Neue Entwicklungen im Data Mining mit Bayesschen Netzen

Rudolf Kruse und Christian Borgelt

Institut für Wissens- und Sprachverarbeitung
Otto-von-Guericke Universität Magdeburg
Universitätsplatz 2, D-39106 Magdeburg

e-mail: {kruse,borgelt}@iws.cs.uni-magdeburg.de

Kurzfassung: Durch Fortschritte in der Hard- und Softwaretechnologie ist es heute möglich, riesige Datenmengen zu erfassen und zu speichern, doch oft wird das in solchen Datensammlungen enthaltene Wissenspotential nicht voll genutzt. Um diesem Mangel abzuhelfen, entstand das Forschungsgebiet „Knowledge Discovery in Databases“, in dem es um die automatische Entdeckung von Wissen in Datenbanken geht. Wir behandeln in diesem Aufsatz ein spezielles Verfahren aus diesem Bereich — das Lernen Bayesscher Netze aus Daten — und gehen auf einige neuere Entwicklungen zur Verbesserung dieses Verfahrens ein.

1 Einleitung

In jedem Unternehmen gibt es heute Systeme zur elektronischen Datenverarbeitung, sei es in der Produktion, im Vertrieb, in der Lagerhaltung oder im Personalwesen. Jedes dieser Systeme benötigt, um seine Aufgabe erfüllen zu können, Daten, die entweder noch in einfachen Dateien oder schon in modernen Datenbanksystemen abgelegt sind. Diese Datenbanken entstanden aus der Notwendigkeit, bestimmte Informationen, wie z.B. die Adresse eines Kunden, schnell finden und verarbeiten zu können. Doch heute, mit dem Aufkommen immer leistungsfähigerer Rechner und durch Fortschritte in der Softwaretechnologie, kann man daran denken, solche Datensammlungen nicht mehr nur einfach zum Abrufen bestimmter, gerade benötigter Informationen zu nutzen, sondern auch nach in diesen Datenhalden verstecktem Wissen zu forschen. Erkennt z.B. ein Versandhaus in seinen Kunden bestimmte Gruppen, charakterisiert etwa durch Einkommen, Beruf, Geschlecht etc., so kann es u.U. wesentlich wirksamer für seine Produkte werben. Findet man in einem Supermarkt durch die Analyse der (mit Scannerkassen leicht erfaßbaren) Kassenbondaten heraus, daß bestimmte Produkte oft zusammen gekauft werden, so kann die Verkaufszahl u.U. durch eine entsprechende Anordnung dieser Produkte in den Regalen gesteigert werden.

Um solches Wissen aus Datenbanken zu gewinnen, reichen die Abfragemöglichkeiten normaler Datenbanksysteme und klassische Datenanalysemethoden oft nicht mehr aus. Mit ihnen lassen sich zwar beliebige Einzelinformationen aus einer Datenbank leicht abrufen, auch kann man einfache Aggregationen berechnen lassen (z.B. den durchschnittlichen Monatsumsatz im Raum Frankfurt im Jahre 1996) oder die Hypothese testen, ob der Wochentag Einfluß auf die Qualität der Produktion hat, doch allgemeinere Muster, Strukturen, Regelmäßigkeiten bleiben dabei unbemerkt. Gerade diese Muster können es jedoch sein, die sich z.B. zu einer Umsatzsteigerung ausnutzen lassen.

Es hat sich daher in den vergangenen Jahren ein eigenes Forschungsgebiet herausgebildet — oft mit den Begriffen „Knowledge Discovery in Databases“ (KDD) und „Data Mining“ (DM) bezeichnet —, in dem es um das automatische Erzeugen und Prüfen von Hypothesen und Modellen zur Beschreibung der in einem gegebenen (großen) Datenbestand vorhandenen Regelmäßigkeiten geht. Die so gefundenen Hypothesen und Modelle können dann, genau wie jedes andere Wissen auch, z.B. zur Prognose oder zur Entscheidungsfindung verwendet werden.

In diesem Aufsatz erläutern wir zunächst den KDD-Prozeß, in dem „Data Mining“ einen Schritt darstellt. Dieser Schritt, in dem die eigentliche Wissensgewinnung stattfindet, läßt sich am besten durch Angabe einer Menge von Aufgaben- oder Problemtypen charakterisieren. Anschließend gehen wir auf einen dieser Problemtypen, die Abhängigkeitsanalyse, ein und betrachten ein beliebtes Verfahren zu seiner Behandlung, das Lernen Bayesscher Netze bzw. allgemein probabilistischer graphischer Modelle aus Daten. Wir geben einen kurzen Überblick über die Grundlage Bayesscher Netze und mögliche Lernverfahren und gehen auf neuere Entwicklungen in diesem Bereich ein, speziell das Lernen der lokalen Struktur Bayesscher Netze.

2 Knowledge Discovery und Data Mining

Man kennt keine allgemeine, immer anwendbare und sicher zum Erfolg führende Methode, um Wissen aus Daten zu gewinnen. Doch es gibt einige Techniken, die oft anwendbar sind, und für viele praktische Probleme zu annehmbaren Ergebnissen führen. Diese Techniken werden im Foschungsgebiet „Knowledge Discovery in Databases“ systematisch untersucht. Ihren Einsatz beschreibt das folgende Schema des KDD-Prozesses.

2.1 Der KDD-Prozeß

In diesem Aufsatz wird der KDD-Prozeß in zwei Vor- und fünf Hauptstufen gegliedert, doch ist diese Gliederung keineswegs verbindlich. Ein einheitliches, allgemein anerkanntes Schema liegt bislang noch nicht vor.

Vorstufen

- Bestimmung des Nutzenpotentials
- Anforderungs- und Durchführbarkeitsanalyse

Hauptstufen

- Sichtung des Datenbestandes, Datenauswahl, ggf. Datenerhebung
- Vorverarbeitung (60-80% des Aufwandes)
  - Vereinheitlichung und Transformation der Datenformate
  - Säuberung (Behandlung von Fehlern, Fehlstellen und Ausreißern)
  - Reduktion / Fokussierung (Stichproben, Attributauswahl, Prototypenbildung)
- **Data Mining** (mit verschiedenen Verfahren)
- Visualisierung (auch parallel zu Vorverarbeitung, Data Mining und Interpretation)
- Interpretation, Prüfung und Bewertung der Ergebnisse
- Anwendung und Dokumentation

In den Vorstufen soll vor allem geklärt werden, ob die Hauptstufen des KDD-Prozesses überhaupt durchlaufen werden sollten. Denn nur wenn der potentielle Nutzen hoch genug, die Kosten einer Durchführung nicht zu hoch und die Anforderungen durch Data-Mining-Verfahren erfüllbar sind, kann mit einem Nutzen gerechnet werden.

In den Hauptstufen werden zunächst die Daten, die auf verborgenes Wissen hin untersucht werden sollen, ausgewählt und in eine Form gebracht, in der Data-Mining-Verfahren angewendet werden können. Dieser Vorverarbeitungsschritt ist gewöhnlich der aufwendigste des ganzen Prozesses. Abhängig von der in der Anforderungsanalyse festgestellten Data-Mining-Aufgabe (siehe unten) werden anschließend Entdeckungstechniken eingesetzt, deren Ergebnisse zur Prüfung und Interpretation visualisiert werden können. Da sich das gewünschte Ergebnis nur selten schon nach dem ersten Versuch ergibt, müssen einige Schritte der Vorverarbeitung (z.B. die Attributauswahl) und die Anwendung der Data-Mining-Verfahren ggf. mehrfach durchlaufen werden. Spätestens hier zeigt sich, daß KDD kein völlig automatisierter, sondern ein interaktiver Prozeß ist. Der Benutzer prüft und bewertet die erzielten Ergebnisse und nimmt, wenn nötig, Anpassungen am Ablauf des KDD-Prozesses vor.

2.2 Data-Mining-Aufgaben

Im Laufe der Zeit haben sich typische Aufgaben herauskristallisiert, die Data-Mining-Verfahren lösen können sollten. Zu diesen gehören vor allem die folgenden, die wir, neben ihrer Bezeichnung, durch eine typische Fragestellung charakterisieren.

- Klassifikation (classification)
  Ist dieser Kunde kreditwürdig?
- Konzeptbeschreibung (concept description)
  Welche Eigenschaften haben reparaturanfällige Fahrzeuge?
- Segmentierung (segmentation, clustering)
  Was für Kundengruppen habe ich?
- Prognose (prediction, trend analysis)
  Wie wird sich der Dollarkurs entwickeln?
- Abhängigkeitsanalyse
  (dependency/association analysis)
  Welche Produkte werden zusammen gekauft?
- Abweichungsanalyse (deviation analysis)
  Gibt es jahreszeitliche Umsatzschwankungen?

Am häufigsten sind Klassifikations- und Prognoseprobleme, da ihre Lösung unmittelbare Auswirkungen auf den Umsatz und den Gewinn eines Unternehmens haben kann. In letzter Zeit werden aber auch Abhängigkeitsanalysen immer öfter benötigt, z.B. wenn Verbundkäufe in Supermärkten (Warenkorbanalyse) oder Abhängigkeiten zwischen Ausstattungsmerkmalen und aufgetretenen Fehlern bei Kraftfahrzeugen untersucht werden sollen. Ein spezielles Verfahren zur Abhängigkeitsanalyse ist das Lernen Bayesscher Netze aus Daten. Auf dieses Verfahren gehen wir im folgenden näher ein.

3 Bayessche Netze

Zur Beschreibung eines Objektes oder eines Falles aus einem gegebenen Weltausschnitt benutzt man gewöhnlich eine Menge von Attributen, z.B. zur Beschreibung eines Autos den Hersteller, den Modellnamen, die Farbe, etc. Je nach Objekt oder Fall aus dem betrachteten Weltausschnitt nehmen diese Attribute bestimmte Werte an, z.B. VW, Golf, rot, etc. In der Wahrscheinlichkeitstheorie werden solche Attribute als *Zufallsvariable* aufgefaßt, die je nach *Elementarereignis* aus dem *Ereignisraum* bestimmte Werte annehmen. Nun sind bestimmte Wertkombinationen häufiger als andere, z.B. sind rote VW Golf häufiger als gelbe BMW Z1. Diese Häufigkeitsinformation wird als Verbundwahrscheinlichkeitsverteilung über dem kartesischen Produkt der Attributwertebereiche dargestellt, d.h. jeder Attributwertkombination wird eine Wahrscheinlichkeit zugeordnet. Da oft sehr viele Attribute notwendig sind, um die Fälle oder Objekte eines gegebenen Weltausschnitts zu beschreiben, verbietet es sich jedoch (wegen der Größe des sich dann ergebenden Raumes möglicher Attributwertkombinationen), diese Verbundverteilung direkt darzustellen. Man sucht daher nach Möglichkeiten, ihre Darstellung zu komprimieren. Zu diesen gehören die (eng miteinander verwandten) Bayesschen Netze [16] und Markovnetze [15].

Bayessche Netze basieren auf dem Produktsatz der Wahrscheinlichkeitsrechnung, der es erlaubt, eine strikt positive Verbundwahrscheinlichkeitsverteilung einer Menge von Zufallsvariablen in Produkte bedingter Wahrscheinlichkeitsverteilungen zu zerlegen. Für diskrete Zufallsvariablen, auf die wir uns im folgenden beschränken (wir nehmen sogar an, daß ihre Wertebereiche endlich sind), lautet dieser Satz ($A_1, \ldots, A_n$ seien die Attribute, $\mathrm{dom}(A_k)$, $k = 1, \ldots, n$, ihre Wertebereiche):

$$
\begin{aligned}
&\forall a_{i_1}^{(1)} \in \mathrm{dom}(A_1), \ldots, a_{i_n}^{(n)} \in \mathrm{dom}(A_n): \\
&P\left(A_1 = a_{i_1}^{(1)}, \ldots, A_n = a_{i_n}^{(n)}\right) \\
&\quad = P\left(A_n = a_{i_n}^{(n)} \middle| A_{n-1} = a_{i_{n-1}}^{(n-1)}, \ldots, A_1 = a_{i_1}^{(1)}\right) \\
&\quad\quad \cdot \ \ldots \\
&\quad\quad \cdot \ P\left(A_3 = a_{i_3}^{(3)} \middle| A_2 = a_{i_2}^{(2)}, A_1 = a_{i_1}^{(1)}\right) \\
&\quad\quad \cdot \ P\left(A_2 = a_{i_2}^{(2)} \middle| A_1 = a_{i_1}^{(1)}\right) \\
&\quad\quad \cdot \ P\left(A_1 = a_{i_1}^{(1)}\right)
\end{aligned}
$$

Die Anwendung des Produktsatzes allein führt jedoch noch nicht zu einer besseren Darstellung, eher zu einer schlechteren, da schon die erste bedingte Verteilung (der jeweils erste Faktor der Produkte) genauso umfangreich ist wie die Verbundverteilung selbst. Ist aber für den betrachteten Weltausschnitt eine Menge von bedingten Unabhängigkeiten

$$
\begin{aligned}
&\forall a_{i_1}^{(1)} \in \mathrm{dom}(A_1), \ldots, a_{i_k}^{(k)} \in \mathrm{dom}(A_k): \\
&P\left(A_k = a_{i_k}^{(k)} \middle| A_{k-1} = a_{i_{k-1}}^{(k-1)}, \ldots, A_1 = a_{i_1}^{(1)}\right) \\
&\quad = P\left(A_k = a_{i_k}^{(k)} \middle| \textstyle\bigcap_{A_j \in \mathrm{parents}(A_k)} A_j = a_{i_j}^{(j)}\right),
\end{aligned}
$$

bekannt, wobei $\mathrm{parents}(A_k) \subseteq \{A_1, \ldots, A_{k-1}\}$, so kann man die Produkte u.U. deutlich vereinfachen (vorausgesetzt die jeweiligen Mengen $\mathrm{parents}(A_k)$ der bedingenden Attribute sind klein). Diese Beziehungen heißen bedingte Unabhängigkeitsaussagen, weil sie ausdrücken, daß gegeben die Werte der Attribute in $\mathrm{parents}(A_k)$ das Attribut A_k von den restlichen Attributen in $\{A_1, \ldots, A_k\}$ unabhängig ist. Es ist klar, daß die erzielbare Vereinfachung von der Reihenfolge der Attribute bei Anwendung des Produktsatzes abhängt, da diese Reihenfolge bestimmt, welche bedingten Unabhängigkeiten überhaupt ausgenutzt werden können. Bei ungünstiger Wahl der Reihenfolge ist vielleicht keine Vereinfachung möglich, während sich bei günstiger Wahl die für die Faktoren benötigten Verteilungen u.U. erheblich verkleinern lassen.

Die sich ergebenden vereinfachten Produkte werden üblicherweise als gerichteter Graph dargestellt — man spricht deshalb auch von *graphischen Modellen* —, in dem es für jedes bedingende Attribut eine Kante zu dem jeweiligen bedingten Attribut gibt, d.h. von jedem Element von $\mathrm{parents}(A_k)$ zu A_k, für $k = 1, \ldots, n$. Dies erklärt auch die Bezeichnung $\mathrm{parents}(A_k)$, denn die in dieser Menge enthaltenen Attribute entpuppen sich so als die Elternknoten des Attributes A_k in diesem gerichteten Graphen (vgl. Abbildung 1). Jedem Knoten des Graphen wird die bedingte Wahrscheinlichkeitsverteilung des zugehörigen Attributes unter seinen Elternattributen zugeordnet.

Mit Hilfe eines solchen Graphen lassen sich auch Schlußfolgerungen ziehen, indem man Beobachtungen, d.h. Festlegungen der Werte einiger Attribute, entlang der Kanten des Graphen unter Verwendung der den Knoten zugeordneten bedingten Wahrscheinlichkeitsverteilungen propagiert.

Das Lernen eines Bayesschen Netzes besteht darin, eine gegebene mehrdimensionale Wahrscheinlichkeitsverteilung unter Verwendung der oben genannten Mittel in möglichst einfache Produkte zu zerlegen bzw. — äquivalent — einen möglichst einfachen Abhängigkeitsgraphen zu finden. Die zu zerlegende Verteilung ist dabei jedoch nicht direkt ge-

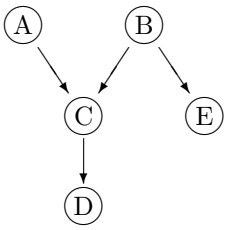

Abbildung 1: Ein einfaches Bayessches Netz, das die Zerlegung der gemeinsamen Verteilung über den Variablen $A, \ldots, E$ durch die Produkte $\forall a \in \mathrm{dom}(A), \ldots, e \in \mathrm{dom}(E) : P(A = a, \ldots, E = e) = P(E = e|B = b) \cdot P(D = d|C = c) \cdot P(C = c|B = b, A = a) \cdot P(B) \cdot P(A)$ darstellt.

geben, sondern es steht nur eine Datenbank von Beispielen zur Verfügung. Diese wird benutzt, um (bedingte) relative Häufigkeiten auszuzählen, aus denen dann die (bedingten) Wahrscheinlichkeiten geschätzt werden.

Ein Algorithmus zum Lernen Bayesscher Netze aus Daten besteht immer aus zwei Teilen: einem Bewertungsmaß und einer Suchmethode. Mit Hilfe des Bewertungsmaßes wird die Güte einer gegebenen Zerlegung (eines gegebenen Graphen) eingeschätzt, während die Suchmethode bestimmt, welche Zerlegungen (welche Graphen) überhaupt betrachtet werden. Oft kann das Bewertungsmaß auch benutzt werden, um die Suche zu steuern (z.B. bei einer Greedy-Suche), da es gewöhnlich das Ziel ist, seinen Wert zu maximieren (oder zu minimieren). Es gibt eine Vielzahl von Bewertungsmaßen, die zum Teil direkt für das Lernen Bayesscher Netze entwickelt wurden, zum Teil aus dem verwandten Lernen von Entscheidungsbäumen (vgl. den folgenden Abschnitt) übernommen werden können. Natürlich können wir diese Maße und die ihnen zugrundeliegenden Ideen hier nicht im Detail besprechen und verweisen daher auf [1, 2].

4 Lernen lokaler Struktur

Während man den gerichteten Graphen, der die in einem gegebenen Weltausschnitt geltenden bedingten Unabhängigkeiten wiedergibt, die *globale Struktur* eines Bayesschen Netzes nennt, bezeichnet man als *lokale Struktur* Regelmäßigkeiten, die eventuell in den den Knoten zugeordneten bedingten Wahrscheinlichkeitsverteilungen vorliegen. Es gibt verschiedene Ansätze, solche Regelmäßigkeiten auszunutzen, um zusätzliche, kontextspezifische

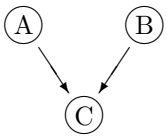

Abbildung 2: Ein Ausschnitt eines Bayesschen Netzes.

Eltern		Kind	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_2	$1 - p_2$
a_2	b_1	p_3	$1 - p_3$
a_2	b_2	p_4	$1 - p_4$
a_3	b_1	p_5	$1 - p_5$
a_3	b_2	p_6	$1 - p_6$

Tabelle 1: Eine bedingte Wahrscheinlichkeitsverteilung für den in Abbildung 2 gezeigten Ausschnitt eines Bayesschen Netzes.

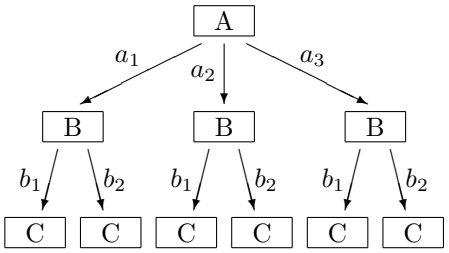

Abbildung 3: Ein voller Entscheidungsbaum für die Variable C zur Darstellung der bedingten Wahrscheinlichkeiten.

Unabhängigkeiten zu erfassen und dadurch ggf. das Ziehen von Schlußfolgerungen in Bayesschen Netzen zu verbessern. Zu diesen Ansätzen gehören Similarity Networks [11] und die verwandten Multinets [10], asymmetrische Darstellungen für die Entscheidungsfindung [20], probabilistische Horn-Klauseln [17], und (erstaunlicherweise erst seit kurzem, obwohl diese Art der Darstellung sehr naheliegend ist) außerdem Entscheidungsbäume [4] und Entscheidungsgraphen [6]. In diesem Aufsatz beschränken wir uns auf den Entscheidungsbaum- bzw. Entscheidungsgraphansatz.

Eine sehr einfache Art, eine bedingte Wahrscheinlichkeitsverteilung anzugeben, ist eine Tabelle, die für jede Kombination von Werten der bedingenden Attribute eine Zeile enthält, die die zugehörige

Eltern		Kind	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_1	$1 - p_1$
a_2	b_1	p_3	$1 - p_3$
a_2	b_2	p_4	$1 - p_4$
a_3	b_1	p_2	$1 - p_2$
a_3	b_2	p_2	$1 - p_2$

Tabelle 2: Eine bedingte Wahrscheinlichkeitsverteilung für den in Abbildung 2 gezeigten Ausschnitt eines Bayesschen Netzes mit einigen Regelmäßigkeiten.

Eltern		Kind	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_1	$1 - p_1$
a_2	b_1	p_2	$1 - p_2$
a_2	b_2	p_3	$1 - p_3$
a_3	b_1	p_3	$1 - p_3$
a_3	b_2	p_4	$1 - p_4$

Tabelle 3: Eine bedingte Wahrscheinlichkeitsverteilung für den in Abbildung 2 gezeigten Ausschnitt eines Bayesschen Netzes mit einer anderen Art von Regelmäßigkeiten.

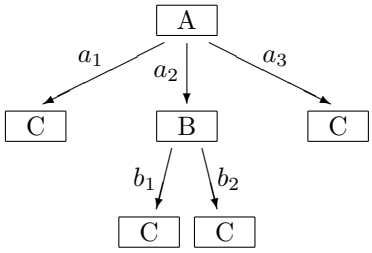

Abbildung 4: Ein vereinfachter Entscheidungsbaum für die Variable C, der die Regelmäßigkeiten der bedingten Wahrscheinlichkeitsverteilung aus Tabelle 2 ausnutzt.

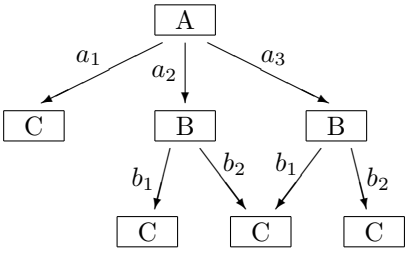

Abbildung 5: Ein Entscheidungsgraph, der die Regelmäßigkeiten in der bedingten Wahrscheinlichkeitsverteilung aus Tabelle 3 ausnutzt und zu dem es keinen äquivalenten Entscheidungsbaum gibt.

Wahrscheinlichkeitsverteilung über den Werten des bedingten Attributes angibt. Als einfaches Beispiel betrachten wir den Auschnitt eines Bayesschen Netzes, der in Abbildung 2 gezeigt ist (wobei wir annehmen, daß in diesem Netz das Attribut C keine anderen Eltern besitzt als die Attribute A und B).

Sei $\mathrm{dom}(A) = \{a_1, a_2, a_3\}$, $\mathrm{dom}(B) = \{b_1, b_2\}$ und $\mathrm{dom}(C) = \{c_1, c_2\}$. Dann müssen in diesem Netz im Knoten C die bedingten Wahrscheinlichkeiten $P(C = c_k \mid A = a_i, B = b_j)$ abgespeichert werden, z.B. wie in Tabelle 1 dargestellt. Die gleiche bedingte Wahrscheinlichkeitsverteilung kann aber auch in einem Baum abgespeichert werden, in dem die Blätter die Wahrscheinlichkeitsverteilungen über den Werten des Attributes C enthalten und jede Ebene innerer Knoten einem Elternattribut zugeordnet ist (siehe Abildung 3). Die Verzweigungen in diesem Baum sind mit den Attributwerten der bedingenden Attribute markiert und folglich stellt jeder Pfad von der Wurzel zu einem Blatt eine Kombination von Attributwerten dar. Offenbar ist ein solcher Baum äquivalent zu einem Entscheidungsbaum für die Variable C (wie er z.B. von dem Entscheidungsbaumlernprogram C4.5 [19] erzeugt werden könnte) mit den folgenden Einschränkungen: Alle Blätter müssen auf derselben Ebene liegen und in einer inneren Ebene des Baums muß auf allen Pfaden das gleiche Attribut getestet werden. Ein solchen Baum kann man auch als *vollen* Entscheidungsbaum bezeichnen.

Nehmen wir nun an, daß es in der bedingten Wahrscheinlichkeitsverteilung einige Regelmässigkeiten gibt, etwa die, die in Tabelle 2 dargestellt sind. Da diese Tabelle deutlich zeigt, daß der Wert des Attributes B nur dann von Bedeutung ist, wenn das Attribut A den Wert a_2 hat, können die Tests des Attributes B aus den mit den Werten a_1 und a_3 markierten Zweigen entfernt werden (siehe Abbildung 4). In der Baumdarstellung führen die Regelmäßigkeiten folglich zu einer deutlichen Vereinfachung.

Unglücklicherweise jedoch sind Entscheidungsbäume nicht mächtig genug, um alle denkbaren Regelmäßigkeiten wiedergeben zu können. Obwohl man große Flexibilität erreichen kann, indem man Änderungen der Reihenfolge der Attributtests,

Markierungen der Zweige durch Mengen von Attributwerten und mehrfache Tests des gleichen Attributs zuläßt, gibt es doch Regelmäßigkeiten, die sich mit einem Entscheidungsbaum nicht ausnutzen lassen, z.B. die in Tabelle 3 gezeigten.

Das Problem ist, daß der Test einer Variable in einem Entscheidungsbaum die Zeilen der bedingten Wahrscheinlichkeitstabelle in disjunkte Mengen zerlegt, die nicht wieder zusammengebracht werden können. Bei der in Tabelle 3 gezeigten Verteilung z.B. trennt ein Test des Attributes B die Zeilen 1 und 2, ein Test des Attributes A die Zeilen 4 und 5. Jeder der beiden Tests verhindert folglich, daß eine der beiden Gleichheiten in den bedingten Wahrscheinlichkeiten genutzt werden kann.

Diese Einschränkung kann jedoch aufgehoben werden, indem man zuläßt, daß ein Knoten mehr als einen Elternknoten hat, man also statt Entscheidungs*bäumen* Entscheidungs*graphen* verwendet [6]. In einem Entscheidungsgraphen lassen sich die Regelmäßigkeiten aus Tabelle 3 leicht ausnutzen (siehe Abbildung 5).

Die lokale Struktur eines Bayesschen Netzes läßt sich leicht aus Daten lernen, indem man in ähnlicher Weise wie beim Lernen von Entscheidungsbäumen vorgeht [19, 6, 2], also an jedem Knoten mit Hilfe eines lokalen Bewertungsmaßes nach dem besten Testattribut sucht, oder indem man zunächst einen vollen Entscheidungsbaum erzeugt und dann nach Vereinigungen von Blättern in diesem Baum sucht, die die Bewertung der bedingten Verteilung durch das zum Lernen verwendete Bewertungsmaß nur geringfügig verschlechtern oder sogar verbessern. In einem abschließenden Schritt kann man noch versuchen, die vorgenommenen Vereinigungen von Blättern zum Entfernen oder Zusammenfassen von inneren Knoten zu nutzen und so einen möglichst einfachen Entscheidungsgraphen zu erhalten.

5 Anwendung

Einige der oben beschriebenen Verfahren wurden von uns in dem Programmprototyp INES (Induktion von NEtzwerkStrukturen aus Daten) implementiert und, worauf wir in diesem Aufsatz jedoch nicht näher eingehen können, in einem Industrieprojekt in Zusammenarbeit mit dem Forschungszentrum Ulm der Daimler-Benz AG erfolgreich auf eine Mercedes-Benz Fahrzeugdatenbank angewandt. Dadurch konnten Abhängigkeiten zwischen Ausstattungsmerkmalen und Fehlern gefunden werden, die zu kennen den Fahrzeugexperten bei ihrer Ursachensuche hilfreich war [3].

6 Zusammenfassung

In diesem Aufsatz haben wir — nach einem Überblick über den KDD-Prozeß und einer Charakterisierung des Data-Mining-Schrittes dieses Prozesses durch eine Sammlung von Aufgabentypen — ein wichtiges Verfahren für die Abhängigkeitsanalyse genauer untersucht, nämlich das Lernen Bayesscher Netze aus Daten. Die globale Struktur solcher Netze stellt die Abhängigkeiten zwischen verschiedenen Variablen dar, die zur Beschreibung des betrachteten Weltausschnitts benötigt werden. Neuere Entwicklungen zur Verbesserung dieses Verfahrens nutzen seine Verwandschaft mit dem Lernen von Entscheidungsbäumen bzw. Entscheidungsgraphen, um eventuell in den bedingten Wahrscheinlichkeitsverteilungen vorhandene Regelmäßigkeiten kompakt darstellen zu können. Dadurch reduziert sich die Zahl der zu speichernden Wahrscheinlichkeiten u.U. erheblich.

Literatur

[1] C. Borgelt and R. Kruse. Evaluation Measures for Learning Probabilistic and Possibilistic Networks. *Proc. 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97)*, Vol. 2:pp. 1034–1038, Barcelona, Spain, 1997

[2] C. Borgelt und R. Kruse. Attributauswahlmaße für die Induktion von Entscheidungsbäumen: Ein Überblick. In: G. Nakhaeizadeh, ed. *Data Mining: Theoretische Aspekte und Anwendungen* pp. 77-98, Physica-Verlag, Heidelberg, 1998

[3] C. Borgelt, R. Kruse und G. Lindner. Lernen probabilistischer und possibilistischer Netze aus Daten. *Künstliche Intelligenz*, Themenheft Data Mining, 1:11–17, 1998

[4] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context Specific Independence in Bayesian Networks. *Proc. 12th Conf. on Uncertainty in Artificial Intelligence (UAI'96)*, Portland, OR, 1996

[5] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*, Wadsworth International, Belmont, CA, 1984

[6] D.M. Chickering, D. Heckerman, and C. Meek. A Bayesian Approach to Learning Bayesian Networks with Local Structure. *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*

(UAI'97), pp. 80–89, Morgan Kaufman, San Franscisco, CA, 1997

[7] G.F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9:309–347, Kluwer, 1992

[8] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / MIT Press, Cambridge, MA, 1996

[9] J. Gebhardt and R. Kruse. POSSINFER — A Software Tool for Possibilistic Inference. In: D. Dubois, H. Prade, and R. Yager, eds. *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, Wiley, 1995

[10] D. Geiger and D. Heckerman. Advances in Probabilistic Reasoning. *Proc. 7th Conf. on Uncertainty in Artificial Intelligence (UAI'91)*, pp. 118–126, Morgan Kaufman, San Franscisco, CA, 1997

[11] D. Heckerman. *Probabilistic Similarity Networks*. MIT Press 1991

[12] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning* 20:197–243, Kluwer, 1995

[13] R. Kruse and D. Meyer. *Statistics with Vague Data*. Reidel, Dordrecht, 1987

[14] R. Kruse, J. Gebhardt und F. Klawonn. *Fuzzy-Systeme, 2. erweiterte Auflage*. Teubner, Stuttgart, 1995.

[15] S.L. Lauritzen and D.J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society, Series B*, 2(50):157–224, 1988

[16] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition)*. Morgan Kaufman, New York 1992

[17] D. Poole. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence*, 64(1):81-129, 1993

[18] J.R. Quinlan. Induction of Decision Trees. *Machine Learning* 1:81–106, 1986

[19] J.R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993

[20] J.E. Smith, S. Holtzman, and J.E. Matheson. Structuring Conditional Relationships in Influence Diagrams. *Operations Research*, 41(2):280–297, 1993