Der Wrapper-Ansatz
Wissensentdeckung - Entdeckungswissenschaft

Oliver Kramer
Email: su002916@access.uni-dortmund.de

16. März 2000

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Wissensentdeckung in Datenbanken 1
- 1.2 Maschinelle Lernverfahren 1
- 1.3 Das Merkmalsauswahlproblem (feature subset selection problem) . . . 1
- 1.4 Übersicht 1

2 ID 3 **2**
- 2.1 Der Algorithmus 2
- 2.2 Definitionen 3

3 Filteransätze **3**

4 Der Wrapper-Ansatz **5**
- 4.1 Aufbau des Wrappers 5
- 4.2 Die Suche im Merkmalsraum 5
- 4.3 Compound Operators 6
- 4.4 Overfitting 7
- 4.5 Leistungsvergleich 7
- 4.6 Schlußbemerkungen 8

1 Einleitung

1.1 Wissensentdeckung in Datenbanken

Die Aufgabe der Wissensentdeckung in Datenbanken (Knowledge Discovery in Databases = KDD) ist die Suche nach Regulariäten und interessanten Mustern in großen Datenbeständen, also zumeist großen Datenbanken. Dabei können die Datensammlungen Texte, WWW-Seiten oder auch Daten aus Experimenten umfassen.

Wissensentdeckung bedient sich Methoden aus den drei Bereichen Statistik, Datenbanken und maschinellem Lernen. In diesem Artikel geht es um den sogenannten Wrapper-Ansatz beim Einsatz maschineller Lernverfahren. Er bewegt sich vor allem im zuletzt genannten Bereich des maschinellen Lernens.

1.2 Maschinelle Lernverfahren

Bei maschinellen Lernverfahren wird eine Regelmenge oder ein Entscheidungsbaum mit Hilfe eines Lernsets, einer ausgewählten Teilmenge klassifizierter Daten, induziert. Danach wird das Lernergebnis mit Hilfe eines Testsets, einer anderen Teilmenge klassifizierter Daten, geprüft. Der Lernvorgang ist schließlich erfolgreich, falls die maschinelle Klassifikation identisch mit der vom Benutzer gegebenen ist.

1.3 Das Merkmalsauswahlproblem (feature subset selection problem)

Ein wesentliches Problem vor dem Lernprozeß ist die Auswahl einer geeigneten Teilmenge von Merkmalen, mit denen ein Lernalgorithmus trainiert wird.

Welche Teilmengen von Merkmalen oder Attributen sind am nützlichsten für den Induktionsalgorithmus und welche können ignoriert werden? Das Ziel der Merkmalsauswahl sollte sein, durch Konzentration auf eine bestimmte Teilmenge von Merkmalen die Qualität der entstehenden Klassifikationsregeln und die Geschwindigkeit des Lernvorganges zu verbessern. Dies wird durch die Berücksichtigung des Informationsgewinns eines jeden Merkmals erreicht. Nach Wahl einer Merkmalsuntermenge lernt der Induktionsalgorithmus schließlich aus der Trainingsmenge - wie sonst auch - die Klassifikationsregeln. Als Methoden der Merkmalsauswahl bieten sich die Filteralgorithmen und der sogenannte Wrapper-Ansatz von Ron Kohavi und George H. John an. Er ist eine mögliche Lösung des sogenannten Merkmalsauswahlproblems mit dem Ziel der Leistungsoptimierung eines lernenden Algorithmus auf einer Trainingsmenge. Diese Methode sucht nach einer optimalen Merkmalsmenge passend zum lernenden Algorithmus und zur Domäne.

1.4 Übersicht

Im Kapitel 2 wird beispielhaft das induktive top-down Lernverfahren mit Entscheidungsbäumen ID3 beschrieben. Darauf folgt eine kurze Beschreibung der Filteransätze in Kapitel 3.

Schließlich erfolgt in Kapitel 4 die Beschreibung des Wrapper-Ansatzes. Die Leistungen der Lernverfahren unter Einsatz des Wrappers in der Praxis werden anhand verschiedener Datensätze mit den Leistungen der reinen Lernalgorithmen mit oder ohne Einsatz von Filtern verglichen.

2 ID 3

2.1 Der Algorithmus

Im folgenden sei der Algorithmus ID3, ein klassisches induktives top-down Lernverfahren aus Beispielen mit statistischer Merkmalsselektion - beschrieben [Morik, 1995]. Das Ziel dieses Lernverfahrens ist die Erstellung eines Entscheidungsbaumes als Erkennungsfunktion. An den Kanten des Baumes stehen Attributwerte, die Knoten sind Verzweigungspunkte und die Blätter Begriffsnamen. Ein neues Beispiel wird klassifiziert, indem man sich für den Pfad entscheidet, dessen Kanten dem Attributwert des Beispiels entsprechen. Erreicht man so ein Blatt, wird das Beispiel dem Begriff des Blattes zugeordnet.

Gegeben ist eine Menge von Beispielen in einer Attribut-Werte-Repräsentation und das Ziel ist die Erstellung eines Entscheidungsbaums zur Klassifikation neuer Objekte. Der Kern des Verfahrens liegt in der Bewertung des Informationsgewinns eines Attributes für die Klassifikation eines Objektes. Aus der Liste aller Attribute wird zunächst das mit dem größten Informationsgewinn gewählt, angewandt und dann aus der Liste entfernt. So wird mit verbleibenden Attributen weiter verfahren - bis kein Attribut mehr Informationen liefert oder die Liste leer ist. Den Informationsgehalt liefert die Entropie: $-\sum^{n} p_m \log_2 p_m$

wobei n die Anzahl verschiedener Attributwerte ist und p_m die Wahrscheinlichkeit für den m-ten Wert. Man vergleicht dann den Informationsgehalt der verschiedenen Attribute mit dem Informationsgehalt der Beispielmenge ohne Attribute und wählt das informativste Attribut oder gar keines.

Im einzelnen lautet der Algorithmus folgendermaßen:

ID3(K,C,A)

Knoten K, Menge von Beispielen C und Liste von Attributen A

- Wenn
  - Attributliste leer ist und nicht alle Beispiele in C gehören zum gleichen Begriff, dann Fehlermeldung.
  - alle Beispiele in C zum gleichen Begriff gehören, dann ist K ein Blatt und wird mit dem Begriff beschriftet.
  - C leer ist, dann Fehlermeldung
- Wähle das Attribut a mit maximalem Informationsgewinn.
- Reduziere die Attributliste A um a. Wir erhalten A'.
- Für alle Attributwerte w_i von a:
  - Hänge Knoten K_i an K und beschrifte ihn mit w_i.
  - Ermittle Mengen der Beispiele C_i aus C, deren Attribut a den Wert w_i hat.
- Rufe ID3(K_i,C_i,A') rekursiv auf.

2.2 Definitionen

Wichtig für die Beurteilung der Merkmalsselektion sind noch die folgenden formalen Definitionen der optimalen Lernmenge, der starken und der schwachen Relevanz.

Optimale Lernmenge: Sei I ein Induktionsalgorithmus und D eine Lernmenge mit Merkmalen $X_1,X_2,...,X_n$. Eine optimale Lernmenge von Merkmalen, X_{opt}, ist eine Teilmenge von Merkmalen, die die Genauigkeit des induzierten Klassifizierers C=I(D) maximiert.

Bei der Beurteilung eines Merkmals kann man zwischen schwacher und starker Relevanz unterscheiden. Ein Merkmal heißt stark relevant, wenn seine Abwesenheit eine Leistungsverschlechterung einer optimalem Bayes-Regel verursacht. Es heißt demgegenüber schwach relevant, wenn es einerseits nicht stark relevant ist, aber in manchen Fällen zu einer verbesserten Genauigkeit einer optimalen Bayes-Regel führt. Formal lauten die Definitionen wie folgt:

Starke Relevanz: Sei S die Menge aller Merkmale und S_i sei S-$\{X_i\}$. Ein Merkmal X_i heißt stark relevant, wenn es Werte x_i, y und s_i gibt, für die $p(X_i = x_i, S_i = s_i) > 0$, so daß $p(Y{=}y|X_i{=}x_i, S_i{=}s_i)!{=}p(Y{=}y|S_i{=}s_i)$.

Schwache Relevanz: Ein Merkmal X_i heiß schwach relevant, wenn es nicht stark relevant ist und eine Teilmenge von Merkmalen S_i' von S_i existiert, für die einige x_i, y und s_i' mit $p(X_i = x_i, S_i' = s_i') > 0$ existieren, so dass $p(Y{=}y|X_i{=}x_i, S_i{=}s_i')!{=}p(Y{=}y|S_i'{=}s_i')$.

Es sei an dieser Stelle darauf hingewiesen, dass nach dieser Definition relevante Merkmale nicht zwingend in der Praxis optimal für einen Lernalgorithmus sind. Auf der Suche nach optimalen Merkmalen sollte man stattdessen den speziellen Lernalgorithmus berücksichtigen, ein Ansatz den der Wrapper erfolgreich beschreitet.

3 Filteransätze

Wie schon erwähnt, gibt es neben dem Wrapper-Ansatz die sogenannten Filteralgorithmen zur Merkmalsselektion. Diese Verfahren arbeiten als ein Folge von Bearbeitungsschritten, wobei die Merkmale unabhängig vom Induktionsalgorithmus verarbeitet werden. Aus der ursprünglichen Merkmalsmenge sucht zuerst der Filter nach bestimmten algorithmusspezifischen Gesichtspunkten eine Untermenge aus, die schließlich dem Induktionsalgorithmus übergeben wird.

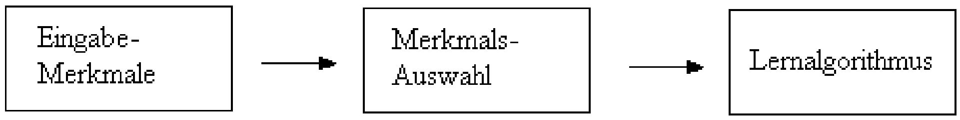

Abbildung 1: Filteralgorithmus. Die Merkmale werden unabhängig vom Algorithmus verarbeitet.

Als Beispiel für einen Filteransatz ist der FOCUS-Algorithmus ein Verfahren zur Merkmalsauswahl für rauschfreie Boolean-Domänen. Er prüft alle Merkmalsteilmengen und wählt die minimale Teilmenge aus, die ausreicht, um die Klassifikation für alle Instanzen der Trainingsmenge zu bestimmen.

Ein weiterer Filter ist der Relief-Algorithmus. Er weist jedem Merkmal ein Gewicht seiner Relevanz zu und versucht, sowohl stark wie auch schwach relevante Merkmale zu finden. Die sogenannten Tree-Filter wählen Merkmale mit Hilfe eines Entscheidungsbaumalgorithmus aus. Typisch ist die Verwendung des Nearest-neighbor-Algorithmus. Der Nachteil ist jedoch, dass Merkmale, die entscheidungsbaumgeeignet sind, nicht zwingend nearest-neighbor-geeignet sind. Desweiteren könnten aufgrund der Fragmentierung relevante Merkmale übersehen werden.

Filteransätze scheinen also keine ausgereiften Werkzeuge zur Merkmalsselektion zu sein. Ihr größter Nachteil ist, dass sie nicht den Lernalgorithmus berücksichtigen, sondern unabhängig davon operieren. Sie besitzen also nicht die Fähigkeit, im Hinblick auf möglichen Informationsgewinn ihre Merkmalsauswahl zu bewerten.

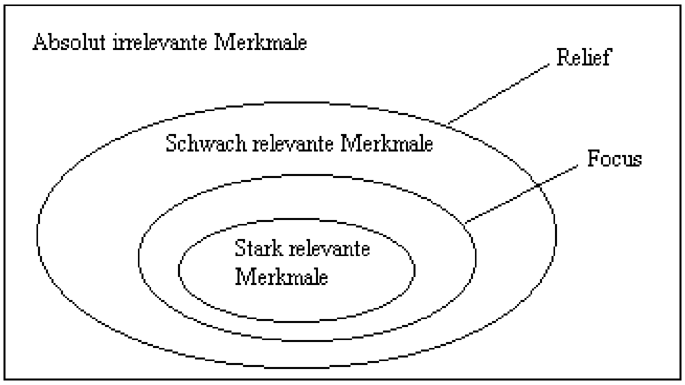

Abbildung 2: Merkmalsauswahl bei Filteralgorithmen.

4 Der Wrapper-Ansatz

4.1 Aufbau des Wrappers

Beim sogenannten Wrapper-Ansatz wird der Induktionsalgorithmus als Black-Box zur Bewertung von Merkmalteilmengen benutzt. Benötigt wird lediglich das Interface und kein weiteres Wissen über den Induktionsalgorithmus. Der Wrapper verwaltet die Suche nach einer guten Merkmalsmenge, indem er die vom Induktionsalgorithmus induzierten Regeln nach Genauigkeit bewertet.

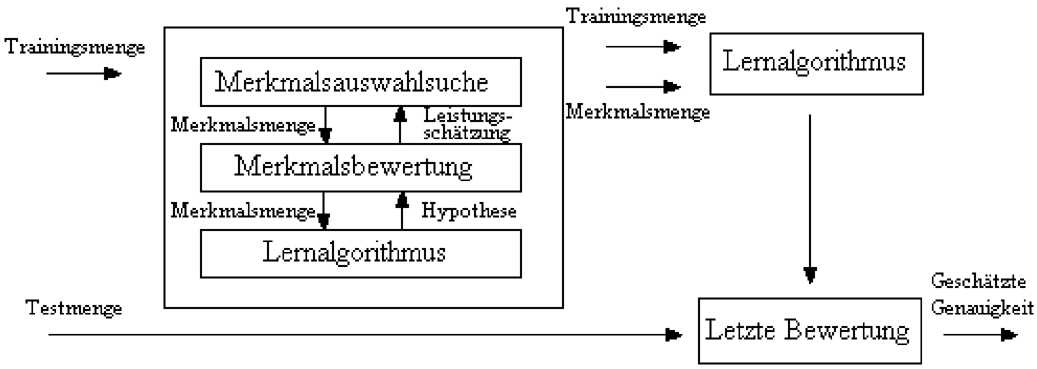

Abbildung 3: Schematische Darstellung des Wrapper-Ansatzes.

4.2 Die Suche im Merkmalsraum

Die Suche nach einer geeigneten Merkmalteilmenge findet in der Potenzmenge der Merkmale statt. Jeder Zustand des Zustandsraums der Wrapper-Suche repräsentiert also eine Merkmaluntermenge. Bei n Merkmalen, sind also n Bits in jedem Zustand zu finden: (1) für Merkmal anwesend und (0) für abwesend. Die einfachen Operatoren im Suchraum fügen ein einzelnes Element zu einem Zustand hinzu (add) bzw. entfernen es (delete). Die Grösse des Suchraumes für n Merkmale beträgt natürlich $O(2^n)$. Deshalb ist bei großen Suchräumen die vollständige Durchsuchung nahezu unmöglich.

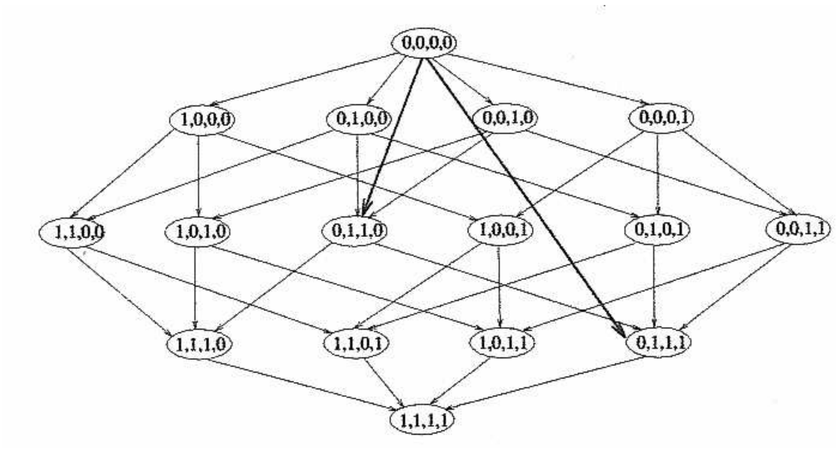

Abbildung 4: Beispiel eines Suchraumes mit nur vier Merkmalen.

Das Ziel der Suche ist der Zustand mit den besten Evaluationen unter Benutzung heuristischer Funktionen. Die Genauigkeit der induzierten Klassifikationsregeln wird geschätzt. Als Evaluationsfunktion wird häufig die n-fache Kreuzvalidierung mit mehrfacher Wiederholung benutzt, wobei die Standardabweichung der Genauigkeitsschätzung die Anzahl der Wiederholungen der n-fachen Kreuzvalidierung bestimmt. In den von Kohavi durchgeführten Experimenten wird eine weitere Kreuzvalidierung ausgeführt, wenn die Standardabweichung über 1% liegt und noch keine n Kreuzvalidierungen ausgeführt wurden.

Als Suchalgorithmus wird die Best-first-Suche verwandt. Die Idee es ist, den laut Schätzung vielversprechendsten bisher generierten Knoten, der noch nicht expandiert wurde, als die aktuelle Lösung zu behandeln. Da unser Problem ein Optimierungsproblem ist und keine explizite Abbruchbedingung existiert, kann die Suche jederzeit abgebrochen werden und die bisher beste Lösung zurückgegeben werden. Diese sollte sich natürlich eigentlich mit der Zeit verbessern. Deshalb formulierte Kohavi in seinen Experimenten die folgende Abbruchbedingung:

Trat während der letzten k Ausdehnungen kein Knoten auf mit einer geschätzten Genauigkeitsverbesserung des bisher besten Ergebnisses von mindestens e, dann breche die Suche ab.

In den Experimenten von John und Kohavi wurde k auf 5 und e auf 0,1% gesetzt.

4.3 Compound Operators

Der Zustandsraum wird durch Knoten organisiert, die Merkmalsuntermengen darstellen. Benutzt man nur die einfachen Operatoren add und delete, so kann sich die Suche ausgehend von der Anfangsmenge als Startzustand nur durch Generierung direkter Nachfolger zum Endzustand, der besten Merkmalsmenge, ausdehnen.Als Ausweg aus dieser Einschränkung entwickelten John und Kohavi die Compound Operators, dynamische Operatoren, die die direkte Verbindung zwischen vielversprechenden nicht direkt benachbarten Knoten herstellen. Z.B. wird bei der Backward-elimination-search ausgehend von der Menge aller Merkmale bei einem einfachen Operator jeweils ein Merkmal gelöscht, wenn alle Kinder dieses Knotens bewertet wurden. Bei i irrelevanten Merkmalen von insgesamt f müssen i*f Knoten bewertet werden. Ähnliches gilt für die Vorwärtssuche. Compound Operators werden nach Bewertung aller Standard-Kinder mit den einfachen Operatoren add und delete dynamisch erzeugt und für nur eine Knotenausdehnung verwendet. Sie vereinen nun genau die Operatoren, die zu den besten Kindern geführt haben und wenden diese in nur einem Schritt an, um sich der vielversprechendsten Lösung so schnell wie möglich zu nähern. So werden erfolgversprechende entferntere Knoten des Suchraumes schneller erreicht. Dieses Verfahren läuft solange, bis die geschätzte Genauigkeit sich nicht verbessert, so wie in der Abbruchbedingung formuliert.

Compound Operators beschleunigen also die Suche nach dem besten Knoten und machen Backward Feature Subset Selection erst vernünftig durchführbar. Bei den Experimenten von John und Kohavi wurden mit Compound Operators und dem Entscheidungsbaum-Algorithmus C4.5 in einer realen DNA-Datenmenge 3555 Knoten bewertet und eine Menge von 12 Merkmalen ausgesucht. Ohne die eingeführten Operatoren hätte der Algorithmus 30.240 Knoten für dieselbe Untermenge berechnen müssen.

Trotzdem ist die Merkmalsauswahl langsam. C4.5 benötigt für das gerade beschriebene Problem 1.5 Minuten für einen Durchlauf. Der Wrapper muß C4.5 jedoch 5 mal für jeden Knoten laufen lassen und es gibt bei diesem Problem hunderte Knoten.

4.4 Overfitting

Sind die Lernergebnisse zu genau an die Beispiele angepaßt, werden die Voraussagen bei der Testmenge ungenauer. Die Varianz, also die Streuung um den Erwartungswert, wird größer und im ganzen steigt die Anzahl der Fehler. Übermäßige Genauigkeitsschätzungen verursachen diesen Effekt. Wenn z.B. bei zufallsgenerierten Mengen mit zufälligen Merkmalen und Klassen bei einer Menge von 100 Instanzen die geschätzte Genauigkeit nach 300 Knotenschätzungen auf über 75% steigt, so ist dies ein deutliches Zeichen für Overfitting.

Experimente zeigen, dass Overfitting vor allem bei einer kleinen Anzahl von Beispielen auftritt.

4.5 Leistungsvergleich

John und Kohavi führten auf verschiedenen Datenmengen Experimente zum Leistungsvergleich zwischen Lernverfahren ohne Merkmalsauswahl, mit einem Filteransatz und mit dem Wrapper-Algorithmus durch.

Die genauen Ergebnisse sind [Kohavi et al, 1999] zu entnehmen. Hier seien nur die wichtigsten Ergebnisse zusammengefaßt.

Der Lernalgorithmus C4.5 konnte weder mit dem Relieved-Filter noch mit dem Wrapper verbessert werden. Dieses Ergebnis ist angesichts der Tatsache, dass C4.5 als modifizierter ID3 selbst eine Merkmalsauswahl verwendet, nicht überraschend.

In realen Datenbeständen verschlechterte sich sogar die Genauigkeit mit Relieved-Filter und verbesserte sich nur leicht mit dem Wrapper. John und Kohavi zufolge arbeitet der Relieved-Filter hervorragend auf künstlichen Datenmengen, die nur starke und gar nicht relevante Merkmale enthalten. Bei realen Datensätzen verbesserte sich die Genauigkeit der Lernregeln von 86.51% beim C4.5 gegenüber C4.5 mit Wrapper und Compound Operators auf 87.27% und bei künstlichen Datensätzen von 79.19% auf 84.68%.

Auch mit Naive Bayes wurden die Experimente durchgeführt. Hier zeigte sich, dass der Wrapper die Leistungen in 5 Datenmengen im Vergleich zu Naive Bayes ohne FSS signifikant verbesserte. Insgesamt wurden die Verbesserungen des Wrappers in keinem Fall vom Filter-Ansatz erreicht. Die Verbesserungen von Naive Bayes durch FSS sind alles in allem erstaunlich. Im oben schon einmal erwähnten DNA-Datensatz reduzierte der Wrapper die Fehlerrate von 6,1% auf 3,9%, was einer relativen Fehlerreduktion von 36% entspricht.

Bei realen Datensätzen verbesserte sich die Genauigkeit der Lernregeln von 87.94% beim reinen Naive Bayes gegenüber Naive Bayes mit Merkmalsauswahl und Comound Operators auf 88.68% und bei künstlichen Datensätzen von 77.96% auf 80.30%.

Auch die Anzahl ausgewählter Merkmale ist natürlich von Interesse. Da der Filter unabhängig vom Induktionsalgorithmus ist, schreibt er natürlich jedem Algorithmus die gleiche Anzahl an Merkmalen vor.

Der Relieved-Filter reduziert die Anzahl der Merkmale um 30%. Demgegenüber verringert der Wrapper die Anzahl der Merkmale noch einmal um bis zu 46% beim C4.5 und 28% beim Naive Bayes. C4.5 reduziert die Anzahl der Merkmale ohne Filter oder Wrapper schon alleine um 37%.

Insgesamt verbessert die Merkmalsauswahl die Ergebnisse signifikant, gerade in realen Domänen, wobei der Wrapper den Filtermethoden in fast allen Fällen überlegen ist. Naive Bayes erzielt zusammen mit Merkmalsauswahl erstaunlich gute Ergebnisse.

4.6 Schlußbemerkungen

Der Wrapper-Ansatz ist ein vielverwendeter Klassiker bei der Merkmalsauswahl in maschinellen Lernverfahren mit Induktionsalgorithmen. Der Wrapper sucht die relevanten und nützlichen Merkmale, um sie dem Lernalgorithmus zu übergeben. Die nützlichen Merkmale sind nicht unbedingt in Bezug auf obige Definition relevant, sondern vielmehr vom speziellen Lernalgorithmus abhängig. Compound Operators beschleunigen die Suche im Merkmalsraum und machen vor allem Backward FSS in Datenmengen mit vielen Merkmalen erst vernünftig durchführbar. John und Kohavi haben einer Methode einen Namen gegeben, die schon zuvor von vielen beim maschinellen Lernen genutzt wurde. Natürlich kann es noch Ansätze zur Verbesserung des Verfahrens geben, gerade bei der verwendeten Suchmethode, z.B. durch Verwendung genetischer Algorithmen oder nichtdeterministischer Ansätze. Auch die Parallelisierung der Knotenevaluation verspricht vor allem Geschwindigkeitsverbesserungen.

Die bei der Benutzung des Wrappers erzeugten Lernregeln sind übrigens in der Regel wesentlich übersichtlicher als die der rohen Induktionsalgorithmen und daher vom Menschen leichter interpretierbar.

Literatur

[Kohavi et al., 1999]Kohavi R., John G.H.: *The Wrapper Approach*. Silicon Graphics, Inc. 1999. Im WWW unter: www.robotics.standford.edu/ ronnyk.

[Morik, 1995] Morik, K: *Maschinelles Lernen*. In: Görz, G. (Hrsg.) et al.: *Einführung in die künstliche Intelligenz*. Bonn: Addison-Wesley 1995. S. 258-261.