Data Mining with Graphical Models

Christian Borgelt

Otto-von-Guericke-Universität Magdeburg
Institut für Wissens- und Sprachverarbeitung
http://fuzzy.cs.uni-magdeburg.de/~borgelt/

Data Mining, oder auch *Wissensentdeckung in Datenbanken*, ist ein noch recht junges Forschungsgebiet, das als Antwort auf die Datenflut entstanden ist, der wir uns heute gegenübersehen. Es widmet sich der Herausforderung, Techniken zu entwickeln, die Menschen helfen können, nützliche Muster in ihren Daten zu finden. Eine dieser Techniken — und sicher eine der wichtigsten, da sie für so häufige Data-Mining-Aufgaben wie die Konstruktion von Klassifikatoren und die Abhängigkeitsanalyse eingesetzt werden kann — ist das Lernen von *graphischen Modellen* aus Datensätzen von Beispielfällen. In meiner Dissertation stelle ich die Idee der graphischen Modelle dar, wobei ich besonders auf die noch weniger bekannten possibilistischen Netze eingehe, für die ich eine bessere Semantik zu liefern versuche. Weiter untersuche ich die Prinzipien des Lernens graphischer Modelle aus Daten und bespreche verschiedene Algorithmen, die für diese Aufgabe vorgeschlagen wurden. Die wesentlichen Leistungen dieser Arbeit bestehen in Verbesserungen und Erweiterungen dieser Algorithmen: Ich schlage eine Projektionsmethode für datenbankinduzierte Possibilitätsverteilungen, einen naiv-Bayes-artigen possibilistischen Klassifikator und mehrere neue Bewertungsmaße und Suchmethoden vor.

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. Diese Systeme entstanden aus der Notwendigkeit, bestimmte Informationen, wie z.B. die Adresse eines Kunden, schnell finden zu können. Doch heute, mit immer leistungsfähigeren Rechnern und durch Fortschritte in der Datenbank- und Softwaretechnologie, kann man daran denken, solche Datensammlungen nicht mehr einfach nur zum Abrufen bestimmter, gerade benötigter Informationen zu nutzen, sondern auch, um nach in diesen Datenhalden verstecktem Wissen zu suchen. Findet man z.B. in einem Supermarkt durch die Analyse der (mit Scannerkassen leicht erfaßbaren) Kassenbondaten heraus, daß bestimmte Produkte oft zusammen gekauft werden, so kann der Umsatz u.U. durch eine entsprechende Anordnung dieser Produkte in den Regalen gesteigert werden.

Doch um solches Wissen aus Datenbanken zu gewinnen, reichen die Abfragemöglichkeiten normaler Datenbanksysteme und die Methoden der klassischen Datenanalyse oft nicht mehr aus. Mit ihnen lassen sich zwar beliebige Einzelinformationen leicht abrufen, auch kann man einfache Aggregationen berechnen lassen oder die Hypothese testen, ob der Wochentag Einfluß auf die Qualität der Produktion hat, doch allgemeinere Muster, Strukturen, Regelmäßigkeiten bleiben 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 z.B. verwendet werden, um Entwicklungen zu prognostizieren oder Entscheidungen zu begründen.

In meiner Dissertation beschäftige ich mich mit zwei der wichtigsten Data-Mining-Aufgaben, nämlich der Konstruktion von Klassifikatoren und der Abhängigkeitsanalyse. Unter den verschiedenen Methoden zur Lösung dieser Aufgaben konzentriere ich mich auf das Lernen graphischer Modelle aus Datensätzen von Beispielfällen, wobei ich sowohl relationale und probabilistische als auch possibilistische graphische Modelle behandle.

2 Graphische Modelle

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. Nun sind manchmal nur bestimmte Wertkombinationen möglich, z.B. weil bestimmte Sonderausstattungen nicht gleichzeitig gewählt werden können, oder bestimmte Wertkombinationen sind häufiger als andere, z.B. sind rote VW Golf häufiger als gelbe BMW Z1. Diese Möglichkeits- oder Häufigkeitsinformation kann als Verteilung über dem kartesischen Produkt der Attributwertebereiche dargestellt werden. D.h., jeder Attributwertkombination wird seine Möglichkeit oder Wahrscheinlichkeit zugeordnet.

Da oft sehr viele Attribute notwendig sind, um einen Weltausschnitt angemessen zu beschreiben, verbietet es sich jedoch (wegen der mit der Zahl der Attribute exponentiell wachsenden Zahl von Wertkombinationen), diese Verteilung direkt darzustellen, z.B. um Schlußfolgerungen zu ziehen. Eine Möglichkeit, mit diesem Problem umzugehen, sind graphische Modelle. Ihnen liegt die Idee zugrunde, daß man Unabhängigkeiten zwischen Attributen ausnutzen kann, um eine solche hochdimensionale Verteilung in eine Menge von (bedingten oder Rand-) Verteilungen auf niedrigdimensionalen Teilräumen zu zerlegen. Diese Zerlegung (und die Unabhängigkeiten, die sie ermöglichen) wird durch einen Graphen dargestellt: Jeder Knoten steht für ein Attribut. Kanten verbinden Knoten, die direkt voneinander abhängen. Außerdem zeigen die Kanten die Wege an, auf denen Evidenz weitergegeben werden muß, wenn Schlußfolgerungen gezogen werden sollen.

Da graphische Modelle zuerst in der Wahrscheinlichkeitstheorie und Statistik entwickelt wurden, stammen die bekanntesten Ansätze aus diesem Bereich, nämlich die Bayeschen Netze [Pea88] und die Markow-Netze [LS88]. Das zugrundeliegende Zerlegungsprinzip wurde jedoch verallgemeinert, wodurch sich die sogenannten bewertungsbasierten Netzen (valuation-based networks) ergaben [She92], und auf die Possibilitätstheorie übertragen [GK96]. Alle Ansätze führten übrigens zu effizienten Implementierungen, z.B. HUGIN [AOJJ89], PULCINELLA [SU91], PATHFINDER [Hec91], und POSSINFER [GK96].

2.1 Zerlegung

Der Begriff der *Zerlegung* ist wahrscheinlich am besten aus der Datenbanktheorie bekannt. In der Tat ist die Datenbanktheorie eng mit der Theorie der graphischen Modelle verwandt. Diese Verwandschaft beruht auf dem Begriff der *Verbundzerlegbarkeit* einer Relation, die in relationalen Datenbanksystemen ausgenutzt wird, um hochdimensionale Relationen mit weniger Redundanz und (natürlich) weniger Speicherbedarf ablegen zu können.
Das Prinzip der Verbundzerlegbarkeit ist, daß eine Relation oft aus bestimmten *Projektionen* rekonstruiert werden kann, indem man den *natürlichen Verbund* dieser Projektionen bildet. Formal kann man dies wie folgt beschreiben: Sei $U = \{A_1, \ldots, A_n\}$ eine Menge von Attributen mit den Wertebereichen $\mathrm{dom}(A_i)$. Weiter sei r_U eine Relation über U. Eine solche Relation kann z.B. durch ihre *Indikatorfunktion* dargestellt werden, die jedem in der Relation enthaltenen Tupel den Wert 1 und allen anderen den Wert 0 zuordnet. Die Tupel selbst werden als Konjunktionen $\bigwedge_{A_i \in U} A_i = a_i$ dargestellt, die für jedes Attribut einen Wert angeben. Die *Projektion* auf eine Teilmenge $M \subseteq U$ der Attribute kann dann definiert werden als die Relation

$$r_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big) = \max_{\substack{\forall A_j \in U-M: \\ a_i \in \mathrm{dom}(A_j)}} r_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big),$$

wobei die etwas ungewöhnliche Notation unter dem Maximum ausdrücken soll, daß das Maximum über alle Werte aller Attribute in $U - M$ zu bilden ist. Mit dieser Notation heißt eine Relation r_U *verbundzerlegbar* bzgl. einer Familie $\mathcal{M} = \{M_1, \ldots, M_m\}$ von Teilmengen von U genau dann, wenn

$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) : \\ &\quad r_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \min_{M \in \mathcal{M}} r_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big).\end{aligned}$$

Man beachte, daß das Minimum der Projektionen äquivalent zum natürlichen Verbund der Relationenalgebra ist, was die Bezeichnung „verbundzerlegbar“ rechtfertigt.
Dieses Zerlegungsschema läßt sich leicht auf den probabilistischen Fall übertragen: Wir müssen nur die Projektion und den natürlichen Verbund durch die entsprechenden probabilitischen Operationen ersetzen. Wir erhalten dann als Zerlegungsformel

$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) : \\ &\quad p_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \prod_{M \in \mathcal{M}} \phi_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big).\end{aligned}$$

Die Funktionen ϕ_M können aus den Randverteilungen auf den Attributmengen M berechnet werden, was zeigt, daß die Berechnung von Randverteilungen die Rolle der Projektion übernimmt. Diese Funktionen werden auch *Faktorpotentiale* genannt [CGH97]. Alternativ kann man eine Zerlegung unter Ausnutzung des (verallgemeinerten) Produktsatzes der Wahrscheinlichkeitsrechnung und unter Verwendung bedingter Verteilungen beschreiben.

Der possibilistische Fall ist noch näher am relationalen, da die Zerlegungsformel mit der relationalen identisch ist: Es treten lediglich Possibilitätsverteilungen π statt der Relationen r auf, d.h. Funktionen, die nicht wie Indikatorfunktionen auf die Werte 0 und 1 beschränkt sind, sondern beliebige Werte aus dem Intervall $[0, 1]$ annehmen können. Auf diese Weise wird eine „graduelle Möglichkeit“ modelliert, und possibilistische graphische Modelle können folglich als „Fuzzifizierungen“ relationaler Modelle gesehen werden.
Wenn man solche Möglichkeitsgrade einführt, stellt sich natürlich die Frage nach der Semantik der Zahlenwerte. In meiner Dissertation stütze ich mich zur Beantwortung dieser Frage i.w. auf das *Kontextmodell* [GK93]. Da die bekannten, auf diesem Modell beruhenden Begründungen der in der Possibilitätstheorie verwendeten Maximum- und Minimumbildungen [BMP95, Geb97] jedoch nicht schlüssig sind, gebe ich eine eigene Begründung.

2.2 Graphische Darstellung

Zerlegungen können bequem durch Graphen dargestellt werden. Erstens kann man durch Graphen die Attributmengen M der Zerlegung angeben. Wie dies geschieht, hängt davon ab, ob der Graph gerichtet oder ungerichtet ist. Wenn er ungerichtet ist, sind die Mengen M die maximalen Cliquen des Graphen, wobei eine Clique ein vollständiger Teilgraph ist, und eine Clique maximal genannt wird, wenn sie nicht Teil einer anderen Clique ist. Wenn der Graph gerichtet ist, kann man die Verteilungen der Zerlegung expliziter angeben: Man kann bedingte Verteilungen verwenden, da die Kantenrichtungen eine Unterscheidung von bedingten und bedingenden Attributen erlauben. Im relationalen und im possibilistischen Fall ergeben sich dadurch allerdings keine Unterschiede, da die bedingten Verteilungen mit den unbedingten identisch sind (da keine Renormalisierung vorgenommen wird).
Zweitens können Graphen benutzt werden, um (bedingte) Abhängigkeiten und Unabhängigkeiten mittels des Begriffs der *Trennung* von Knoten anzugeben. Was unter „Trennung“ zu verstehen ist, hängt wieder davon ab, ob der Graph gerichtet oder ungerichtet ist. Wenn er ungerichtet ist, wird Knotentrennung wie folgt definiert: Wenn X, Y und Z drei disjunkte Knotenmengen sind, dann trennt Z X von Y, wenn alle Pfade von einem Knoten in X zu einem Knoten in Y einen Knoten in Z enthalten.
Für gerichtete kreisfreie Graphen wird Knotentrennung so definiert [Pea88]: Wenn X, Y und Z drei disjunkte Knotenmengen sind, dann trennt Z X von Y, wenn es keinen Pfad (mit beliebigen Kantenrichtungen) von einem Knoten in X zu einem Knoten in Y gibt, auf dem die folgenden Bedingungen gelten:

1. Jeder Knoten, auf den zwei Kanten des Pfades gerichtet sind, ist entweder in Z oder hat einen Nachkommen in Z, und
2. jeder andere Knoten ist nicht in Z.

Mit Hilfe dieser Trennungskriterien werden *bedingte Unabhängigkeitsgraphen* definiert: Ein Graph ist ein bedingter Unabhängigkeitsgraph bzgl. einer (mehrdimensionalen) Verteilung, wenn er durch Knotentrennung nur in der Verteilung geltende bedingte Unabhängigkeiten darstellt. Bedingte Unabhängigkeit bedeutet (für drei Attribute A, B und C mit A

unabhängig von C gegeben B, die Verallgemeinerung ist offensichtlich), daß

$$P(A = a, B = b, C = c) = P(A = a \mid B = b) \cdot P(C = c \mid B = b)$$

im probabilistischen Fall und

$$\pi(A = a, B = b, C = c) = \min\{\pi(A = a \mid B = b), \pi(C = c \mid B = b)\}$$

im possibilistischen und relationalen Fall.
Diese Formeln deuten außerdem an, daß bedingte Unabhängigkeit und Zerlegbarkeit eng verbunden sind. Formal wird diese Verbindung durch Sätze hergestellt, die besagen, daß eine Verteilung genau dann zerlegbar bezüglich eines Graphen ist, wenn dieser ein bedingter Unabhängigkeitsgraph ist. Im probabilistischen Fall wird ein solcher Satz üblicherweise [HC71] zugeschrieben. Im possibilistischen Fall kann ein analoger Satz gezeigt werden, wenn auch bestimmte Einschränkungen nötig sind [Geb97].
Schließlich erweist sich der einem graphischen Modell zugrundeliegende Graph als sehr nützlich für die Ableitung von Algorithmen zur Propagation von Evidenz, da die Weitergabe von Information durch sich Nachrichten zuschickende Knotenprozessoren implementiert werden kann. Details findet man z.B. in [CGH97].

3 Lernen graphischer Modelle aus Daten

Da ein graphisches Modell die in einem gegebenen Weltausschnitt geltenden Abhängigkeiten und Unabhängigkeiten übersichtlich darstellt und effizientes Schlußfolgern erlaubt, ist es ein sehr mächtiges Werkzeug — sobald es erstellt ist. Sein Aufbau durch menschliche Experten kann jedoch aufwendig und langwierig sein. Daher hat sich die Forschung in den vergangenen Jahren stark mit dem Lernen graphischer Modelle aus einem Datensatz von Beispielfällen beschäftigt. Obwohl gezeigt werden konnte, daß diese Lernaufgabe im allgemeinen Fall NP-hart ist [CGH94], wurden einige sehr erfolgreiche heuristische Algorithmen entwickelt [CH92, HGC95, GK95].
Einige dieser Ansätze, speziell probabilistische, sind jedoch auf das Lernen aus *präzisen* Daten beschränkt. D.h., die Beschreibung der Beispielfälle darf weder fehlende Werte noch mengenwertige Information enthalten, sondern es muß je Beispielfall genau einen Wert für jedes Attribut geben. In Anwendungen ist diese Voraussetzung jedoch selten erfüllt: Datenbanken sind meist unvollständig und nützliche unpräzise Information (im Sinne einer Menge von Werten für ein Attribut) ist oft verfügbar (wenn sie auch häufig vernachlässigt wird, da herkömmliche Datenbanken sie meist nicht angemessen verarbeiten können). Wir sehen uns daher der Herausforderung gegenüber, die bestehenden Lernalgorithmen auf unvollständige und unpräzise Daten zu erweitern.
Im Bereich der probabilistischen graphischen Modelle wird versucht, diese Herausforderung mit dem Expectation-Maximization-Algorithmus [DLR77, BKS97] zu meistern. In meiner Dissertation wende ich mich dagegen den possibilistischen graphischen Modellen zu, da die Possibilitätstheorie [DP88] eine sehr bequeme Behandlung von fehlenden Werten und unpräzisen Informationen erlaubt.

3.1 Prinzipien des Lernens graphischer Modelle aus Daten

Um graphische Modelle aus Daten zu lernen, gibt es i.w. drei Ansätze:

- Direkter Test einer Verteilung auf Zerlegbarkeit bzgl. eines gegebenen Graphen.
- Konstruktion eines bedingten Unabhängigkeitsgraphen über Unabhängigkeitstests.
- Auswahl von Kanten über das Messen der Stärke der Abhängigkeit von Attributen.

Jedoch ist keiner dieser Ansätze perfekt. Der erste Ansatz scheitert daran, daß man wegen der mit der Zahl der Attribute überexponentiell steigenden Zahl von Graphen nicht alle Graphen durchmustern kann. Der zweite Ansatz geht meist von der starken Annahme der perfekten Darstellbarkeit der bedingten Unabhängigkeiten aus und benötigt Unabhängigkeitstests hoher Ordnung, die nur bei sehr großen Datensätzen einigermaßen verläßlich durchgeführt werden können. Für den dritten Ansatz können leicht Beispiele gefunden werden, bei denen er einen suboptimalen Graphen liefert (siehe meine Dissertation). Der zweite und der dritte Ansatz führen jedoch, ggf. unter zusätzlichen Annahmen, zu guten heuristischen Algorithmen, die gewöhnlich aus zwei Teilen bestehen:

1. einem *Bewertungsmaß* (um die Qualität eines Modells einzuschätzen) und
2. einer *Suchmethode* (um den Raum der möglichen Modelle zu durchmustern).

Allerdings wird nicht immer direkt im Raum der möglichen Graphen gesucht. Es kann z.B. nach bedingten Unabhängigkeiten gesucht werden oder nach den besten Elternknoten zu einem Attribut. Die Charakterisierung des Algorithmus durch ein Bewertungsmaß eine Suchmethode ist jedoch auch diesen Fällen passend.

3.2 Berechnung von Projektionen

Neben den beiden im vorangehenden Abschnitt genannten Bestandteilen eines Lernalgorithmus für graphische Modelle benötigt man noch eine Operation für eine technische Aufgabe, nämlich die Schätzung der bedingten oder Randverteilungen aus einem Datensatz von Beispielfällen. Diese Operation wird oft vernachlässigt, da sie im relationalen und im probabilistischen Fall trivial ist, jedenfalls bei präzisen Daten. Im ersteren ist sie eine Operation der Relationenalgebra (eben die Projektion, weswegen ich hier allgemein von der Berechnung von Projektionen spreche), im letzteren besteht sie im einfachen Auszählen bestimmter relativer Häufigkeiten. Nur wenn unpräzise Informationen vorhanden sind, ist diese Operation komplizierter. Man greift in diesem Fall (siehe oben) auf den Expectation-Maximization-Algorithmus [DLR77, BKS97] zurück, der recht aufwendig ist.
Einfacher ist die Behandlung unpräziser Information in der Possibilitätstheorie, speziell wenn sie auf dem Kontextmodell aufgebaut wird. Man kann dann jeden Fall des Beispieldatensatzes als einen Kontext auffassen, und die Impräzision bequem innerhalb des Kontextes behandeln. Allerdings gibt es auch im possibilistischen Fall Schwierigkeiten bei der Bestimmung der Projektionen, wie ich in meiner Dissertation an einem einfachen Beispiel zeige: Es gibt keine naive Operation (wie einfaches Auszählen), mit der sich die Randverteilungen direkt aus dem Beispieldatensatz bestimmen ließen. Es gelingt mir

in meiner Dissertation jedoch, eine Vorverarbeitungsmethode zu entwickeln, die den *Abschluß* des Beispieldatensatzes *unter Tupelschnitt* berechnet. Aus diesem Abschluß lassen sich die Randverteilungen durch eine einfache Maximumbildung (analog zum Auszählen bedingter Häufigkeiten) sehr effizient bestimmen.

3.3 Bewertungsmaße

Ein *Bewertungsmaß* dient dazu, die Güte eines gegebenen Kandidatenmodells bzgl. eines gegebenen Datensatzes von Beispielfällen einzuschätzen, so daß man herausfinden kann, welches Modell am besten auf die Daten paßt. Eine wünschenswerte Eigenschaft eines Bewertungsmaßes ist Zerlegbarkeit, d.h., die Bewertung der Güte des gesamten Modells sollte sich aus lokalen Bewertungen, z.B. Bewertungen von Cliquen oder gar einzelnen Kanten, zusammensetzen lassen. Die meisten solcher Bewertungsmaße messen die Stärke der Abhängigkeit von Attributen, da dies sowohl für den zweiten als auch für den dritten Ansatz zum Lernen graphischer Modelle aus Daten (siehe Abschnitt 3.1) notwendig ist, entweder, um einzuschätzen, ob eine bedingte Unabhängigkeit vorliegt, oder um die stärksten Abhängigkeiten zwischen Attributen zu finden.
Im probabilistischen Fall gibt es eine Vielzahl von Bewertungsmaßen, die auf sehr verschiedenen Ideen beruhen und für sehr unterschiedliche Zwecke entwickelt wurden. Insbesondere lassen sich alle Maße, die zum Lernen von Entscheidungsbäumen entwickelt wurden, auf das Lernen graphischer Modelle übertragen, wenn diese Möglichkeit auch nur selten erkannt und entsprechend selten genutzt wird. In meiner Dissertation habe ich eine Reihe von Maßen zusammengetragen (z.B. Informationsgewinn(verhältnis), Gini-Index, Relieff-Maß, K2-Metrik, minimale Beschreibungslänge etc.). Ich erläutere die ihnen zugrundeliegenden Ideen und wie sie zum Lernen von graphischen Modellen eingesetzt werden können. Außerdem entwickle ich eine Erweiterung der K2-Metrik [CH92, HGC95] und eines Maße, das auf dem Prinzip der minimalen Beschreibungslänge beruht [Ris83]. In diesen Erweiterungen führe ich einen „Sensitivitätsparameter“ ein, mit dem sich die Tendenz, weitere Kanten in das graphische Modell einzusetzen und es so komplexer zu machen, steuern läßt. Ein solcher Parameter hat sich für Anwendungen als wichtig erwiesen (siehe die in Abschnitt 4 kurz beschriebene Anwendung bei DaimlerChrysler).
Bewertungsmaße für possibilistische graphische Modelle lassen sich auf zwei Weisen ableiten: Erstens kann man die enge Verwandtschaft zu relationalen Netzen ausnutzen, indem man sich auf den aus der Fuzzy-Mengentheorie [KGK94] bekannten Begriff des α-*Schnittes* stützt. Possibilitätsverteilungen lassen sich so als *Menge von Relationen* deuten, mit einer Relation je Möglichkeitsgrad α. Man sieht dann leicht, daß eine Possibilitätsverteilung genau dann zerlegbar ist, wenn jeder ihrer α-Schnitte zerlegbar ist. Folglich lassen sich Bewertungsmaße für possibilistische graphische Modelle aus entsprechenden Maßen für relationale graphische Modelle ableiten, indem man ihren Wert über alle möglichen Werte α integriert. Auf diese Weise läßt sich z.B. der Spezifizitätsgewinn [Geb97] aus dem Hartley-Informationsgewinn [Har28] herleiten, von dem ich in meiner Dissertation einige Varianten entwickle (z.B. durch Normalisierung).

Eine zweite Möglichkeit, Bewertungsmaße für den possibilistischen Fall zu erhalten, besteht in der Analogiebildung zum probabilistischen Fall. So leite ich in meiner Dissertation u.a. ein possibilistisches Analogon der wechselseitigen Shannon-Information her.

3.4 Suchmethoden

Die verwendete *Suchmethode* bestimmt, welche Graphen betrachtet werden. Da sich eine erschöpfende Suche wegen der großen Zahl möglicher Graphen verbietet, werden heuristische Verfahren eingesetzt. Diese Verfahren schränken gewöhnlich die Menge der betrachteten Graphen stark ein und nutzen den Wert des Bewertungsmaßes, um die Suche zu steuern. Außerdem verhalten sie sich oft „gierig“ (greedy) bzgl. der Modellgüte.
Die einfachste Suchmethode ist die Konstruktion eines optimalen spannenden Baumes für gegebene Kantengewichte. Dieses Verfahren wurde zuerst in [CL68] genutzt, wobei der *Shannon-Informationsgewinn* die Kantengewichte lieferte. Im possibilistischen Fall kann man statt des Informationsgewinns den oben erwähnten Spezifizitätsgewinn einsetzen, um einen analogen Algorithmus zu erhalten [Geb97], aber auch fast alle anderen Maße (probabilistische wie possibilistische) lassen sich verwenden.
Eine naheliegende Erweiterung ist die „gierige“ Suche nach Elternknoten in gerichteten Graphen, die oft auf einer vorzugebenden topologischen Ordnung der Attribute aufsetzt: Zu Beginn wird das Bewertungsmaß für einen elternlosen Knoten berechnet. Dann werden schrittweise Eltern hinzugefügt, wobei stets das Attribut gewählt wird, das den höchsten Wert des Bewertungsmaßes liefert. Die Suche endet, wenn keine weiteren Elternkandidaten mehr zur Verfügung stehen oder sich der Wert des Maßes nicht mehr verbessert. Diese Suchmethode wurde im K2-Algorithmus [CH92] eingesetzt, mit der bereits erwähnten *K2-Metrik* als Bewertungsmaß. Auch diese Suchmethode läßt sich leicht auf den possibilistischen Fall übertragen, indem man das Bewertungsmaß austauscht.
In meiner Dissertation entwickle ich außerdem zwei neue Suchmethoden. Erstere geht von einem optimalen spannenden Baum (siehe oben) aus und fügt Kanten hinzu, wenn durch den Baum dargestellte bedingte Unabhängigkeiten nicht gelten. Welche Kanten hinzugefügt werden dürfen, unterliegt aber bestimmten, leicht zu überprüfenden Einschränkungen, die sicherstellen, daß die Cliquen des Graphen höchstens drei Knoten enthalten. Dadurch wird außerdem sichergestellt, daß der sich ergebende Graph Hyperbaumstruktur hat. (Ein Hyperbaum ist ein kreisfreier Hypergraph, und in einem Hypergraphen ist die Einschränkung, daß Kanten nur zwei Knoten verbinden dürfen, aufgehoben: eine Hyperkante kann beliebig viele Knoten miteinander verbinden.) Das zweite von mir vorgeschlagene Suchverfahren verwendet die Methode des simulierten Ausglühens, um direkt einen Hyperbaum zu lernen. Für diesen Algorithmus mußte ich vor allem eine Methode zum zufälligen Erzeugen und Verändern eines Hyperbaumes entwickeln.
Diese beiden Suchmethoden sind sehr nützlich, da man mit ihnen die Komplexität von späteren Schlußfolgerungen in dem gelernten graphischen Modell bereits zur Lernzeit kontrollieren kann, denn diese Komplexität hängt entscheidend von der Größe der Hyperkanten des gelernten Hyperbaumes ab.

4 Anwendung

In einer Zusammenarbeit zwischen der Universität Magdeburg und der DaimlerChrysler AG hatte ich Gelegenheit, Algorithmen zum Lernen graphischer Modelle auf reale Fahrzeugdaten anzuwenden. Ziel der Analyse war die Aufdeckung möglicher Ursachen für aufgetretene Fehler/Schäden. Der Ansatz ist zwar sehr einfach (es wurde ein zweischichtiger Graph gelernt: eine Schicht beschreibt Bauzustandsmerkmale des Fahrzeugs, die andere Fehler/Schäden), war aber sehr erfolgreich. In Vergleichen mit menschlichem Expertenwissen konnten mit einem von mir entwickelten Prototyp leicht und in kurzer Zeit Hinweise auf Fehlerursachen gefunden werden, nach denen menschliche Experten zuvor wochenlang gesucht hatten. Für den Erfolg erwiesen sich u.a. die von mir in zwei Bewertungsmaße eingeführten Sensitivitätsparameter (siehe Abschnitt 3.3) als wichtig.

Literaturverzeichnis

[AOJJ89] Andersen, S.; Olesen, K.; Jensen, F.; Jensen, F.: HUGIN — A Shell for Building Bayesian Belief Universes for Expert Systems. In Proc. 11th Int. J. Conf. on Artificial Intelligence. Detroit, MI, USA, 1989, S. 1080–1085.

[BKS97] Bauer, E.; Koller, D.; Singer, Y.: Update Rules for Parameter Estimation in Bayesian Networks. In Proc. 13th Conf. on Uncertainty in Artificial Intelligence. Providence, RI, USA, 1997, S. 3–13.

[BMP95] Baldwin, J.; Martin, T.; Pilsworth, B.: FRIL — Fuzzy and Evidential Reasoning in Artificial Intelligence. Research Studies Press/J. Wiley & Sons, Taunton/Chichester, United Kingdom, 1995.

[CGH94] Chickering, D.; Geiger, D.; Heckerman, D.: Learning Bayesian Networks is NP-Hard (Technical Report MSR-TR-94-17). Techn. Ber., Microsoft Research, Advanced Technology Division, Redmond, WA, USA, 1994.

[CGH97] Castillo, E.; Gutierrez, J.; Hadi, A.: Expert Systems and Probabilistic Network Models. Springer, New York, NY, USA, 1997.

[CH92] Cooper, G.; Herskovits, E.: A Bayesian Method for the Induction of Probabilistic Networks from Data. In Machine Learning, Bd. 9:(1992), S. 309–347.

[CL68] Chow, C.; Liu, C.: Approximating Discrete Probability Distributions with Dependence Trees. In IEEE Trans. on Information Theory, Bd. 14 (3):(1968), S. 462–467.

[DLR77] Dempster, A.; Laird, N.; Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. In Journal of the Royal Statistical Society (Series B), Bd. 39:(1977), S. 1–38.

[DP88] Dubois, D.; Prade, H.: Possibility Theory. Plenum Press, New York, NY, USA, 1988.

[Geb97] Gebhardt, J.: Learning from Data: Possibilistic Graphical Models, Habilitationschrift. University of Braunschweig, 1997.

[GK93] Gebhardt, J.; Kruse, R.: The Context Model — An Integrating View of Vagueness and Uncertainty. In Int. Journal of Approximate Reasoning, Bd. 9:(1993), S. 283–314.

[GK95] Gebhardt, J.; Kruse, R.: Learning Possibilistic Networks from Data. In Proc. 5th Int. Workshop on Artificial Intelligence and Statistics. Fort Lauderdale, FL, USA, 1995, S. 233–244.

[GK96] Gebhardt, J.; Kruse, R.: POSSINFER — A Software Tool for Possibilistic Inference. In Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications (Dubois, D.; Prade, H.; Yager, R., Hg.), J. Wiley & Sons, New York, NY, USA, 1996, S. 407–418.

[Har28] Hartley, R.: Transmission of Information. In The Bell Systems Technical Journal, Bd. 7:(1928), S. 535–563.

[HC71] Hammersley, J.; Clifford, P.: Markov Fields on Finite Graphs and Lattices, 1971. Unpublished manuscript, cited in [Ish81].

[Hec91] Heckerman, D.: Probabilistic Similarity Networks. MIT Press, Cambridge, MA, USA, 1991.

[HGC95] Heckerman, D.; Geiger, D.; Chickering, D.: Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. In Machine Learning, Bd. 20:(1995), S. 197–243.

[Ish81] Isham, V.: An Introduction to Spatial Point Processes and Markov Random Fields. In Int. Statistical Review, Bd. 49:(1981), S. 21–43.

[KGK94] Kruse, R.; Gebhardt, J.; Klawonn, F.: Foundations of Fuzzy Systems. J. Wiley & Sons, Chichester, United Kingdom, 1994.

[LS88] Lauritzen, S.; Spiegelhalter, D.: Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. In Journal of the Royal Statistical Society, Series B, Bd. 50 (2):(1988), S. 157–224.

[Pea88] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA, USA, 1988.

[Ris83] Rissanen, J.: A Universal Prior for Integers and Estimation by Minimum Description Length. In Annals of Statistics, Bd. 11:(1983), S. 416–431.

[She92] Shenoy, P.: Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems. In Fuzzy Logic for the Management of Uncertainty (Zadeh, L.; Kacprzyk, J., Hg.), J. Wiley & Sons, New York, NY, USA, 1992, S. 83–104.

[SU91] Saffiotti, A.; Umkehrer, E.: PULCINELLA: A General Tool for Propagating Uncertainty in Valuation Networks. In Proc. 7th Conf. on Uncertainty in Artificial Intelligence. Los Angeles, CA, USA, 1991, S. 323–331.

Christian Borgelt, geboren am 6. Mai 1967 in Bünde (Westfalen), Besuch der Grundschule Südlengern und des Gymnasiums am Markt Bünde, Abitur 1986. Wehrdienst in Munster 1986–1987. Studium der Informatik und Physik an der Universität Carolo Wilhelmina zu Braunschweig ab 1987, Vordiplom in Informatik 1989, Vordiplom in Physik 1992, Diplom in Informatik 1995. 1995 zunächst angestellt bei der Firma Lineas Informationstechnik GmbH, Braunschweig, anschließend tätig im Forschungszentrum der Daimler-Benz AG, Ulm (Arbeitsgruppe Maschinelles Lernen und Data Mining, Prof. G. Nakhaeizadeh). Seit 1996 wissenschaftlicher Mitarbeiter an der Otto-von-Guericke-Universität Magdeburg, Institut für Wissens- und Sprachverarbeitung (Arbeitsgruppe neuronale Netze und Fuzzy-Systeme, Prof. R. Kruse). 2000 Promotion zum Dr.-Ing. mit der hier beschriebenen Arbeit.