung von Algorithmen bereits für das gesamte Feld des maschinellen Lernens erläutert. Der häufigste praktische Einsatz parametrischer oder nichtparametrischer Ansätze findet sich im statistischen Lernen. Statistisches Lernen mittels PARAMETRISCHER ANSÄTZE geht davon aus, dass die dem Zufallsexperiment zugrunde liegenden Verteilungen im Vorfeld bereits bekannt sind. Das bedeutet, in parametrischen Lernszenarien müssen die PARAMETER bestimmt werden, die bekannte Verteilungen auf das konkrete Zufallsexperiment anpassen[39]. Bei-

38 Diese Art der Entscheidung wird formal als STATISTISCHE INFERENZ bezeichnet.

39 In der Statistik wird an dieser Stelle der Begriff SCHÄTZER benötigt, um diejenigen Parameter zu benennen, die bestimmt werden müssen, bevor wiederum

spiele für solche Parameter sind der ERWARTUNGSWERT und die VARIANZ einer Verteilung[40]. Ein Beispiel für einen parametrischen Ansatz ist die Methode der maximalen Wahrscheinlichkeit, bei der für eine festgelegte Verteilung und eine konkrete Stichprobe diejenigen Parameter ermittelt werden, die der Entstehung einer Stichprobe der beobachteten Art die größte Wahrscheinlichkeit zuordnen. Die Anwendung dieser Methode verlangt in Konsequenz, dass sämtliche denkbaren Parameter getestet werden und ist dementsprechend aufwendig. Diese Vorgehensweise steht im deutlichen Gegensatz zu dem Ansatz, eine Güte- oder Fitnessfunktion zu definieren oder die Interessantheit einer Struktur zu bewerten, um im Autoadaptionsprozess die jeweils optimalen Parameter zu finden. SEMIPARAMETRISCHE ANSÄTZE setzen ein Vorwissen zu einem Teil der auftretenden Verteilungen voraus. Ein Beispiel für einen semiparametrischen Ansatz ist eine Clusteranalyse, bei der für manche Cluster von einer parametrisierbaren Verteilung ausgegangen wird und für andere nicht. Die NICHTPARAMETRISCHEN ANSÄTZE schließlich gehen davon aus, dass die Verteilungen im Vorfeld nicht bekannt sind und als Teil des Strukturvorschlages gesucht beziehungsweise erstellt werden müssen. Ein Autoadaptionsprozess wird auch dann als nichtparametrisch bezeichnet, wenn die Stichprobe nicht mittels einer Verteilung erklärt werden kann. Ein Beispiel für einen nichtparametrischen Ansatz liegt vor, wenn angenommen wird, dass keine systematische parametrisierbare Verteilung vorliegt, sondern nur die Eintrittswahrscheinlichkeiten unabhängiger Einzelergebnisse berechnet beziehungsweise abgeschätzt werden. Parametrische und nichtparametrische Ansätze werden zwar auf unterschiedliche Weise im Rahmen der Autoadaptionsprozesse realisiert, die Unterschiede liegen allerdings in erster Linie in der Wahl der eingesetzten mathematischen Methoden. Die prinzipiellen Vorgehensweisen sind jeweils vergleichbar. Die exemplarische Betrachtung einer Variante statistischen Lernens ist daher ausreichend, um ein prinzipielles Ver-

Aussagen zu den, häufig verborgenen, Parametern der Verteilungen möglich sind.

40 Der Erwartungswert beschreibt den Wert, der als mittlere Ausgabe erwartet werden kann, wenn mittels der Verteilung ein zufälliger Wert ausgegeben wird. Die Varianz beschreibt, wie stark die Verteilung um den Erwartungswert herum variiert, das heißt, wie sehr die Werte der Verteilung voneinander abweichen können.

ständnis des statistischen Lernens als einem Teilbereich des maschinellen Lernens zu erlangen. Die folgenden Darstellungen werden sich entsprechend auf das BAYES'SCHE LERNEN konzentrieren. Bayes'sches Lernen ist einerseits in der Praxis von großer Relevanz und basiert andererseits auf einer sehr kompakten mathematischen Grundlage. Die Betrachtung des Bayes'schen Lernens ermöglicht die Entwicklung einer nützlichen Intuition, wie mathematisches Hintergrundwissen im maschinellen Lernen zum Einsatz kommen kann. Im Weiteren wird zwischen der Grundform und einer modellbasierten Variante Bayes'schen Lernens unterschieden – zwischen dem ASSOZIATIONSLERNEN und den BAYES'SCHEN NETZEN.

Bayes'sches Lernen

Bayes'sches Lernen basiert wie die meisten Varianten statistischen Lernens auf der Arbeit mit einem speziellen Teil der mathematischen Begriffswelt und mit speziellen Methoden zum Umgang mit Wahrscheinlichkeiten. Der in diesem Fall zum Einsatz kommende und für die mathematische Theorie sehr grundlegende Begriff ist derjenige der BEDINGTEN WAHRSCHEINLICHKEIT. Angenommen, ein Biosupermarkt führt eine Befragung von Kunden durch und erkundigt sich, ob die befragte Person sich vegetarisch ernährt. Der Anteil der Kunden, die diese Frage bejahen, könnte nun sehr stark von Aspekten wie dem Bio-Sortiment des Supermarktes beeinflusst werden. Mit einer solchen Umfrage wird die Wahrscheinlichkeit gemessen, dass eine Person sich vegetarisch ernährt, unter der Bedingung, dass sie in diesem speziellen Biosupermarkt einkauft. Diese Statistik wird als eine bedingte Wahrscheinlichkeit bezeichnet. Die bedingte Wahrscheinlichkeit eine Person zu interviewen, die sich vegetarisch ernährt, unterscheidet sich je nach Kontext der Interviewreihe möglicherweise erheblich.

Abbildung 31: Bedingte Wahrscheinlichkeit in Supermärkten

Die Bestimmung einer bedingten Wahrscheinlichkeit unterscheidet sich nicht von derjenigen einer UNBEDINGTEN WAHRSCHEINLICHKEIT. Im Falle der Messung einer unbedingten Wahrscheinlichkeit wird die Häufigkeit eines bestimmten Ereignisses bestimmt und in ein Verhältnis zur Gesamtzahl von Ereignissen gesetzt. Eine bedingte Wahrscheinlichkeit wird analog ermittelt. Der einzige Unterschied besteht darin, dass die Gesamtzahl von Ereignissen aufgrund einer expliziten Vorannahme reduziert wird und nur eine Teilmenge der Gesamtzahl von Ereignissen betrachtet wird. Die Bedeutung solcher bedingten Wahrscheinlichkeiten ergibt sich wie folgt: sollte die Wahrscheinlichkeit Vegetarier anzutreffen in dem betrachteten Biosupermarkt deutlich größer sein, als es dem Anteil der Vegetarier an der Gesamtbevölkerung entspricht, so kann vermutet werden, dass der Besuch des Biosupermarktes nicht unabhängig von den Essgewohnheiten einer Person ist. Anders gesagt, bedingt die Wahl des Supermarktes die Essgewohnheiten und die Essgewohnheiten bedingen die Wahl des Supermarktes. Dieser Zusammenhang wird als ASSOZIATIONSREGEL bezeichnet und es kann nicht auf das Vorliegen einer Kausalität und insbesondere nicht auf die Richtung einer Kausalität geschlossen werden. Der typische Fehler an dieser Stelle besteht darin, zu vermuten, dass das Betreten eines Bio-Supermarktes Menschen zu einer vegetarischen Ernährung veranlasst. Ein Beispiel für die Aufklärung solch einer Fehlinterpretation besteht darin, dass eine Langfriststudie über mehrere Jahrzehnte durchgeführt wurde, um die Auffassung

zu widerlegen, dass Menschen aufgrund einer vegetarischen Ernährung eine höhere Lebenserwartung besitzen (Chang-Claude et al. 2005).

Bedingte Wahrscheinlichkeiten sind im statistischen Lernen von großer Bedeutung, weil bei der Suche nach der Hypothese, die das Ergebnis einer speziellen Stichprobe am wahrscheinlichsten erklärt, in erster Linie solche bedingten Wahrscheinlichkeiten bestimmt werden müssen. Die mathematische Theorie reduziert mit dem SATZ VON BAYES für die Suche nach der wahrscheinlichsten Hypothese die Anforderungen auf zwei Statistiken. Erstens muss die unbedingte Wahrscheinlichkeit der Hypothese selbst bestimmt werden und zweitens muss die bedingte Wahrscheinlichkeit bestimmt werden, dass die Trainingsdaten auftreten – unter der Bedingung, dass die Hypothese richtig ist[41].

Praktische Anwendungen Bayes'schen Lernens finden sich bei vielen Varianten von Produktempfehlungen. Eines der bekanntesten Beispiele ist der Onlineversand von Amazon. Dem Kunden wird bei Amazon zu jedem Kauf vorgestellt, welche Artikel von anderen Kunden zusammen mit den vom ihm selbst gekauften Artikeln erstanden wurden. Das heißt, der Lernvorgang besteht in diesem Fall nur darin, zu jedem Artikel eine Liste der ebenfalls gekauften Artikel zu aktualisieren und diese Liste bei Bedarf anzuzeigen. Die dahinterliegende Idee besteht in der Messung einer bedingten Wahrscheinlichkeit. Wenn die Wahrscheinlichkeit des Kaufes eines Artikels unter gewissen Bedingungen höher ist, möchte der Versandhandel die Artikel genau dann bewerben, wenn diese Bedingungen gerade eingetreten sind. Das Vorliegen einer Kausalität spielt hierbei für den Händler keine Rolle und wird auch nicht vorausgesetzt, das Ziel ist der Verkauf eines weiteren Artikels. Generell sind WARENKORBANALYSEN zur Erstellung von Kundenprofilen und zielgruppengerechten Werbemaßnahmen typische Anwendungen für Bayes'sches Lernen. Das Beispiel des Onlineversands hat für den Händler den Vorteil, dass dort eine erfolgreiche Werbemaßnahme unmittelbar den Umsatz steigert und eine fehlgeschlagene Werbemaßnahme annähernd ohne Konsequenzen bleibt. In Kontexten hingegen,

41 Nicht benötigt wird etwa die bedingte Wahrscheinlichkeit, dass die Hypothese richtig ist, falls die Trainingsdaten aufgetreten sind. Der aus der mathematischen Theorie folgende abduktive Bias liegt darin, dass von der Korrektheit dieser beiden Statistiken ausgegangen werden muss und dass beide zumindest teilweise aus Vorwissen berechnet werden müssen.

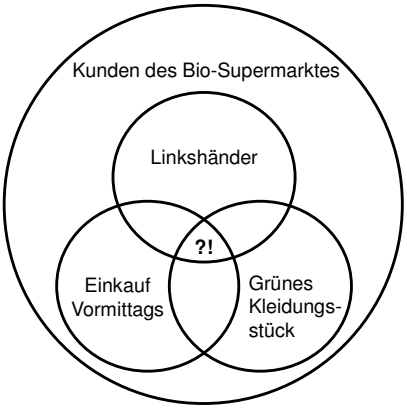
in denen medizinische Verträglichkeitsuntersuchungen durchgeführt werden oder die Verlässlichkeit von Diagnosen geprüft werden soll, sind fehlerhafte Strukturvorschläge sehr viel folgenreicher. Wichtig ist hier festzustellen, dass die Verfügbarkeit einer ausgearbeiteten mathematischen Theorie in der Praxis eine große Stärke statistischen Lernens darstellt, da beispielsweise Zuverlässigkeitsaussagen bezüglich der Strukturvorschläge möglich sind. Die Darstellung der beiden Hauptvarianten Bayes'schen Lernens bietet jedoch bereits genügend Gelegenheit zur Entwicklung eines interdisziplinären Verständnisses statistischen Lernens. Die Betrachtung des statistischen Lernens wird daher wie angekündigt im Rahmen der Betrachtung des Assoziationslernens und der Bayes'schen Netze erfolgen und es wird keine Diskussion der kontextabhängigen mathematischen Weiterentwicklungen angestrebt.

Assoziationslernen

Das Ziel des Autoadaptionsprozesses beim ASSOZIATIONSLERNEN besteht in der Suche nach Zusammenhängen in Form von bedingten Wahrscheinlichkeiten, den ASSOZIATIONSREGELN[42]. Die gesuchten bedingten Wahrscheinlichkeiten zu berechnen ist prinzipiell sehr einfach, da lediglich Häufigkeiten verglichen werden müssen. Die Herausforderung ergibt sich daraus, dass Assoziationsregeln Aussagen zu beliebig vielen der Attribute der Daten beinhalten können und es daher extrem viele mögliche Assoziationsregeln gibt. Angenommen, im Beispiel des Biosupermarktes ist wiederum die Wahrscheinlichkeit gesucht, dass ein Kunde sich vegetarisch ernährt. In diesem Fall könnte die Bedingung der Wahrscheinlichkeit sein, dass nur Kunden betrachtet werden, die Vormittags befragt wurden oder dass darüber hinaus nur diejenigen Personen relevant sind, die Linkshänder sind und zum Zeitpunkt des Interviews ein grünes Kleidungsstück getragen haben.

42 Entsprechend werden weder eine Verteilung noch Parameter einer solchen gesucht.

Abbildung 32: Bedingte Wahrscheinlichkeiten mit 3 Attributen



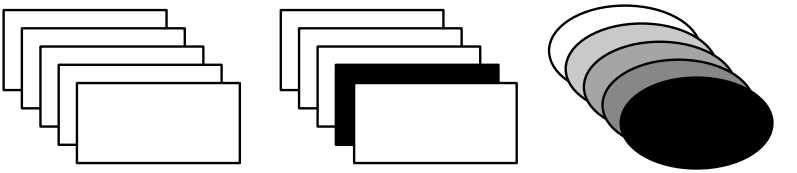
Hier wurde zuerst ein einzelnes und dann drei der Attribute der Trainingsdaten als Bedingung für die Wahrscheinlichkeit angesehen.

Assoziationsregeln weichen etwas von der Grundidee des statistischen Lernens ab, da alle korrekt berechneten Assoziationsregeln die Trainingsdaten gleich gut abbilden. Die Idee hinter der Erstellung einer Assoziationsregel ist weniger, die Eigenschaften zukünftiger Rohdaten prognostizieren zu können als vielmehr, interessante Eigenschaften der vorliegenden Trainingsdaten zu beschreiben. Wenn die Prognosefähigkeit im Fokus steht, werden entsprechend Methoden des instanzenbasierten Lernens mitbetrachtet und es wird versucht, interessante Assoziationsregeln zu identifizieren. Die dafür notwendige Suche nach interessanten Assoziationsregeln ist aufgrund der extrem großen Anzahl von denkbaren Assoziationsregeln sehr kompliziert[43]. Die unüberschaubare Anzahl denkbarer Bedingungen für Wahrscheinlichkeiten wird in der Praxis häufig mit Hilfe von einfachen Vorgaben drastisch reduziert. Beispielsweise kann verlangt werden, dass mindestens zwei Trainingsdaten auf einmal betrachtet werden und dass jede formulierte Aussage mindestens für diese beiden Trainingsdaten zutreffend ist. Die Interessantheit einer bedingten Wahrscheinlichkeit ergibt sich darüber hinaus nicht rein aus ihrer Größe. So könnte die Zahl der Kunden in

43 Die Möglichkeiten eine beliebige Teilmenge aus einer Gesamtmenge auszuwählen ergeben zusammen die Potenzmenge der Gesamtmenge. Diese wächst exponentiell mit der Anzahl der Elemente der Gesamtmenge.

der obigen Schnittmenge sehr klein sein und keine Kunden enthalten, die sich vegetarisch ernähren. Die bedingte Wahrscheinlichkeit wäre damit gleich null. Diese Assoziationsregel ist dennoch nicht sehr interessant, wenn die Aussage nur sehr wenig Kunden betrifft. Assoziationsregeln werden daher noch in einer zweiten Dimension bewertet, der Anzahl von Trainingsdaten, die von einer bestimmten Regel noch betroffen werden. Die nachfolgende Visualisierung zeigt ein Beispiel für eine Menge von Trainingsdaten, die als geometrische Objekte dargestellt sind.

Abbildung 33: Trainingsdaten als geometrische Objekte



In diesem Beispiel könnte eine Assoziationsregel lauten ›unter der Bedingung, dass ein schwarzes Objekt gewählt wird, ist die Wahrscheinlichkeit ein rechteckiges Objekt zu erhalten 50%‹. Diese Assoziationsregel betrifft allerdings nur zwei Trainingsdaten. Die Assoziationsregel ›unter der Bedingung, dass ein Rechteck gewählt wird, ist die Wahrscheinlichkeit ein weißes Objekt zu erhalten 90%‹ hingegen betrifft zehn Objekte und erreicht dennoch einen hohen Prozentwert. In den meisten Kontexten wäre eine Assoziationsregel der zweiten Art damit interessanter. Wichtig ist hier, dass auch 50% noch eine vergleichsweise große bedingte Wahrscheinlichkeit ist, da insgesamt fünf Farben in den Trainingsdaten vertreten sind und die schwarzen Objekte mit einem Anteil von zwei von fünfzehn nur 13% der Gesamtzahl von Objekten ausmachen.

Zusammengefasst kann die Suche nach Assoziationsregeln mit Hilfe der folgenden Schritte **A** bis **E** beschrieben werden.

A. Das MLA sucht nach Aussagen, die sehr spezielle Zusammenhänge zwischen einer kleinen Anzahl von Trainingsdaten beschreiben und berechnet die zugehörigen bedingten Wahrscheinlichkeiten.

Aussagen, die Anforderungen an besonders viele Attribute der Trainingsdaten stellen, betreffen meist nur sehr kleine Teilmengen der Trainingsdaten. Entsprechend ergeben sich aus den Trainingsdaten meist vergleichsweise

niedrige oder hohe bedingte Wahrscheinlichkeiten. Von Interesse sind im Autoadaptionsprozess die vergleichsweise großen bedingten Wahrscheinlichkeiten, die jedoch sehr wahrscheinlich nur Teile der Trainingsdaten betreffen. Das heißt, es sollte der Grad der Spezialisierung der Aussage gesenkt werden, auch wenn dadurch die bedingte Wahrscheinlichkeit sinkt.

B. Nach der Identifikation einer bedingten Wahrscheinlichkeit, die größer als ein im Vorfeld festgelegter Grenzwert ist, wird geprüft, ob die zugehörige Aussage eine Anzahl von Trainingsdaten betrifft, die größer als ein zweiter im Vorfeld festgelegter Grenzwert ist.
C. Wenn zu wenige Trainingsdaten betroffen sind, werden in der entsprechenden Aussage enthaltene Anforderungen verworfen, bis die Aussage entweder ausreichend viele Trainingsdaten betrifft oder die bedingte Wahrscheinlichkeit nur noch knapp über dem Grenzwert liegt und die Aussage verworfen wird.

Im obigen Beispiel könnte eine Aussage über vormittags einkaufende, linkshändige Personen, die grüne Hosen tragen, zu einer Aussage über vormittags einkaufende, linkshändige Personen werden. Es ist anzunehmen, dass die Menge an Supermarktkunden, die die genannten Merkmale aufweisen, durch die Entfernung der Forderung eines grünen Kleidungsstücks deutlich größer geworden ist. Es kann weiter angenommen werden, dass der Anteil der Menschen, die sich vegetarisch ernähren, sich durch die Vergrößerung der Gruppe stark verändert hat.

D. Wenn die resultierende Aussage eine Anzahl von Trainingsdaten betrifft, die größer als ein zweiter im Vorfeld festgelegter Grenzwert ist, wird die Aussage als Assoziationsregel bezeichnet, festgehalten und an den Nutzer übermittelt. Wenn die resultierende Aussage sich auf zu wenige Trainingsdaten bezieht, wird sie verworfen.
E. Der Prozess beginnt wieder mit Schritt **A**.

Das Verwerfen von Anforderungen ähnelt der Stutzung von Entscheidungsbäumen, wenngleich im statistischen Lernen deutlich andere Schwerpunkte gesetzt werden. Die Idee besteht nicht darin, die Rohdaten in Klassen aufzuteilen und es soll keine grafische Repräsentation erstellt werden. Stattdessen sollen Aussagen über bedingte Wahrscheinlichkeiten getroffen

werden und dazu ist es notwendig den Interessantheitsgrad von Aussagen zu bewerten. Die resultierende Suche nach interessanten Assoziationsregeln ähnelt wiederum dem instanzenbasierten Lernen und speziell der Subgruppenentdeckung, allerdings ist das Ausmaß dieser Ähnlichkeit veränderlich. Das Assoziationslernen wird in erster Linie über die Vorgabe der beiden genannten Grenzwerte beeinflusst – der Vorgabe einer minimalen Menge von Trainingsdaten, die von der Assoziationsregel betroffen sein müssen und einer minimalen Höhe für die bedingte Wahrscheinlichkeit. Mit Hilfe dieser Grenzwerte kann insbesondere gesteuert werden, ob die Assoziationsregeln auf die Gesamtmenge der Trainingsinstanzen anwendbar sind oder ob ihre Aussagen nur für spezielle Teilbereiche gültig sind. Diese Wahl entscheidet entsprechend, wie sehr sich im Assoziationslernen die Idee der Subgruppenentdeckung widerspiegelt.

Unabhängig davon, wie der Ablauf der Schritte von **A** bis **E** durch externe Vorgaben beeinflusst wird, ist es problematisch, diese Vorgehensweise als Autoadaptionsprozess zu bezeichnen. Zwar werden Strukturvorschläge ausgegeben und der Prozess orientiert sich an Trainingsdaten, allerdings wird dabei nur in geringem Ausmaß autoadaptiv vorgegangen. Die konkrete Durchführung von Schritt **C** erfordert ein Vorgehen, dass anderen Lernstrategien ähnelt, etwa bezüglich der Festlegung, in welcher Reihenfolge die Anforderungen fallen gelassen werden oder ob Anforderungen graduell oder vollständig fallen gelassen werden. Das Vorgehen erinnert dennoch insgesamt mehr an ein mathematisches Optimierungsverfahren als an ein Konzept, das eine Form von Selbstorganisation abbildet.

Bayes'sche Netze

BAYES'SCHE NETZE nutzen bedingte Wahrscheinlichkeiten, um lokale Zusammenhänge innerhalb der Trainingsdaten nachzubilden und setzen diese lokalen Zusammenhänge im Rahmen des Autoadaptionsprozesses zu einer globalen Struktur zusammen. Bayes'sche Netze basieren auf der Idee, grundlegendes Vorwissen über den Kontext unmittelbar in ihrer Struktur abzubilden und können entsprechend auch von einem unerfahrenen Nutzer leicht erstellt werden. Sogar unklare oder umstrittene Zusammenhänge können vom Nutzer sofort in die Struktur des Netzes integriert werden, so dass diese im Rahmen des Autoadaptionsprozesses geprüft werden können.

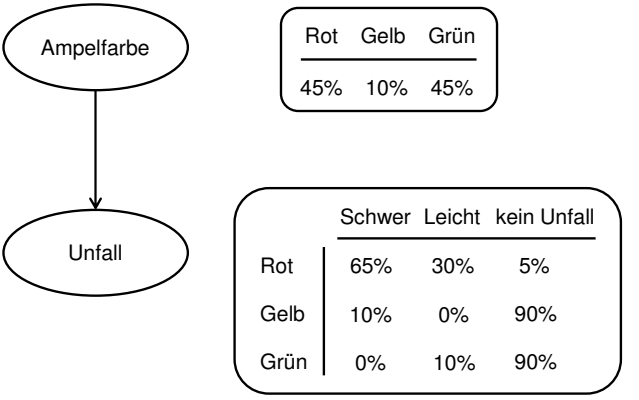
Die grafische Darstellung eines Bayes'schen Netzes entspricht einer Ansammlung von Knoten, die über gerichtete Verbindungen, die keine Zir-

kel bilden dürfen, miteinander verbunden sind. Die Darstellung entspricht der eines AZYKLISCHEN GRAPHEN und ähnelt formal der Darstellung von azyklischen künstlichen neuronalen Netzen. Die Methoden zur Bestimmung der Struktur und zum Umgang mit Bayes'schen Netzen spiegeln dies auch wider, allerdings dient die grafische Darstellung bei Bayes'schen Netzen der Erweiterung des Verständnisses der Nutzer und spiegelt deren Hintergrundwissen beziehungsweise deren Vermutungen wider. Die Knoten werden zu diesem Zweck jeweils mit einem EINFLUSSFAKTOR[44] identifiziert und die möglichen Zustände dieses Einflussfaktors werden in Form einer WAHRSCHEINLICHKEITSTABELLE festgehalten. Am einfachsten kann die Struktur Bayes'scher Netze an einem Beispiel dargestellt werden. Angenommen, es soll die Wahrscheinlichkeit abgeschätzt werden, bei der Überquerung einer Straße in der Nähe einer Kraftfahrzeug-Ampel einen Unfall zu erleiden. Diese Situation lässt sich als bedingte Wahrscheinlichkeit ausdrücken. Die bedingten Wahrscheinlichkeiten lassen sich dabei als Tabelle darstellen, in der neben den drei Ampelfarben[45] auch zwei unterschiedlich schwere Unfälle und ein unfallfreier Normalfall unterschieden werden. In der Visualisierung wird weiter angenommen, dass die beobachtete Ampel gleich lang ein rotes und grünes Signal zeigt, während nur in 10% der Zeit ein gelbes Signal zu sehen ist.

44 Dieser Faktor wird formal als ZUFALLSVARIABLE bezeichnet.

45 Es wurde angenommen, dass ›Gelb-Rot‹ aus Sicht der Fußgänger einem roten Signal entspricht.

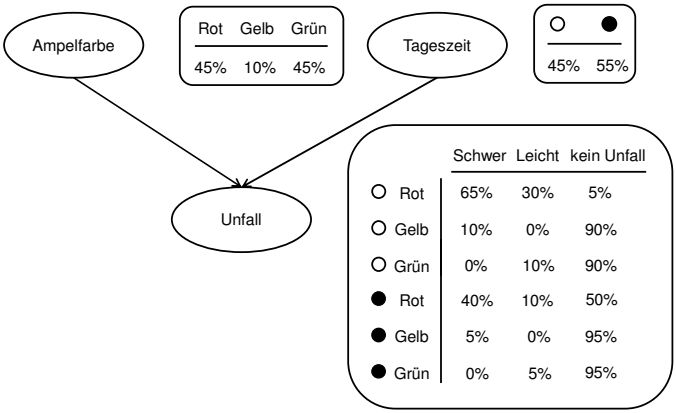
Abbildung 34: Bayes'sches Netz mit Wahrscheinlichkeitstabellen



Das Ergebnis dieses Netzes wäre die Erkenntnis, dass die Unfallgefahr bei gelbem und grünem Licht gleich groß ist, dass jedoch die Schwere des Unfalls unterschiedlich ist. Basierend auf diesem Netz kann nun geschätzt werden wie wahrscheinlich es ist, dass jemand angefahren wird.

Die gerichtete Verbindung zwischen dem Knoten der Ampelfarbe und dem Knoten des Unfalls gibt an, dass die Ampelfarbe die Schwere des Unfalls beeinflusst, allerdings nicht notwendigerweise kausal erzeugt, da es eine beliebige Anzahl von Einflussfaktoren für jeden Knoten geben kann. Bayes'sche Netze mit nur einem Einflussfaktor, wie es hier die Ampelfarbe war, stellen lediglich bedingte Wahrscheinlichkeiten dar, aber schon die Einbeziehung von einem zweiten Einflussfaktor lässt ein sehr komplexes Ergebnis entstehen. Die Unterscheidung von Tag und Nacht führt für das obige Beispiel zu folgender Struktur.

Abbildung 35: Bayes'sches Netz mit zwei Einflussgrößen



In diesem fiktiven Fall ist erkennbar, dass die Messreihe des ersten Beispiels wahrscheinlich tagsüber erstellt wurde und dass die Überquerung einer Straße in der Dunkelheit deutlich sicherer ist als bei Tageslicht. Von den beiden genannten Einflussfaktoren abgesehen, könnte es noch eine Vielzahl von nicht ohne Weiteres messbaren Einflussfaktoren geben, wie das durchschnittliche Verkehrsaufkommen am jeweiligen Kalendertag oder die mittlere Bremskraft eines Kraftfahrzeugs. Solche verborgenen Parameter, deren Existenz aus Vorwissen gefolgert wurde, können genau wie den messbaren Einflussfaktoren direkt zusätzlichen Knoten zugeordnet werden.

Ein praktisches Beispiel für den Einsatz von Vorwissen speziell in der medizinischen Diagnose ist die Annahme der Existenz einer Krankheit, die bestimmte beobachtbare Symptome erzeugt. Diese Krankheit selbst kann zwar nicht gemessen, aber dennoch als Einflussfaktor in ein Bayes'sches Netz aufgenommen werden. In diesem Aspekt grenzen sich Bayes'sche Netze wesentlich gegen das Assoziationslernen ab, bei dem die Darstellung von verborgenen Parametern nicht möglich war. Weiter werden verborgene Parameter im Gegensatz zu KNN transparent dargestellt und die Erstellung der Ihnen zugeordneten Knoten beruht direkt auf dem Vorwissen des jeweiligen Nutzers.

Das typischste Verfahren zur Realisierung Bayes'scher Netze ist der EXPECTATION-MAXIMIZATION-ALGORITHMUS[46], ein iterativer Autoadaptionsprozess zur Bestimmung verborgener Parameter, der auf der Vorgabe einer Netzstruktur aus Vorwissen aufbaut[47]. Die Bezeichnung des Algorithmus leitet sich direkt aus seinem Vorgehen ab:

A. Im Erwartungsschritt werden mittels der aktuell prognostizierten Wahrscheinlichkeitstabellen die zu erwartenden Werte der nicht messbaren Einflussfaktoren berechnet. Diese Parameterwerte werden für den nächsten Schritt behandelt, als wären sie gemessen worden und somit Teil der Trainingsdaten.
B. Im Maximierungsschritt werden die Wahrscheinlichkeitstabellen den Einflussfaktoren so angepasst, dass die Wahrscheinlichkeit des Auftretens der Trainingsdaten möglichst groß ist.
C. Anschließend wird wieder mit Schritt **A** fortgefahren, bis ein stabiler Zustand erreicht wird.

Kurz gesagt, wird immer abwechselnd angenommen, dass die angenommenen Wahrscheinlichkeitstabellen respektive die geschätzten Parameter korrekt vorliegen.

Die Hauptstärke Bayes'scher Netze liegt wie bereits beschrieben darin, komplexe Kontexte mittels gut verständlicher Aussagen zu Teilstrukturen zu modellieren. Insofern realisieren Bayes'sche Netze eine CLUSTERANALYSE, insbesondere kann der EM-Algorithmus in vielen Kontexten als ein

46 Kurz EM-ALGORITHMUS. Eine technische Darstellung, die mit dem EM-Algorithmus arbeitet und bei Vorliegen sehr guter mathematischer Vorkenntnisse einen interessanten Eindruck vermittelt, findet sich bei Friedman (Friedman 1998).

47 Die Netzstruktur selbst kann im Rahmen des Erwartungs-Maximierungs-Algorithmus ebenfalls adaptiert werden, indem mehrere Strukturen verglichen werden. Die Anzahl der möglichen Netzstrukturen wächst jedoch extrem schnell mit der Anzahl der Knoten, so dass es nicht möglich ist, alle zu überprüfen. Die Gegenmaßnahme hierzu besteht darin, dass die Einflussfaktoren im Gegensatz zu KNN nur dann über gerichtete Verbindungen mit anderen Einflussfaktoren beziehungsweise Knoten verbunden werden, wenn auch wirklich ein Zusammenhang vermutet wird.

Verfahren zur Clusteranalyse betrachtet werden. Das bedeutet jedoch nicht, dass Bayes'sche Netze einen Teilbereich des instanzenbasierten Lernens darstellen, da die Grundidee deutlich in der Nutzung der bedingten Wahrscheinlichkeiten zur verständlichen und intuitiven Darstellung von Zusammenhängen zwischen verschiedenen Einflussfaktoren liegt. Aus der interdisziplinären Perspektive ist festzuhalten, dass hier zwei unterschiedliche Lernstrategien zur Erstellung von Autoadaptionsprozessen durch verschiedene, aber doch vergleichbare Varianten eines Algorithmus realisiert werden, ohne dass die jeweilige Grundidee aufgegeben wird.

Eine Schwäche Bayes'scher Netze, die über das statistische Lernen hinaus von Interesse ist, stellt der Effekt des HINWEGERKLÄRENS dar. Dieser Effekt äußert sich beispielsweise darin, dass wenn – unabhängig von dem Beispiel der Verkehrssicherheit – bereits *bekannt* ist, dass eine Straße nass ist, die möglichen Ursachen für die Nässe voneinander abhängig sind beziehungsweise werden. Die Wahrscheinlichkeit, dass es geregnet hat, sinkt beispielsweise, wenn zusätzlich bekannt ist, dass gerade jemand sein Auto gewaschen hat. In diesem Fall ist der Fakt, dass die Straße nass ist, bereits erklärt und kann zur Beantwortung der Frage, ob es geregnet hat, nicht mehr unmittelbar genutzt werden. Eine Netzstruktur, die diesen Einfluss der Autowäsche auf das Wetter abbildet, ist weder falsch noch unplausibel, aber sie postuliert mehr bedingte Wahrscheinlichkeiten beziehungsweise Zusammenhänge als zur Modellierung des Kontextes benötigt würden. Ein plausibles und inhaltlich korrektes Bayes'sches Netz muss entsprechend noch nicht sinnvoll nutzbar sein. Diese Schwäche basiert auf einer impliziten Annahme bei der Konstruktion Bayes'scher Netze: sowohl formal als auch intuitiv gilt für jedes Paar von Einflussfaktoren ohne gerichtete Verbindung implizit, dass diese Einflussfaktoren UNABHÄNGIG voneinander sind. Im Rahmen der Erstellung des Bayes'schen Netzes können unterschiedlich strenge Anforderungen formuliert werden, die erfüllt sein müssen, bevor zwei Einflussfaktoren als unabhängig angesehen werden können. Beispielsweise kann gefordert werden, dass die Unabhängigkeit experimentell überprüft werden muss. Tatsächlich kann sogar im genannten Beispiel angenommen werden, dass eine Autowäsche vermutlich nicht an Tagen, an denen es regnet, stattfinden wird. Die Straße wird durch diese Verteilung auf mehrere Tage häufiger nass sein, als sie es wäre, wenn die beiden Einflussfaktoren völlig unabhängig voneinander wären. Der Versuch der Vermeidung solcher Fehler durch die Einfügung von zusätzlichen Zusammen-

hängen zwischen Einflussfaktoren führt häufig zu einer Überanpassung der Bayes'schen Netze.

Bayes-Klassifikatoren

Die Betrachtung von Assoziationslernen und Bayes'schen Netzen diente im Bisherigen der Darstellung des Einflusses der mathematischen Stochastik auf das maschinelle Lernen. Darüber hinaus lassen sich zwei der wichtigsten Klassifikatoren des maschinellen Lernens als Übergang von Assoziationslernen zu Bayes'schen Netzen verstehen. Diese beiden Algorithmen sind der NAIVE und der OPTIMALE BAYES-KLASSIFIKATOR.

Die diesen Klassifikatoren zugrunde liegende Beobachtung ist, dass das Hauptproblem bei der Manipulation und Einschätzung von Einflussfaktoren deren komplexe Abhängigkeiten voneinander darstellen. Wie bereits angedeutet wurde, kann das Wetter die Wahrscheinlichkeit einer Autowäsche beeinflussen oder ein Medikament kann in Kombination mit einem anderen Wirkstoff unerwünschte Nebenwirkungen entstehen lassen. Diese Abhängigkeit von Einflussfaktoren führt zu enorm komplexen Strukturen und erschwert das statistische Lernen. Der einfachste Ansatz zur Vereinfachung dieser Situation besteht darin, dass pauschal und mitunter wider besseres Wissen angenommen wird, dass die gemessenen Attribute der Trainingsinstanzen STATISTISCH UNABHÄNGIG sind. Zwei Größen sind statistisch unabhängig voneinander, wenn die bedingten Wahrscheinlichkeiten genau so groß sind wie die unbedingten Wahrscheinlichkeiten. Wenn etwa ein Studiengang die gleiche Geschlechteraufteilung aufweist wie die Gesamtbevölkerung, dann ist die Einschreibung in diesen Studiengang statistisch unabhängig von dem Geschlecht. Ein Gegenbeispiel wäre die Größenverteilung in einer professionellen Basketballmannschaft und in der Gesamtbevölkerung. Die Tätigkeit als Profibasketballer ist offenbar nicht unabhängig von der Körpergröße. Völlige statistische Unabhängigkeit liegt in der Realität fast nie vor, wird aber dennoch angenommen[48]. Mit Hilfe der

48 Natürlich können nur die Trainingsdaten als unabhängig angenommen werden, da sonst keinerlei Kanten im entstehenden Bayes'schen Netz enthalten wären, weil die Einflussfaktoren keine Einflüsse ausüben. Dennoch kann auch für die übrigen Größen eine etwas reduzierte Form von Unabhängigkeit angenommen werden: die BEDINGTE UNABHÄNGIGKEIT. Hierbei wird eine Größe als von ge-

aus dieser Annahme resultierenden Vereinfachungen bezüglich der bedingten Wahrscheinlichkeiten wird der Einsatz von neuen statistischen Autoadaptionsprozessen möglich. Das zentrale resultierende Verfahren zum Einsatz von bedingten Wahrscheinlichkeiten zur Klassifikation von Eingabedaten wird aufgrund der notwendigen, aber kontrafaktischen Vereinfachung als naiver Bayes-Klassifikator bezeichnet. Naive Bayes-Klassifikatoren beruhen wie die Suche nach interessanten Assoziationsregeln darauf, die Anzahl der zu betrachtenden bedingten Wahrscheinlichkeiten so gering wie möglich zu halten, nutzen jedoch Bayes'sche Netze zu Darstellung ihrer Struktur.

Die Auswertung eines Bayes-Klassifikators entspricht den nachfolgenden Schritten **A** bis **E**. Hier ist festzuhalten, dass gezielt die Auswertung und nicht die Erstellung eines Bayes-Klassifikators dargestellt wird, da die Klassifikatoren als solche für das maschinelle Lernen von größerer Bedeutung sind als die Methode ihrer Erstellung.

A. Die Attribute eines Eingabedatums werden als statistisch unabhängig betrachtet und einzeln ausgewertet.

B. Ein Attribut des Eingabedatums wird ausgewählt. Im weiteren Schritt **B** wird als Bedingung angenommen, dass das Attribut denjenigen Wert aufweist, den es für das Eingangsdatum annimmt. Davon ausgehend wird jeweils berechnet, wie groß die bedingte Wahrscheinlichkeit ist, dass das Eingabedatum einer der im Problemkontext vorgegebenen Klassen angehört.

Dieser Schritt ist praktisch schnell verständlich. Angenommen, Passanten sollen bezüglich der Wahrscheinlichkeit eingeschätzt werden, dass sie Teil eines professionellen Basketballteams sind. Zu diesem Zweck werden verschiedene Daten erhoben, unter anderem die Körpergröße, die Sprunghöhe und die Größe der Hände[49]. Im Schritt **B** wird zuerst die Größe der Hände des Passanten als Attribut ausgewählt und mit derjenigen aller gemessenen

nau einer Größe statistisch abhängig betrachtet, wie etwa im Falle einer Kausalität.

49 Alle drei Daten sind prinzipiell voneinander abhängig, können aber dennoch beim Einzelnen sehr stark abweichen, das bedeutet hier ist die Annahme einer statistischen Unabhängigkeit vielversprechend.

Profi-Basketballer und der Restbevölkerung verglichen. Die beiden Häufigkeiten werden in ein Verhältnis gesetzt und bilden die Wahrscheinlichkeit Profisportler zu sein unter der Bedingung, Hände einer gewissen Größe zu besitzen.

C. Der Schritt **B** wird für alle Attribute des Eingabedatums durchgeführt.
D. Jede Wiederholung von Schritt **B** erzeugt für jede mögliche Klassenzuordnung eine Wahrscheinlichkeit. Die entstandenen attributsabhängigen Wahrscheinlichkeiten werden kombiniert[50].
E. Die Klasse mit der größten kombinierten Wahrscheinlichkeit wird ausgewählt.

Die Nutzung von naiven Bayes-Klassifikatoren bildet neben Entscheidungsbäumen und KNN eines der oder sogar das im maschinellen Lernen am häufigsten eingesetzte Verfahren (Grieser et Fürnkranz 2006; Russell et al. 2007). Die resultierenden Klassifikatoren liefern häufig gute Ergebnisse, insbesondere solange die Einflussfaktoren in der Realität keinen zu starken Einfluss aufeinander ausüben. Varianten von Gewichtungen der Urteile einer Zusammenstellung naiver Bayes-Klassifikatoren bilden einige der effektivsten allgemein einsetzbaren Algorithmen im maschinellen Lernen, etwa bei Text-Klassifikationen im Rahmen derer Nachrichten bestimmten Themenfeldern zugeordnet werden sollen (Rennie 2001).

Die Motivation der Annahme statistischer Unabhängigkeit der Attribute bestand darin, dass anderenfalls eine zu große Anzahl von bedingten Wahrscheinlichkeiten denkbar und deren Berechnung sehr aufwendig wäre. Die Vorgehensweise des naiven Bayes-Klassifikators kann abgesehen von Schritt **A** formal auch umgesetzt werden, ohne die vereinfachende Annahme der statistischen Unabhängigkeit zu machen, wenngleich die Schritte **B** bis **D** in diesem Fall mathematisch einiger weiterer Erklärung bedürfen. Diese Lösung wird als OPTIMALER BAYES-KLASSIFIKATOR bezeichnet und sie erzielt im Mittel die besten Ergebnisse aller maschinellen Lernverfahren. Dieses Verfahren ist zwar in der Praxis selten realisierbar, da es extrem aufwendig ist sämtliche notwendigen Wahrscheinlichkeiten zu bewerten, der optimale Bayes-Klassifikator kann jedoch als eine Kenngröße verwen-

50 Die Wahrscheinlichkeiten werden multipliziert und das maximale Produkt wird im nächsten Schritt ausgewählt.

det werden, um in kleinen Testfällen die Performanz anderer Lernverfahren einschätzen zu können.

2.3.7 Analytisches Lernen

> »Beim analytischen Lernen wird versucht mit Hilfe eines vorgegebenen Wissens aus Beobachtungen Hypothesen abzuleiten. Diese Hypothesen können dann dem vorgegebenen Wissen hinzugefügt werden, um die Wissensbasis für zukünftige Lernvorgänge zu erweitern.«
>
> (Spix 1998)

Analytisches Lernen kann in verschiedenen Ausprägungen auftreten. Einerseits kann DEDUKTIVES LERNEN umgesetzt werden, das auf der Manipulation bereits bestehenden Vorwissens beruht, auf Messwerte verzichtet und neue Aussagen aus bisherigen Aussagen folgert. Andererseits kann eine induktiv-deduktive Mischform umgesetzt werden, die aus Beispielen allgemeine Regeln extrahiert. Im Weiteren wird primär auf die induktive logische Programmierung als ein Beispiel für solch eine Mischform eingegangen. Zwar sind beide Ausprägungen des analytischen Lernens im maschinellen Lernen realisierbar, der Verzicht auf Messwerte beim deduktiven Lernen macht dieses jedoch zu einem Grenzfall.

Induktive logische Programmierung

Motivation

Das Ziel der INDUKTIVEN LOGISCHEN PROGRAMMIERUNG, kurz ILP, besteht in der automatischen Generierung von Hintergrundwissen in Form von REGELN, die als logische Aussagen der Form ›alle Menschen kreisen um die Sonne‹ formuliert werden sollen. Während das Bayes'sche Lernen bedingte Wahrscheinlichkeiten als Mittel zur Struktursuche nutzt und Entscheidungsbäume als aussagenlogische Strukturen interpretiert werden können, manipuliert eine ILP Elemente der Prädikatenlogik, um in der Lage zu sein Regeln und damit Strukturvorschläge erstellen zu können[51]. Das Problem,

51 Die bei der ILP eingesetzten eingeschränkten Formen der Prädikatenlogik weisen eine höhere Darstellungskraft auf als Entscheidungsbäume, die auf Aussa-

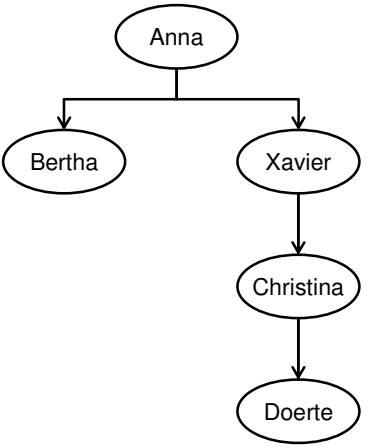
das durch den Einsatz von Prädikatenlogik gelöst wird, besteht darin, dass manche Zusammenhänge, die sich in den Trainingsdaten widerspiegeln, nicht ohne größere Umwege als Werte von Attributen beschreibbar sind, ein Beispiel ist der Zusammenhang ›ist Mutter von‹.

Die induktive logische Programmierung spielt wie das deduktive Lernen und die noch dargestellten Stützvektormethoden im zweiten Hauptteil keine zentrale Rolle. Aus diesem Grund wird sich die Darstellung darauf beschränken, herauszustellen, welche neuen Aspekte die induktive logische Programmierung der Diskussion des maschinellen Lernens beisteuern kann.

Einführungsbeispiel

Ein Beispiel für die Zusammenhänge, die in der ILP behandelt werden, ist der folgende Stammbaum.

Abbildung 36: Stammbaum als Einführungsbeispiel zur ILP



Angenommen, hierzu wäre als Hintergrundwissen bekannt, welche der Personen wessen Kind ist und welche der Personen weiblich sind. Der Zusammenhang beziehungsweise das Konzept ›ist Mutter von‹ hingegen soll gefunden werden. Die Erstellung dieses Konzeptes setzt das Vorliegen von Trainingsdaten in Form von positiven und negativen Beispielen voraus. Sei

genlogik basieren (Wikipedia Contributors 2012, Induktive logische Programmierung).

entsprechend angenommen, dass die folgenden beiden Zusammenhänge als wahr bekannt sind:

- ›Anna ist die Mutter von Bertha‹
- ›Christina ist die Mutter von Doerte‹

Weiter seien die folgenden beiden Zusammenhänge als falsch bekannt:

- ›Xavier ist die Mutter von Christina‹
- ›Anna ist die Mutter von Christina‹

Die ILP wird jetzt versuchen, anhand der Beispiele eine logische Regel für das Konzept ›A ist Mutter von B‹ zu entwickeln. Eine solche Regel würde voraussichtlich aussagen ›B ist Kind von A und A ist weiblich‹.

Funktionsbeschreibung und Definition

Häufig entwickelt die ILP Regeln, die Trainingsdaten korrekt prognostizieren und verwirft dann die entsprechenden Trainingsdaten. Diese ILP arbeiten daher in gewisser Weise gegensätzlich zum instanzenbasierten Lernen, das die Trainingsdaten möglichst und langfristig intensiv nutzen will. Der resultierende Autoadaptionsprozess des ILP kann wie folgt zusammengefasst werden.

A. Der gesuchte Zusammenhang wird mittels Trainingsdaten in Form von positiven und negativen Beispielen beschrieben.
B. Eine Aussagenmenge von Hintergrundwissen zu den Trainingsdaten wird etabliert.
C. Es wird eine Aussage gesucht, aus der sich mindestens ein positives Beispiel der Trainingsdaten herleiten lässt und aus der sich kein Negativbeispiel herleiten lässt[52].
D. Wenn im Schritt **C** eine Aussage gefunden wurde, wird sie zum Hintergrundwissen hinzugefügt, die Trainingsdaten werden um das Positivbeispiel reduziert und Schritt **C** wird wiederholt.

52 Formal sind hierbei nur Aussagen zulässig, in denen nur Variablen als Argumente vorkommen, da sonst wiederum eine zu große – rein formal eine unendlich große – Anzahl von Aussagen denkbar ist und geprüft werden muss.

E. Wenn im Schritt **C** keine Aussage gefunden wurde oder keine Positivbeispiele mehr zu erklären sind, wird der Prozess abgebrochen und das erweiterte Hintergrundwissen wird als Strukturvorschlag ausgegeben.

Im Allgemeinen liefert diese Vorgehensweise eine Regel, aus der sich einige der positiven, jedoch kein negatives Beispiel herleiten lassen. Die Forderung, dass keine negativen Beispiele herleitbar sein dürfen, kann gelockert werden, indem nur gefordert wird, dass die Anzahl der positiven Beispiele deutlich größer sein muss als die der negativen Beispiele[53]. Eine andere Modifikation des ILP ist es, zu fordern, dass das Hintergrundwissen vollständig im Rahmen des Autoadaptionsprozesses generiert werden muss.

Deduktives Lernen

DEDUKTIVES LERNEN erweitert bereits bestehendes Vorwissen, indem neue Aussagen aus bereits bekannten Aussagen gefolgert werden, und verzichtet dabei auf die Adaption des Vorwissens auf Basis von Messwerten. Deduktives Lernen versucht zwar, so wie das übrige maschinelle Lernen, Strukturen zu finden, die zu vorliegenden Trainingsdaten passen, allerdings ist es dennoch ein Grenzfall maschinellen Lernens, da die Strukturvorschläge ausschließlich aus dem im Vorfeld gegebenen Vorwissen abgeleitet werden. Die Trainingsdaten werden nur herangezogen um zu prüfen, welche der hergeleiteten Strukturvorschläge im konkreten Kontext relevant sind[54].

Das deduktive Lernen ist aus interdisziplinärer Perspektive insofern interessant, als es den Grenzbereich zwischen maschinellem Lernen und nichtlernenden Algorithmen beleuchtet. Aus diesem Grund soll anhand zweier Varianten deduktiven Lernens ein Einblick in diesen Teil des maschinellen Lernens ermöglicht werden.

53 Die Differenz der beiden Anzahlen wird als der HEURISTISCHE WERT einer Regel bezeichnet.

54 WISSENSLEVEL-LERNEN als Variante des deduktiven Lernens wird auch als RELEVANZBASIERTES LERNEN bezeichnet, allerdings ist die Einschätzung der Relevanz des Vorwissens allem deduktiven Lernen gemein – wenn auch in unterschiedlicher Ausprägung.

Erklärungsbasiertes Lernen

Das ERKLÄRUNGSBASIERTE LERNEN sucht in einem gegebenen Vorwissen nach einer Erklärung für ein Trainingsdatum. Anschließend wird diese Erklärung verallgemeinert und als Strukturvorschlag festgehalten.

> Ein Beispiel: 'Ein intelligenter Höhlenmensch brät seine erbeutete Eidechse an einem angespitzten Stock über einem Lagerfeuer, um sich seine Finger nicht am Feuer zu verbrennen. Seine weniger intelligenten Genossen, die zu diesem Zweck bislang ihre Finger benutzen, beobachten ihn dabei. Aus dieser Beobachtung und ihrem Hintergrundwissen können sie ableiten, daß man eine Eidechse rösten kann, ohne sich die Finger dabei zu verbrennen, indem man einen dünnen, spitzen Stock benutzt. Durch eine Generalisierung kommen sie zu dem Schluß, daß sich jedes Kleintier mit einem dünnen, langen, festen, spitzen Gegenstand gefahrlos über einem Feuer rösten läßt.'
>
> (Spix 1998)

Die Erkenntnis, dass Kleintiere sich auf diese Weise rösten lassen, lässt sich formal auch ohne die Beobachtung aus dem Vorwissen der Höhlenmenschen ableiten. Die Beobachtung dient eher einer Inspiration als der Vorwegnahme eines Experimentes.

Wissenslevel-Lernen

WISSENSLEVEL-LERNEN sucht formal nicht nach einer Erklärung für Trainingsdaten, sondern nach Anwendungskontexten für Vorwissen. Diese Variante deduktiven Lernens nutzt Trainingsdaten, um abstraktes Wissen mit Hilfe von Trainingsdaten zu konkretisieren.

> Ein Beispiel: 'Eine Reisende kommt erstmals nach Brasilien und trifft ihren ersten Brasilianer. Sie hört ihn Portugiesisch sprechen und erkennt, daß sein Name Fernando ist. Aufgrund ihres Vorwissens, daß innerhalb eines Landes die meisten Bewohner eine Sprache sprechen, folgert sie, daß Brasilianer Portugiesisch sprechen, jedoch folgert sie nicht, daß alle Brasilianer Fernando heißen, da Namensgleichheit nicht eine allgemeingültige Eigenschaft der Bewohner eines Landes ist.'
>
> (Spix 1998)

Die Menschen in Brasilien hätten ausgehend vom Vorwissen der Reisenden auch jede andere Sprache sprechen können, sie wusste zunächst nur, dass die ihr unbekannte Muttersprache für die meisten Einwohner identisch ist. Die Aussage, dass in Brasilien alle Menschen Portugiesisch sprechen, beinhaltet somit sowohl mehr als das Vorwissen als auch mehr als eine Beschreibung der Beobachtung, das heißt, die Trainingsdaten wurden in gewisser Hinsicht doch als Messwert genutzt.

2.3.8 Stützvektormethoden

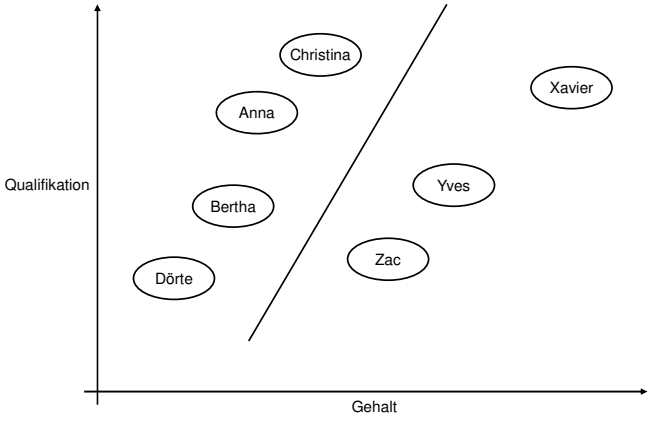
Motivation

Analog zu statistischem Lernen basieren auch STÜTZVEKTORMETHODEN nicht auf der Idee, eine Form von Selbstorganisation zu mathematisieren, sondern auf der Implementierung mathematischer Optimierungsverfahren in Kontexten, die von Eingabedaten abhängen. Die grundsätzlichen Ideen hinter den zugrunde liegenden Optimierungsverfahren wiederum entstammen der MATHEMATISCHEN OPTIMIERUNG und nicht der Informatik und sind für eine interdisziplinäre Betrachtung des maschinellen Lernens entsprechend nicht zentral. Im Gegensatz zu statistischem Lernen wird bei Stützvektormethoden darüber hinaus nicht auf dem Umgang mit einem auch außerhalb der Mathematik bedeutsamen Konzept – wie es die Wahrscheinlichkeit war – aufgebaut. Stützvektormethoden selbst sind dennoch in der Praxis von einiger Bedeutung, vor allem wenn klare Zielvorgaben formulierbar sind. Diese Situation kann auch durchaus eintreten, nachdem bereits ein anderer Ansatz des maschinellen Lernens eingesetzt wurde und die Performanz des Strukturvorschlages erhöht werden soll.

Einführungsbeispiel

Stützvektormethoden basieren auf der Trennung von Trainingsdaten. Eine Veranschaulichung für solch eine Trennung ist die nachfolgende Auswertung einer fiktiven Umfrage zur Entgeltgleichheit.

Abbildung 37: Fiktive Grafik zum Entgelt von Angestellten

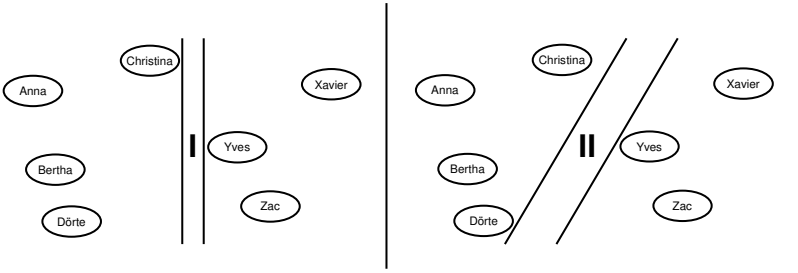


Hier können die Trainingsdaten ohne Schwierigkeiten getrennt werden und stehen somit bereit für eine weitere Analyse durch die Nutzer.

Funktionsbeschreibung

Stützvektormethoden suchen trennende Geraden, Ebenen oder höherdimensionale Ebenen, die als HYPEREBENEN bezeichnet werden. Die Nutzung von Hyperebenen setzt eine Codierung voraus, bei der die Trainingsdaten als Punkte in mehrdimensionalen Räumen identifiziert werden. Die Trennung mittels Hyperebenen ist insofern verwandt mit dem instanzenbasierten Lernen, als auch Stützvektormethoden einen Abstandsbegriff voraussetzen und den Suchraum in Teilräume aufspalten, um Eingabedaten klassifizieren zu können. Das Ziel der Trennung liegt häufig darin, eine Trennung mit einer möglichst großen neutralen Zone zwischen den getrennten Trainingsdaten zu identifizieren. Die Zone **II** in der folgenden Visualisierung wird gegenüber der Zone **I** bevorzugt.

Abbildung 38: Lineare Trennung



Die dem Rand am nächsten liegenden Trainingsdaten werden als STÜTZVEKTOREN bezeichnet und sobald die Stützvektoren identifiziert wurden, spielen die übrigen Trainingsdaten für die Stützvektormethoden meist keine Rolle mehr – insofern sind die Unterschiede zum instanzenbasierten Lernen deutlich erkennbar.

Stärken und Schwächen

Stützvektormethoden sind in der Praxis insbesondere für Aufgaben in der Bilderkennung, wie etwa Handschrifterkennung, sehr gut geeignet. Im Praxisbeispiel zu künstlichen neuronalen Netzen wurde dieser Anwendungsfall bereits betrachtet, tatsächlich wurden jedoch sowohl KNN als auch Stützvektormethoden eingesetzt. Die Fehlerrate der betrachteten KNN konnte, wie bereits erwähnt, von 1,6% auf 0,7% verbessert werden, während die Fehlerrate bei Stützvektormethoden im gleichen Kontext ohne Berücksichtigung von Vorwissen zunächst bei 1,1% lag und auf 0,56% gesenkt werden konnte (Russell 2007, S. 914ff). Auch im Allgemeinen können Stützvektormethoden besonders in Kontexten eingesetzt werden, in denen die Trainingsinstanzen eine Vielzahl von Attributen aufweisen, die sich als Zahlenwert codieren und somit ohne Weiteres als Punkte im Raum interpretieren lassen – wie etwa die Farbe von Bildpunkten bei der Bilderkennung. Stützvektormethoden sind damit generell in ähnlichen Kontexten wie KNN einsetzbar. Ein Vorteil der Stützvektormethoden liegt darin, dass die Codierung von Daten als Punkte im Raum die Eingabereihenfolge der Attribute der Daten irrelevant werden lässt. Das wiederum impliziert, dass Strukturen, die sich in der Eingabereihenfolge verbergen, nicht ohne einen Umweg entdeckt werden können.

Eine weitere Schwäche der Stützvektormethoden besteht darin, dass eine zu große Anzahl irrelevanter Attribute nur schwer gehandhabt werden kann. Zwar können die Attribute formal berücksichtigt werden, allerdings ist in der Grundidee der Stützvektormethoden nicht mitgedacht, dass unterschiedliche Gewichtungen für die Abstände zwischen wichtigen und unwichtigen Attributen sinnvoll sein können. Das Auftreten irrelevanter Attribute führt damit bei Stützvektormethoden zu einer Überanpassung, da jede Abweichung und jeder Abstand als gleich bedeutsam beurteilt wird.

2.4 CHARAKTERISTIK DES MASCHINELLEN LERNENS

Der Übergang von der Darstellung der Perspektive der Informatik zur Verortung des maschinellen Lernens durch die Technikphilosophie kann entlang der Frage vollzogen werden, ob die Autoadaptionsprozesse des maschinellen Lernens einer Suche, Optimierung, Klassifikation oder keiner der drei Charakterisierungen entsprechen. Die im Vorherigen vollzogene Aufspaltung des maschinellen Lernens in Teilgebiete, die durch ihre zugrunde liegenden Ideen unterschieden wurden, bietet interdisziplinären Betrachtungen eine stabile Grundlage. Selbstverständlich wurden und werden in der Informatik über die genannten Varianten maschinellen Lernens hinaus noch eine Vielzahl von weiteren Ansätzen für maschinelles Lernen entwickelt. Unabhängig davon, wie erfolgreich oder prominent diese Ansätze relativ zu den vorgestellten Varianten maschinellen Lernens zu bewerten sind, liegt der Fokus der jeweiligen Entwicklung nur sehr selten auf der Entwicklung einer systematisch neuen Art maschinell zu lernen. Stattdessen werden Kombinationen verschiedener Lernstrategien, Weiterentwicklungen bestehender Algorithmen und insbesondere Anpassungen von MLA an konkrete Kontexte realisiert. Solche Maßnahmen haben – bisher – ebenso wie Mischformen der bereits beschriebenen Verhaltensweisen keine systematisch neuartigen Verhaltensweisen bei MLA entstehen lassen. In Hinblick auf die übergreifende Frage nach der Selbstorganisation und der Veränderung von Technik sind solche Anpassungen dementsprechend ebenso wenig von zentraler Bedeutung wie bezüglich der Diskussion der Fragen, wie ein Autoadaptionsprozess sich charakterisieren lässt. Die bisherige Darstellung der wesentlichen Konzepte zur Erstellung von Autoadaptionsprozessen stellt die Grundlage für die Einsicht dar, dass eine pauschale Antwort auf die Fragen nach der Prozesscharakteristik des maschinellen

Lernens als Gesamtgebiet nicht zielführend ist. Eine Antwort auf diese Frage ist hochgradig von dem betrachteten Teilgebiet des maschinellen Lernens abhängig.

Auch wenn eine pauschale Einordnung des maschinellen Lernens nicht sinnvoll möglich ist, können sehr wohl einige der in der Diskussion gebräuchlichen Begriffe als aus technikphilosophischer Sicht ungeeignet bestimmt werden. Die Identifikation von Autoadaptionsprozessen mit der nachträglich zugeschriebenen Funktion der aus den Autoadaptionsprozessen resultierenden Strukturvorschläge wie bei der Beschreibung eines MLA als KLASSIFIKATOR ist ein solcher Fall. Die Interpretation eines Autoadaptionsprozesses als eine SUCHE ist zwar prinzipiell möglich, bietet aus technikphilosophischer Sicht jedoch ebenfalls keinen großen Mehrwert, denn die Durchführung einer Suche impliziert die Vorgabe eines zu Suchenden und gewisser Bewertungskriterien. Eine Suche unterscheidet sich somit nur insofern von einer OPTIMIERUNG, als dass die Rede von einer Suche eine weniger systematische Vorgehensweise annimmt. Auch wenn die Unterscheidung zwischen einer Optimierung und einer Suche nicht unmittelbar hilfreich ist, vermittelt die Tatsache, dass im maschinellen Lernen der Begriff der Suche verwendet wird, einen Eindruck von der Denkweise der Informatik über das maschinelle Lernen. Einige Ansätze des maschinellen Lernens formulieren durchaus den Anspruch sich von einer reinen Optimierung abzusetzen. Eine mögliche technikphilosophische Entsprechung dieser Abgrenzungsversuche und der resultierenden Formen von Artefakten wird im Folgenden entworfen.