KDD-Modelle für die Informationsfusion

Ingolf Geist
Institut für Technische und Betriebliche Informationssysteme
Otto-von-Guericke-Universität Magdeburg
Postfach 4120, D-39016 Magdeburg
geist@iti.cs.uni-magdeburg.de

Zusammenfassung

Die Informationsfusion verbindet Methoden aus verschiedenen Gebieten der Informatik, um eine Integration und Interpretation von Daten aus verteilten, heterogenen Quellen zu ermöglichen. Dieser interaktive und iterative Prozess soll im Ergebnis zu Informationen höherer Qualität führen. Ziel aus Datenbanksicht ist es, ein effizientes Framework für die Informationsfusion zu schaffen. Ein erster Schritt hierzu ist ihre formale Beschreibung. Durch die Verwandtschaft zu KDD-Prozessen werden in dieser Arbeit verschiedene Frameworks für die Beschreibung von KDD-Prozessen auf die Möglichkeit der Nutzung in der Informationsfusion untersucht. Dabei erfolgt eine Beschränkung auf eine anfrageorientierte Sichtweise auf den Prozess, da diese den interaktiven und iterative Charakter der Fusion sehr gut widerspiegeln kann.

1 Einleitung

Die Informationsfusion kann als ein Prozess der Interpretation von Daten aus verschiedenen, heterogenen Quellen gesehen werden. Dieser Prozess besteht aus mehreren Schritten – der Integration und Aufbereitung, Vorverarbeitung, Analyse und Nachverarbeitung sowie Darstellung der Daten. Hierbei sind die Schritte nicht unabhängig voneinander, sondern Erkenntnisse aus späteren Schritten können Änderungen an vorhergehenden erfordern.

Aus dieser Darstellung der Informationsfusion ist zu ersehen, dass sie ein interaktiver und iterativer sowie heterogener Prozess ist, in dem der Anwender eine wichtige Rolle spielt und durch ein effizientes System unterstützt werden muss.

Ziel ist es, die verschiedenen Methoden der Informationsfusion zusammen in einer effizienten Umgebung ausführen zu können. Dieses erfordert neben dem effizienten Zugriff auf verteilte, heterogene Quellen eine formale Spezifikation der Informationsfusion, um einerseits die verschiedenen Schritte und Daten zu integrieren und andererseits weitere Optimierungspotenziale ausnutzen zu können.

Der Rest der Arbeit ist folgendermaßen aufgeteilt: Im Abschnitt 2 werden die Anforderungen an eine Spezifikation für die Informationsfusion dargelegt. Anschließend wird im Abschnitt 3 verschiedene Beschreibungsmöglichkeiten des KDD(Knowledge Discovery in Databases)-Prozesses beschrieben, da ein Teil der Informationsfusion dem KDD-Prozess entspricht. Der Abschnitt 4 gibt eine Zusammenfassung und einen Ausblick auf weitere Arbeiten.

2 Anforderungen

Die Informationsfusion ist ein heterogener Prozess, das heißt unterschiedliche Datenbearbeitungs- und Datenanalysemethoden müssen in einer Spezifikation vereinigt werden. Weiterhin können Datenquellen unterschiedlicher Art – z.B. Datenbanksysteme, Webseiten oder auch Textdateien – die Ausgangsdaten liefern. Die Methoden zur Informationsfusion sind neben einfachen

Datenbankoperationen (z.B. Join, Union, Aggregation usw.) auch erweiterte Operationen wie Methoden des maschinellen Lernens oder Berechnungen von Statistiken sowie Integrationsoperatoren für die Aufhebung von Integrationskonflikte. Diese einzelnen Datenquellen und Operationen müssen durch einen Anwender in einfacher Weise zusammengestellt bzw. ausgeführt und geändert werden können.

Aus diesen Überlegungen ergeben sich verschiedene Punkte, die eine Spezifikation eines solchen Prozesses abdecken muss:

- *Beschreibung der Datenquellen*: Die Daten und ihre Systeme sind in einer Weise zu beschreiben, dass eine schnelle und flexible Integration ermöglicht wird. Dieses gewährleistet ebenfalls die Auswahl der relevanten Daten für den Analyseprozess.

- *Beschreibung der DM-Modelle*: Durch die Anwendung von Data Mining (DM) Methoden entstehen DM-Modelle, in denen die Ergebnisse definiert sind. Somit wird eine Beschreibung der Daten und der Metadaten dieser Modelle benötigt.

- *Beschreibung der Operationen*: Diese Beschreibung soll nur die Art der Operationen darstellen und von der eigentlichen Implementation abstrahieren. Durch die Parametrisierbarkeit der Operationen, wird es dem Anwender erlaubt, die Ergebnismenge seinen Anforderungen entsprechend zu konfigurieren. Weiterhin ist eine Ad-hoc-Anwendung der Operationen zu gewährleisten.

- *Abgeschlossenheit der Spezifikation*: Ähnlich zu Datenbanksprachen sollen die Ergebnisse von Operationen wieder als Eingabe für weitere Operationen benutzt werden können, wodurch der iterative Charakter der Fusion unterstützt wird. Das heißt aber auch, dass DM-Modelle und Daten ähnlich zu behandeln sind.

- *Erweiterbarkeit*: Die Spezifikation muss so gestaltet sein, dass neue Methoden und Quellen darauf abgebildet werden können und nahtlos in das Framework einpassbar sind.

- *Effiziente Implementierbarkeit*: Durch die Beschreibung soll eine effiziente Umsetzung der Informationsfusion ermöglicht werden und neue Optimierungspotenziale für den gesamten Prozess erkennbar werden.

Aus den Eigenschaften und Anforderungen der Informationsfusion ergeben sich Ähnlichkeiten zu den Anforderungen an eine Beschreibung des KDD-Prozesses [HK01] und somit liegt es nahe, zu untersuchen, wie unterschiedliche Ansätze in diesem Bereich auf die Informationsfusion angewendet werden können.

3 KDD Beschreibungen

Es existieren in der Literatur verschiedene Ansätze, die versuchen, den gesamten KDD-Prozess formal zu beschreiben. Dabei werden ähnliche Anforderungen wie im Abschnitt zuvor gestellt. Diese Beschreibungen können hierbei in *Data Mining Anfragesprachen*, *formale Modelle* und *standardisierte Schnittstellenspezifikationen* unterschieden werden, die allerdings untereinander benutzt werden können.

3.1 Data Mining Anfragesprachen

In bisherigen Arbeiten wurden verschiedene DM-Anfragesprachen (z.B. DMQL [HFK+96], MS-QL [IV99]) bzw. Erweiterungen von SQL um DM-Operatoren (z.B. der `Mine Rule` Operator [MPC96]) entwickelt. Hierbei wird auf eine enge Integration mit relationalen Datenbanken geachtet, was z.B. aus der Verwandschaft mit SQL hervorgeht. Im Folgenden sollen die Sprachen DMQL und MSQL kurz mit ihren Hauptkonzepten vorgestellt werden.

3.1.1 DMQL – Data Mining Query Language

Eine DMQL-Anfrage kann in fünf Primitive unterteilt werden, die die verschiedenen Anforderungen einer Data Mining-Operation unterstützen. Zunächst kann der Anwender durch eine an SQL angelehnte Syntax die für ihn *relevanten Daten* auswählen. Das zweite Primitiv bestimmt die *Art des Wissens*, welches entdeckt werden soll – z.B. Klassifikation oder Assoziationsregeln. Die *Modellierung von Hintergrundwissen* stellt in DMQL das dritte Konzept dar. Dieses wird durch die explizite Angabe von Hierarchien erreicht. Durch die Angabe von *Thresholds* wird die Relevanz der Ergebnisse durch den Anwender festgelegt. DMQL unterstützt dafür die Parameter Signifikanz, Konfidenz und Redundanz. Das letzte Primitiv einer DMQL-Anfrage bestimmt, in welcher Form die Ergebnisse präsentiert werden.

Die Anfragen in DMQL führen jeweils eine DM-Operation aus. Weiterhin bieten sie eine Datenvor- und Datennachverarbeitung durch die Nutzung von Hierarchien, durch das Anzeigeprimitiv sowie durch die SQL-Syntax für die Datenaufbereitung. Allerdings ist es nicht möglich die Anfragen zu schachteln und somit das Ergebnis einer DM-Operation als Eingabe für weitere Operationen zu nutzen.

3.1.2 MSQL

Eine zweite DM-Anfragesprache ist MSQL, welche die Entdeckung und Verwendung von Assoziationregeln unterstützt. Die Ideen für diese Sprache sind aus der Tatsache entstanden, dass durch Data Mining eine sehr große Menge an Regeln entstehen kann, die effizient weiter bearbeitet werden muss. Die Ziele der Sprache sind die Einbindung von SQL, eine Unterstützung von Vor- und Nachbearbeitung von Daten, die Möglichkeit der iterativen Verfeinerung der Ergebnisse sowie die Verbindung von Regelgenerierung und -abfrage.

Ausgehend von Standard-SQL wurden die folgenden vier Erweiterungen vorgeschlagen. Das `GetRules`-Kommando erlaubt die Generierung von Regeln, die Parameter wie Support und Konfidenz erfüllen. Mit Hilfe von `SelectRule` können Regeln nach anwenderdefinierten Kriterien abgefragt werden. Tupel aus einer Relation, die bestimmte Regeln erfüllen, werden durch die Nutzung von `SATIFIES` bzw `VIOLATES` abgefragt. Der `Encoding`-Operator übernimmt die Vorverarbeitung der Daten, z.B. eine Diskretisierung von kontinuierlichen Wertebereichen.

Der gesamte KDD-Prozess kann durch eine Folge von Anfragen modelliert werden. Weiterhin können die Anfragen so geschachtelt werden, dass die Nachbearbeitung der entstandenden Regeln ermöglicht wird, ohne dass zuvor alle Regeln explizit materialisiert werden mussten.

3.2 Formale Modelle

In der letzten Zeit wurde sich in der Literatur mit einer formalen Beschreibung des KDD-Prozesses in seiner Gesamtheit, d.h. von Datenintegration, -aufbereitung, -analyse bis zur Nachbearbeitung, auseinander gesetzt. Zwei Ansätze sind beispielsweise die *Induktiven Datenbanken* und das *3W-Modell*.

3.2.1 Induktive Datenbanken

Angelehnt an den Begriff der deduktiven Datenbanken wurde in [BKM99] das Konzept der *Induktiven Datenbank* als ein Framework für KDD vorgestellt. Das Schema einer Induktiven Datenbank ist demnach $\mathcal{R} = (\mathbf{R}, (\mathcal{Q}_{\mathbf{R}}, e, \mathcal{V}))$, wobei $\mathbf{R}$ ein Datenbankschema, $\mathcal{Q}_{\mathbf{R}}$ eine Sammlung von Regeln, $\mathcal{V}$ eine Menge von Ergebniswerten und e eine Funktion ist, die jedes Paar $(\mathbf{r}, \theta_i)$ auf ein Wert in $\mathcal{V}$ abbildet. Dabei stellt $\mathbf{r}$ eine Datenbank für $\mathbf{R}$ dar und es gilt $\theta_i \in \mathcal{Q}_{\mathbf{R}}$. Eine Instanz von $\mathcal{R}$ ist als $(\mathbf{r}, s)$ definiert, wobei $s \subseteq \mathcal{Q}_{\mathbf{R}}$ gilt.

Der KDD-Prozess wird in diesem Ansatz durch eine Sequenz von Anfragen definiert. Hierbei werden Datenbankoperationen auf den Daten als auch auf den Regeln ebenso unterstützt wie die *Apply*-Operation. Diese letztgenannte Operation überbrückt die Grenze zwischen Daten und Regeln und ermittelt alle Tupel für die eine Regel gilt. Dieses Framework unterstützt die Nutzung von verschiedenen Regeln wie z.B. Assoziationsregeln und kann mit Hilfe der MSQL-Anfragesprache implementiert werden. Als mögliche Optimierungen kommen Ideen aus objekt-relationalen Datenbanken zum Einsatz.

3.2.2 Das 3W-Modell

Einen allgemeineren Ansatz wählt das *3W-Modell* [JLN00], welches verschiedene Data Mining- und Datenaufbereitungsverfahren unterstützt. Dieses geschieht durch die Einteilung des Prozesses in *drei Welten*: intensionale Dimensionen, extensionale Dimensionen und Rohdaten.

Die zentrale Idee dieses Ansatzes ist die Tatsache, dass alle Datenanalyse oder -aufbereitungsverfahren die Eigenschaft haben, den Datenraum in Gebiete, hier *Regionen* genannt, aufzuteilen. Die einzelnen Regionen können in einer Hierarchie angeordnet werden, welches das zweite zentrale Konzept des Modells widerspiegelt.

In der intensionalen Welt (I-World) sind die Regionen durch die Beschreibung ihrer Mitglieder, also durch Regeln, definiert. Formal ist eine Region eine Instanz eines Dimensionsschemas, welches aus einer Menge von *hierarchischen* und korrespondierenden *Constraint*-Attributen sowie weiteren Regionseigenschaften besteht. Mehrere in Beziehung stehender Regionen bilden eine Dimension. Operationen auf Regionen sind einmal durch die räumlichen Beziehungen zwischen der Regionen als auch durch die Einteilung in eine Hierarchie definiert. Diese Operationen bilden eine Dimensionenalgebra. Die extensionalen Dimensionen (E-World) sind durch die explizite Aufzählung ihre Mitglieder, also der Tupel, die sie erfüllen gekennzeichnet. Eine eine erweiterte relationale Algebra dient als Anfragesprache in der E-World. Die Rohdaten sind einfache Relationen.

Für die Modellierung eines kompletten Prozesses müssen die Grenzen zwischen den Welten überwunden werden. Diese Aufgabe wird von vier Operatoren übernommen. Die Operation $mine(\mu)$ stellt die eigentliche DM-Operation dar und erzeugt mit Hilfe eines DM-Algorithmus' eine Instanz eines Dimensionsschemas. Durch $populate(\alpha)$ werden Daten in ein intensionales Modell eingefügt, d.h. der Anwender sieht explizit welche Daten welchen Regionen angehören. Es entsteht eine extensionale Dimensionsinstanz. Der $lookup(\lambda)$-Operator liefert für eine extensionale Instanz eine intensionale Instanz zurück, indem er die hierarchischen Werte auf die einzelnen Constraints der intensionalen Beschreibung abbildet. *Refresh* ist ein Macro, welches eine Menge von intensionalen Instanzen, die auf der ein und derselben Datenmenge basieren und einen intensionalen Ausdruck bilden, durch die Nutzung des *populate*-Operators in einem Schritt aktualisiert.

Durch Nutzung der "Brücken-Operatoren" und der Operationen in den Welten kann der KDD-Prozess im 3W-Modell modelliert werden. Für die Informationsfusion müssen zusätzlich Integrationsoperationen definiert werden, um eine Ad-hoc-Integration der Daten zu unterstützen.

3.3 Standardisierte Spezifikationen

Als weitere Möglichkeiten zur Beschreibung von KDD-Prozessen sollen die Standards *PMML* [dmg01] und *OLE DB für Data Mining* [Mic00] behandelt werden. PMML ist eine XML-Sprache zur Definition von DM-Modellen zum Zwecke des Austausches zwischen verschiedenen Komponenten. Hierbei können die Verfahren sowie das DM-Schema und das Datenschema angegeben werden. OLE DB für Data Mining bietet gegenüber OLE DB und OLE DB für OLAP zusätzlich das virtuelle Objekt `Data Mining Model`, in dem die DM-Attribute als auch der DM-Algorithmus angegeben sind. Ein solches Modell kann durch eine `CREATE`-Anweisung oder aus einer PMML-Beschreibung erstellt werden, wobei es zunächst leer ist.

Durch das `INSERT`-Kommando erfolgt die Population des Modells aus einer Trainingsmenge, die ein beliebiger OLE DB Data Provider bereitstellt. Dieses trainierte Modell kann durch ein *Prediction Join* andere Datenmengen analysieren und die Ergebnisse kann der Anwender durch eine `SELECT`-Anweisung abfragen und anzeigen lassen. Somit ist die Integration von verschiedenen Methoden und Datenquellen möglich. Durch eine Reihe von Operationsanwendungen ist es möglich, einen KDD-Prozess zu definieren.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Anforderungskatalog für eine Beschreibung der Informationsfusion gegeben. Weiterhin wurden verschiedene Möglichkeiten der Beschreibung von KDD-Prozessen beschrieben, die eine Verwandschaft zu einem Teil der Informationsfusion haben. Dabei wurde sich hauptsächlich auf anfrageorientierte Konzepte konzentriert, da diese dem interaktiven und iterativen Charakter des Prozesses am besten gerecht werden.

Die beiden vorgestellten DM-Sprachen bieten dem Anwender eine einfache Möglichkeit des Aufrufs von DM-Operationen, sind aber nicht mächtig genug bezüglich der Abgeschlossenheit bzw. der Anzahl der unterstützten Methoden, da z.B. MSQL auf Assoziationregeln spezialisiert ist. Die Induktiven Datenbanken und das 3W-Modell beschreiben jeweils ein Framework für den KDD-Prozess auf unterschiedliche Art. Hierbei stellt das 3W-Modell den umfassenderen Ansatz dar, da die Induktiven Datenbanken auf Assoziationsregeln ausgerichtet sind. Als Datenbanktechnologien kommen für das 3W-Modell geometrische DBMS und für Induktive Datenbanken objekt-relationale DBMS in Frage. Die standardisierten Spezifikationen erlauben eine Integration von verschiedenen Data Mining Providern und Consumern über eine einheitliche Schnittstelle und erlauben so die Integration von heterogenen Methoden und Quellen. Weiterhin spezifiziert OLE DB für Data Mining ebenfalls eine Anfragesprache für Data Mining.

Aus diesen ersten Überlegungen soll eine Fusionsalgebra abgeleitet werden. Wobei das 3W-Modell ein Ansatz ist, der als Grundlage genommen werden könnte. Dazu sollen weitere Fusionsoperationen untersucht und Integrationoperationen eingebunden werden. Ziel ist es, eine Ad-hoc-Definition eines Prozesses dem Anwender zu ermöglichen. Ein zweiter Punkt ist die Entwicklung von Optimierungsstrategien über diese formale Spezifikation. Hierbei könnten beispielsweise Anfrageteile als eine Art materialisierte Sicht dienen, um bereits vorberechnete Mining-Ergebnisse für die Analyse zu benutzen.

Literatur

[BKM99] Jean-Francois Boulicaut, Mika Klemettinen, and Heikki Mannila. Modeling KDD Processes within the Inductive Database Framework. In *Data Warehousing and Knowledge Discovery, First International Conference DaWaK '99*, pages 293–302. Springer-Verlag, 1999.

[dmg01] PMML 1.1 – Predictive Model Markup Language. http://www.dmg.org/html/pmml_v1_1.html, March 2001.

[HFK+96] J. Han, Y. Fu, K. Koperski, W. Wang, and O. Zaiane. DMQL: A Data Mining Query Language for Relational Databases. In *SIGMOD'96 Workshop. on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada*, June 1996.

[HK01] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, 2001.

[IV99] Tomasz Imielinski and Aashu Virmani. MSQL: A Query Language for Database Mining. *Data Mining and Knowledge Discovery*, 3(4):373–408, December 1999.

[JLN00] Theodore Johnson, Laks V. S. Lakshmanan, and Raymond T. Ng. The 3W Model and Algebra for Unified Data Mining. In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases*, pages 21–32. Morgan Kaufmann, 2000.

[Mic00] Microsoft. *OLE DB for Data Mining*, July 2000.

[MPC96] Rosa Meo, Giuseppe Psaila, and Stefano Ceri. A New SQL-like Operator for Mining Association Rules. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pages 122–133. Morgan Kaufmann, 1996.