Datenanalyse-Plattform InformationMiner

Frank Rügheimer, Rudolf Kruse

Institut für Wissens- und Sprachverarbeitung, Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg
Tel.: (0391) 67 18182
Fax: (0391) 67 12018
E-Mail: {ruegheim, kruse}@iws.cs.uni-magdeburg.de

Zusammenfassung

Data-Mining-Verfahren haben in der Vergangenheit ihre Nützlichkeit bei der Bearbeitung verschiedenster Fragestellungen gezeigt. Umfangreiche Softwarepakete stellen heute Methoden und Datenaufbereitungsverfahren bereit und begleiten ihre Anwender bei vielen Schritten des Wissensentdeckungsprozesses [6]. Die für Spezialanwendungen erforderlichen maßgeschneiderten Analyselösungen können jedoch mit den bisher verfügbaren Werkzeugen oft nur unzureichend unterstützt werden und durch die Beschränkung auf ein vorher festgelegtes Methodenrepertoire stehen vielversprechende oder optimierte Verfahrenvarianten mitunter nicht zur Verfügung. Mit einer von unserer Arbeitsgruppe entwickelten Software versuchen wir diese Aspekte stärker zu berücksichtigen und eine erweiterbare Plattform zur effizienten Lösung von anwendungsspezifischen Data-Mining-Problemen bereitzustellen.

1 Motivation

Gefördert durch Weiterentwicklung von Techniken zur Erhebung, Speicherung und Verarbeitung von Daten sowie die zunehmende Verfügbarkeit geeigneter Analysemethoden, konnte Data-Mining in den letzten Jahren zur Bearbeitung einer Vielzahl unterschiedlichster Problemstellungen aus Wissenschaft und Wirtschaft eingesetzt werden. Aktuelle Werkzeuge unterstützen gewöhnlich mehrere Phasen des Data-Mining-Prozesses und stellen zahlreiche etablierte Verfahren bereit. Diese Bandbreite ermöglicht die Bearbeitung durchaus unterschiedlicher Problemklassen.

Als nachteilig erweist sich jedoch mitunter die beschränkte Erweiterbarkeit dieser Pakete. Gerade der Einsatz neuerer Analysemethoden kann hierdurch erschwert werden. So kann ein potentiell Erfolg versprechendes Verfahren in der Praxis unberücksichtigt bleiben, da innerhalb des vorhandenen Programmpakets keine entsprechende Implementierung bereitgestellt wird. Eventuell verfügare externe Implementierungen dagegen, sind gegebenfalls nur mit erheblichen Aufwand in den Data-Mining-Prozess einzubinden; dem Anwender ist es dann oftmals nicht möglich von den im Paket implementierten unterstützenden Tools, z.B. zur Vorverarbeitung oder Evaluierung, zu profitieren.

Einige der in Forschung und Wirtschaft auftretenden komplexen Fragestellungen lassen sich bei Beschränkung auf Standardverfahren allein nicht hinreichend detailliert behandeln. In vielen Fällen gelingt es jedoch unter Ausnutzung anwendungsspezifischen Hintergrundwissens, etwa mittels einer Komplexitätsreduktion anhand

bekannter Nebenbedingungen, Erfolge zu erzielen. Dagegen können geschickte Kombinationen unterschiedlicher Verfahren den Zugang zu Problemen erschließen, sich mit einer Methode allein nicht bearbeiten lassen. Gemeinsam mit der großen Variabilität in Anforderungen und Zielen bei Data-Mining-Problemen inspiriert dies ein System für die Entwicklung hochgradig flexibler und vom Nutzer spezifisch konfigurierbarer Lösungen. Ein geeigneter Ansatz findet sich im Konzept eines Software-Baukastens. Hierbei können vorgefertigte Module vom Nutzer zu einer Systemlösung zusammengestellt werden. Da enthaltenen Module zudem einzeln konfiguriert werden können, lässt sich das gewünschte Maß an Flexibilität realisieren. Zugleich ist jedoch zu berücksichtigen, dass die effiziente Nutzung eines derartigen Systems nach außen hin eine Reduktion des Konfigurationsbedarfes erfordert, so dass anwendbare Lösungen mit geringem Zeitaufwand erstellt werden können. Der scheinbare Gegensatz lässt sich durch Bereitstellung von Mechanismen zur weitestgehend automatischen Vorkonfiguration aufheben. Auch die Bereitstellung spezieller sachbereichsbezogener Verfahrensbibliotheken kann die Anwender entscheidend bei der zügigen Lösung von Data-Mining-Problemen unterstützen.

Schließlich sollten neue mit aktuellen Anwendungsfeldern einhergehende Anforderungen Berücksichtigung finden. Aufgaben aus dem Bereich der Biologie, z.B. die Analyse von Genexpressionsmustern zeichnen sich durch besonders hochdimensionale Datenräume aus. Kennzeichend für die Analyse multimedialer Daten ist die Notwendigkeit der Verabeitung stark heterogener Datenbestände. Für die Bearbeitung derartiger Probleme geeignete Algorithmen sollen daher in Zukunft ebenfalls in die Methodenbibliotheken eingehen.

2 Systemstruktur

Die in der Arbeitsgruppe entwickelte Software InformationMiner orientiert sich an einer graphischen Beschreibung der Verarbeitungsstroms beim Data-Mining. In dieser prozessorientierten Sicht werden Teilfunktionalitäten, wie das Einlesen oder Vorverarbeiten von Daten, die Visualisierung, die Verwendung von Analyseverfahren oder Anwendung der gewonnenen Modelle jeweils als Knoten in einem gerichteten azyklischen Graphen repräsentiert. Die Kanten in diesem Graphen weisen auf eine Übertragung von Information zwischen den beteiligten Knoten hin (Pipes&Filter-Konzept [8]). Die graphische Darstellung des Verarbeitungsstroms ist als wesentliches Element in die mit Java implementierte Nutzeroberfläche eingebunden, wo die Anwender durch Anpassung dieses Graphen die von ihnen gewünschten Abläufe spezifizieren. Abbildung 1 zeigt einen einfachen Verarbeitungsstrom zur Erzeugung eines Fuzzy-Clustermodells [2, 10].

2.1 Verfahrensbibliothek

Als Grundlage für die Erstellung von Verarbeitungsströmen dient ein umfangreiches Repertoire vorimplemetierter Methoden, die entsprechenden Knoten zugeordnet sind. Diese unterschiedlichen Knotentypen werden über Verfahrensbibliotheken bereitgestellt. Entsprechend ihrer Funktionalität können Knoten Kategorien zugeordnet werden. Neben modellerzeugenden Verfahren, wie der Induktion von

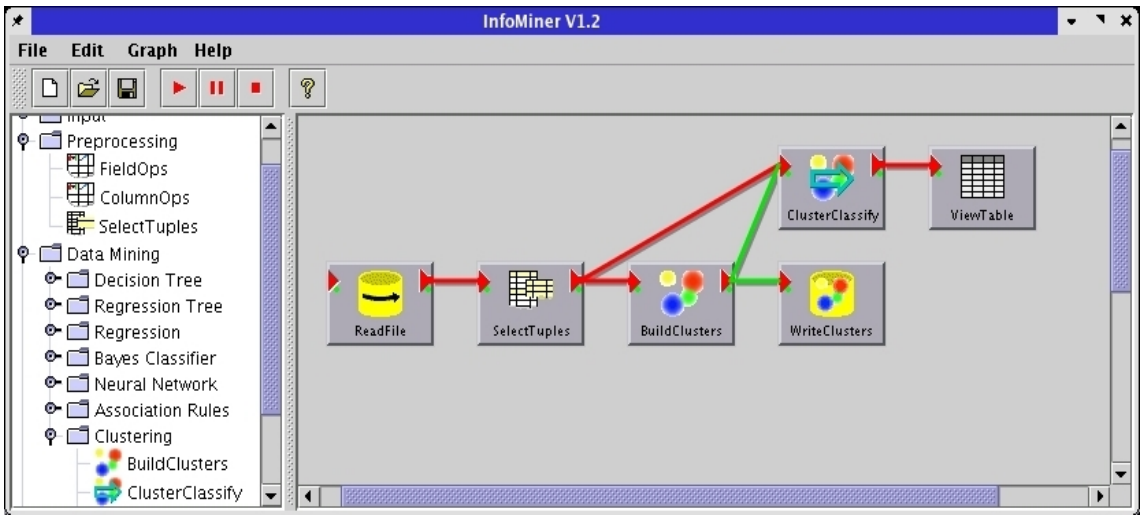

Abbildung 1: InformationMiner-Oberfläche einfachem Verarbeitungsstrom

Entscheidungs- und Regressionsbäumen [14, 4], Regellernern [1], Clusteringalgorithmen [2, 9, 10] und Neuronalen Netzen, werden Tools zur Unterstützung weiterer Abschnitte des Data-Mining-Prozesses bereitgestellt. Die Standardbibliothek unterscheidet hier Datenimport, Vorverarbeitung, Data-Mining, Visualisierung, Evaluierung oder Ausgabe. Durch weitere Unterteilung dieser Kategorien ergibt sich eine baumartige Struktur. Einigen Verfahren sind mehrere Knotenmodelle zugeordnet, da beispielsweise die Generierung eines Entscheidungsbaummodells von dessen Einsatz zur Datenklassifikation entkoppelt werden soll. Zur Erstellung eines Verarbeitungstromes werden die benötigten Knotenmodelle vom Verfahrensbaum auf die Arbeitsfläche gezogen, wo dann eine entsprechende Instanz erzeugt wird. Der neue Knoten lässt sich anschließend mit weiteren Knotenelementen verbinden und über nach außen geführte Parameter konfigurieren.

Auf technischer Seite werden die verfügbaren Knotenmodelle durch spezielle Java-Klassen definiert, die eine standardisierte Parameter- und Schnittstellenspezifikation und den Aufruf der dem Knoten zugeordeten Operation bereitstellen. Jeder Eintrag im Register der Verfahrensbibliothek verweist auf eine derartige Knotendefinitionsklasse und erklärt deren Positionierung(en) im Verfahrensbaum. Je nach geplantem Einsatzbereich kann die Verfahrensbibliothek von Entwicklerseite aus spezifisch angepasst und gegebenfalls um dort benötigte Methoden erweitert werden. Das Softwarepaket selbst kann also in individuellen Konfigurationen ausgeliefert werden und so ein dem jeweiligen Sachbereich angemessenes Verfahrenrepertoire bereitstellen (vertikales System). Dieser Ansatz gestattet später eine zügigere Lösungsentwicklung, da sich die Anwender lediglich mit den in ihrem Bereich sinnvoll einsetzbaren Methoden vertraut machen müssen.

2.2 Verbindungskomponente

Die Verbindungskomponente von InformationMiner ermöglicht es Knoten zu komplexen Verarbeitungsströmen zu verknüpfen. Die Kopplung von Knoten erfolgt über Konnektoren [8], mit denen jeweils die Menge der Ein- bzw. Ausgaben der von den Knoten dargestellten Prozesse repräsentiert wird. Durch Klicken und Ziehen

lassen sich die Ausgaben eines Knotens einem anderen Knoten als Eingabe zuordnen. Um die Zulässigkeit derart spezifizierter Verarbeitungsströme zu sichern, wird vor Ergänzung einer Kante ermittelt, ob hierdurch Zyklen im Graphen geschlossen würden und überprüft, dass mindestens eine der Eingaben des nachgeschalteten Knotens mit der Ausgabe des neuen Vorgängers kompatibel ist. Eine automatische Zuordnung von Knotenausgaben zu den vom Nachfolger benötigten Eingaben erleichtert gleichzeitig den Anwendern die Konfiguration der von ihnen erstellten Data-Mining-Lösungen (siehe auch 2.3).

Für die eigentliche Verarbeitungs werden die im Graphen beschiebenen Abhängigkeiten aufgelöst, um eine Ausführungssequenz zu generieren. Hier erklärt sich auch das oben erwähnte Verbot gerichteter Zyklen, da sich in so einem Fall keine adequate Sequenz finden ließe. Nach Durchführung der Berechnungen können Teilergebnisse in den Knoten zwischengespeichert werden. Hierdurch lassen sich bei späterer Modifikationen am Graphen oder nach Rekonfiguration einzelner Teilprozesse, noch gültige Resultate weiterverwenden, was die Abarbeitungszeit z.T. merklich reduziert. Eine Ausführung einzelner Teilpfade im Graphen ist ebenfalls möglich. Die genannten Eigenschaften erlauben eine zügige Entwicklung von Streams und gestatten es Nutzern die Auswirkungen von Modifikationen direkt zu beobachten.

Eine Besonderheit der Software besteht darin, dass sich durch Verbinden von Verarbeitungsknoten neben reinen Daten (rote Kanten) auch komplette Modelle (grün) übertragen lassen (vgl. Abbildung 1). Diese Eigenschaft gestattet es auch Anwendern ohne besondere Vorkenntnisse im Bereich Datenanalyse mit Hilfe einmal erstellter Verarbeitungsströme wiederholt auf jeweils aktuellen Daten basierende Modelle zu erstellen. Hiermit wird der Erfahrung Rechnung getragen, dass datengenerierende Prozesse oftmals in produzierenden Abteilungen von Unternehmen angesiedelt sind und die Ergebnisse der Auswertungen direkt vor Ort benötigt werden.

2.3 Konfiguration von Teilprozessen

Um die erwünschte angemessene Berücksichtigung anwendungsspezifischer Vorgaben zu erreichen, ist ein hohes Maß an Konfigurierbarkeit der zur erstellenden Analyselösungen unerlässlich. Wie bereits im Abschnitt 1 erwähnt sollte der Nutzer aber bei der Wahrnehmung von Konfigurationsaufgaben entlastet werden. Bei der Plattform InformationMiner werden Streams bereits während ihrer Erstellung mit einer Basiskonfiguration versehen. Diese basiert einerseits auf der automatischen Zuordnung der Knoten Ein- und Ausgaben (vgl. 2.2), andererseits auf Ergänzung noch fehlender notwendiger Parameter durch sinnvolle Standardeinstellungen. Der Nutzer gelangt so nach Ergänzung nur weniger Angaben (z.B. dialoggestützte Auswahl der Datenquellen) zu einsatzbereiten Verarbeitungströmen, mit denen sich gewöhnlich bereits akzeptable Analyseergebnisse erzielen lassen.

Für eine detailiertere Konfiguration stehen für alle Knoten generische Dialoge zur Verfügung (Abbildung 2). Diese gestatten den Zugriff auf die mit dem jeweiligen Prozess assoziierten Parameter und werden direkt aus den in der Knotenspezifikation abgelegten Angaben erzeugt.

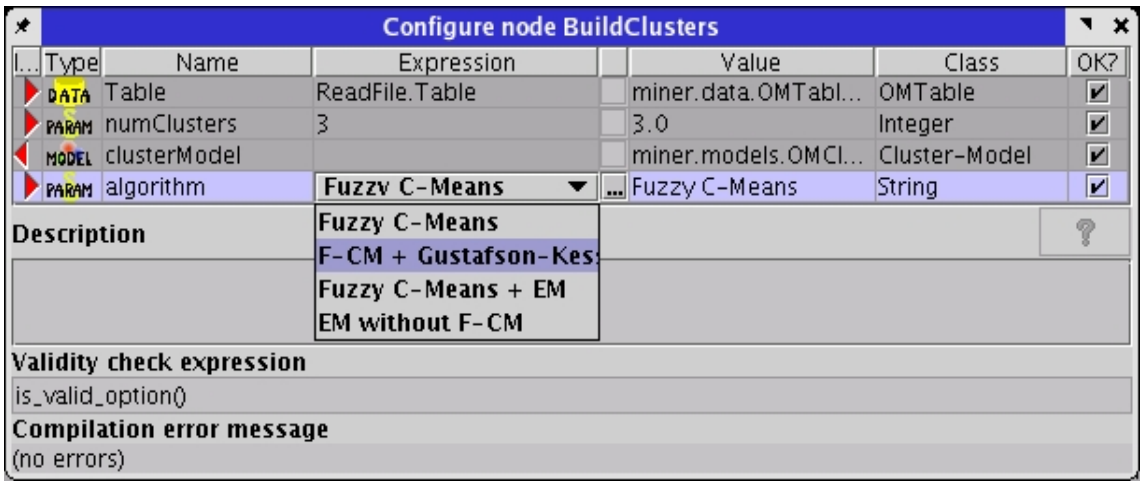

Abbildung 2: Beispiel eines Dialoges zur Knotenkonfiguration

2.4 Einbindung externer Programme

InformationMiner weist eine offene Architektur auf, die eine Integration externer Software gezielt födern soll. Somit können bei Anwendungen, für die bereits Verfahren z.B. zur spezifischen Datenaufbereitung vorhanden sind, jene in die Gesamtlösung übernommen werden. Auch die Berücksichtigung von Algorithmenvarianten, die für das real vorliegende Problem optimiert wurden, ist so möglich. Weiterhin lassen sich auf diese Weise einfache kommandozeilenbasierte Implementierungen neuer Algorithmen zügig in ein komplexeres Data-Mining-System integrieren.

Die Implementierung kann dadurch extern und mittels einer vom Entwickler selbst wählbaren Programmiersprache erfolgen. Da die externen Programme nur auf dem Zielsystem ausgeführt werden müssen, können hier nämlich, im Gegensatz zu den plattformunabhängigen Oberflächenkomponenten, Binärprogramme eingesetzt werden. Durch Ausnutzung sprachspezifischer Bibliotheken und Konstrukte kann beispielsweise der Entwicklungsaufwand reduziert werden. Auch Optimierungen, die mit den in Java verfügbaren Datenstrukturen oder mit einem rein objektorientierten Paradigma nicht realisierbar wären, lassen sich so gezielter durchführen. Obwohl in einigen Fällen zusätzlicher Rechenaufwand, z.B. für die Konvertierung von Datenformaten zu berücksichtigen ist, erweisen sich derartige Lösungen insgesamt oft als vorteilhaft. Tatsächlich wurde die Mehrheit der im Basissystem enthaltenen Lernverfahren auf diese Weise bereitgestellt. Beim Aufruf wird dabei auf eine entsprechende plattformspezifische Variante des Binärprogramms zurückgegriffen. Die eigentliche Anbindung an die von InformationMiner bereitgestellte Oberflächenkomponente erfolgt über die bereits erwähnten Knotendefinitionsklassen, die als Wrapper [5] fungieren. Diese Klassen enthalten Knotenbezeichnung, Namen und Typ der Ein- und Ausgabeparameter sowie, im Falle externer Programme, die Generierung des notwenigen Kommandozeilenaufrufe, weiterhin optional einen Verweis auf ein zu verwendendes Icon. Die Erzeugung und Verwaltung der Knotenobjekte selbst sowie der Konfigurationsdialoge kann mit Hilfe der dort abgelegten Information automatisiert werden. Auf Grundlage dieses Frameworks ist von Entwicklerseite eine zügige Auslieferung sachbereichsspezifischer Paketvarianten möglich.

3 Anwendung

InformationMiner wird gegenwärtig im Rahmen einer Kooperation mit dem Deutschen Sparkassen- und Giroverband (DSGV) eingesetzt. Die Software dient dabei als Basisplattform zur Bearbeitung der im Finanzwesen auftretenden Fragestellungen. Zusätzlich zu den bereitgestellten Standardkomponenten werden in Zusammenarbeit mit dem Kooperationspartner fortwährend weitere, für den Anwendungsbereich relevante Verfahren integriert. Gleichzeitig werden so Erfahrungen aus dem Einsatz unter realen Bedingungen gewonnen.

Insgesamt betont das System eine stark anwendungsorientierte Perspektive, in der die Berücksichtigung problemspezifischer Restriktionen und gegebenfalls die Verwendung angepasster Verfahren als wichtiges Mittel zur Verbesserung der Modellierungs- und Vorhersagequalität gesehen wird. Diese Ausrichtung spiegelt sich auch in der von InformationMiner unterstützten Trennung des Designs von Datenanalyselösungen von deren produktiver Anwendung durch die Endnutzer dieser Lösungen wider. Das nachfolgende einfache Beispielszenario soll stellvertretend einige Abläufe bei der Arbeit mit InformationMiner verdeutlichen. Ziel soll hier die Erzeugung eines Klassifikators für die Irisdaten sein.

3.1 Datenimport und Datenexploration

Der verwendete Datensatz stammt aus dem UCI Machine Learning Repositiory (`http://www.ics.uci.edu/~mlearn/MLRepository.html`) und enthält in Form einer ASCII-Tabelle abgelegte Beispiele zur Blütengeometrie dreier Ochideenarten. Einer Analyse geht zunächst eine einfache Exploration des Datensatzes vorraus:

Um die Daten zu importieren, wird aus der Methodenbibliothek der Knoten für die Dateieingabe gewählt und für das Einlesen der Quelldatei eingerichtet (Abbildung 3). Eine einfache Darstellung der darin enthaltenen Beispiele kann durch

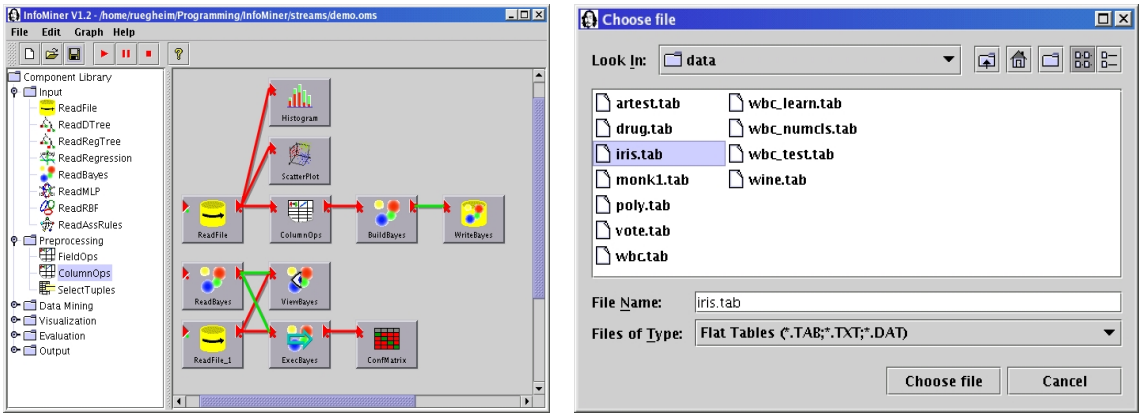

Abbildung 3: Verarbeitungsströme (l.) und Dateiauswahldialog (r.)

einen Scatterplot geleistet werden. Anhand einfacher Histogramme ist eine erste Beurteilung der Verteilung der Attributwerte für die jeweiligen Irisspezies möglich. Die entsprechenden Knoten werden daher mit dem Eingabeknoten verbunden. Bei Ausführung des Verarbeitungströmes werden beide Tools gestartet und auf die eingelesenen Daten angewendet. Wie eine sich zeigt, lässt sich eine der drei Klassen

deutlich abgrenzen, während zwischen den verbleibenden eine geringe Überlappung besteht (Abbildung 4).

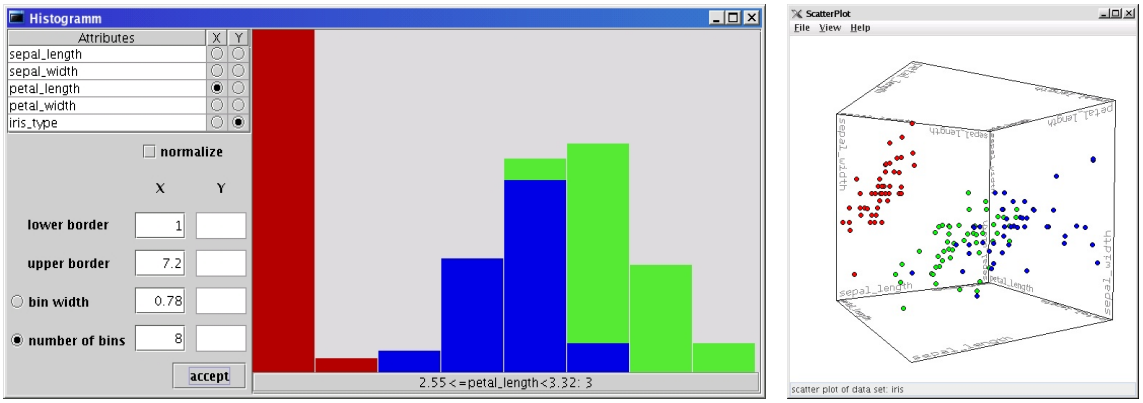

Abbildung 4: Werkzeuge zur Datenexploration

Die Konvertierung oder Filterung der Daten ist mittels der bereitgestellten Vorverarbeitungsknoten möglich. Die im Beispiel eingesetzten Daten müssen nicht aufbereitet werden. Daher wird lediglich der `ColumnOps`-Knoten ergänzt, der Berechnungen unter Rückgriff auf die Attributwerte eines Eintrages gestattet und u.A. etwa Normalisierungen ermöglicht.

3.2 Modellbildung und Evaluierung

Zur Bearbeitung der Klassifikationsaufgabe soll hier ein Bayes-Klassifikator genutzt werden. Der Knoten für die Erstellung des Modells kann direkt mit der Preprocessingausgabe verbunden werden. Das von diesem Knoten erzeugte Modell wird zur späteren Wiederverwendung gespeichert. Eine Beurteilung ist durch Anwendung des wieder importierten Modells auf Testdaten möglich. Mit Hilfe des Knotens zur Berechnung der Verwechslungsmatrix kann das Klassifikationsergebnis zusammengefasst werden. Um auch das Modell selbst darzustellen, wird schließlich der `ViewBayes`-Knoten ergänzt und mit dem Modellimport sowie den Testdaten verbunden. Prinzipiell wäre es auch möglich gewesen, die Evaluierung direkt, also ohne Speicherung des erzeugten Klassifikators durchzuführen. Die hier gewählte Variante mit separater Erzeugung eines Modells entspricht jedoch eher den in der Praxis zu erwartenden Bedingungen. Der zweite, in der unteren Bildschirmhälfte dargestellte Verarbeitungsstrom repräsentiert eine Anwendung des fertigen Modells, auch wenn dies hier lediglich zum Zwecke der Evaluierung erfolgt. Auf ganz ähnliche Weise können andere Klassifikatoren induziert werden, so dass sich Methoden auch direkt vergleichen lassen.

3.3 Feineinstellung der Modellparameter

Bei Verwendung des Bayes-Klassifikators ohne weitere Konfiguration kommt die naive Variante des Verfahrens zum Einsatz. Dabei werden jedoch implizit starke Unabhängigkeitsannahmen getroffen. Ein flexibleres Modell erhält man durch Einstellen eines vollen Bayes-Klassifikators. Der Konfigurationsdialog für `BuildBayes`

bietet diese Option. Nach Modifikation der Verfahrensparameter werden von InformationMiner sämtliche noch in von `BuildBayes` abhängigen Knoten gespeicherten Zwischenergebnisse als ungültig markiert. Die anschließende Wiederholung der Berechnungen liefert ein verbessertes Ergebnis mit einer verringerten Anzahl fehlerhafter Zuordnungen (Visualisierung siehe Abbildung 5).

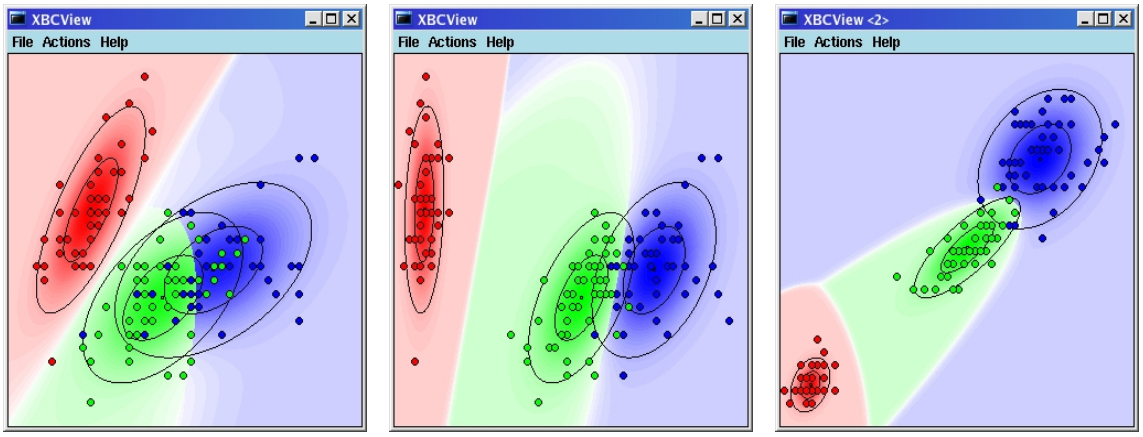

Abbildung 5: Visualisierung eines vollen Bayes-Klassifikators

Für den praktischen Einsatz bietet sich eine Trennung von Modellerzeugung und -einsatz an. Wie oben demonstriert ist das durch die Speicherung erzeugter Modelle in externen Dateien realisierbar. Das Bereitstellen fertiger Streams zur Datenanalyse und Modellbildung ermöglicht auch nicht in der Verwendung von Data-Mining-Methoden geschultem Personal die Durchführung von Untersuchungen oder Modellaktualisierungen. Insbesondere dann, wenn die Ergebnisse nah am datenerzeugenden Prozess benötigt werden, ist diese Vorgehensweise von Vorteil, da Analysetätigkeiten so abteilungsintern durchgeführt werden können.

Ein Vorteil der modularen Prozesskonzeption von InformationMiner tritt hingegen zu Tage, wenn Änderungen in Abläufen eine Anpassung des Analyseprozesses verlangen. Bei einer Umstellung der Datenhaltung kann etwa statt der Nutzung lokaler Dateien der Zugriff auf einen Datenbankserver erwünscht sein. Im Verarbeitungsstrom wären in diesem Fall lediglich die Datenimport- und -exportknoten zu ersetzen.

4 Geplante Erweiterungen

Obwohl die bisher beschriebenen Funktionen bereits zum gegenwärtigen Zeitpunkt ein weites Anwendungsspektrum eröffnen, konnten einige der in der Einleitung benannten Aspekte bisher noch nicht vollständig umgesetzt werden. Geplante Erweiterungen der Kernfunktionalität und der Methodenbibliothek sollen diese offenen Zielstellungen realisieren sowie die Arbeit mit der Plattform weiter vereinfachen.

4.1 Wizard für Knoteneinbettung

Die gegenwärtig genutzte einheitliche Knotenschnittstelle von InformationMiner erlaubt zwar bereits eine zügige Anbindung externer Software an die Data-Mining-

Umgebung, jedoch erfordert dies eine Einarbeitung in die von InformationMiner bereitgestellten Schittstellen- und Parameterklassen sowie grundlegende Java-Programmierkenntnisse. Die vereinheitlichte Definition der Schnittstellen würde jedoch auch eine Integration mittles eines dialoggesteuerten Tools, welches die relevanten Daten vom Nutzer abfragt und in die entsprechenden Quelltexte umsetzt, gestatten. Hierdurch sollte eine weitere Vereinfachung der Verfahrensintegration erreicht werden können.

4.2 Umgang mit hochdimensionalen Datenräumen

Das in der Einleitung angesprochene Problem hochdimensionaler Datenräume ist im Data-Mining seit längerem bekannt. Derarige Problemstellungen wurden in der Vergangenheit häufig nicht direkt gelöst, sondern durch eine vereinfachte Beschreibung approximiert. Hierdurch wurde eine Bearbeitung zwar oft erst ermöglicht, doch gingen diese Ansätze mit einer verringerten Qualität der Ergebnisse einher. Nicht immer ist das für die Anwendung unproblematisch und viele aktuelle Fragestellungen, gerade aus dem Bereich der Bioinformatik, erfordern die Berücksichtigung einer sehr großen Zahl miteinander in Beziehung zu setzender Variablen.

Erfreulicherweise wurden in den vergangenen Jahren einige Verfahren veröffentlicht, die effiziente Operationen auf hochdimensionalen Datenräumen ermöglichen. Hier sind zum Einen verschiedene Ausprägungen der Graphischen Modelle [12, 13] zu nennen, die unter Ausnutzung vorhandener (bedingter) Unabhängigkeiten eine erhebliche Reduktion des Speicherbedarfs und der für Schlußfolgerungen erforderlichen Berechnungen gestatten, ohne die Ergebnisqualität signifikant zu beeinträchtigen. Implementierungen dieser Verfahren werden in absehbarer Zeit für InformationMiner verfügbar sein. Auf der anderen Seite finden sich Verfahren, die auf hierarchischen Fuzzy-Regeln basieren [7]. Diese erlauben den Anwendern u.a. eine Datenexploration auf unterschiedlichen Detailstufen, sind in der Lage Impräzision wiederzugeben und weisen eine hohe Verständlichkeit auf.

4.3 Verarbeitung heterogener Daten

Während Implementierungen vieler Data-Mining Verfahren von einheitlich strukturierten Daten ausgehen, ist diese Eigenschaft in der Praxis nicht immer gegeben. Bei der Verarbeitung von multimedialen Daten ist hiermit sogar definitionsgemäß zu rechnen. Je nach Anwendung und Ausprägung der Heterogenität eignen sich verschiedenste Lösungsansätze. Im Falle unvollständig klassifizierter Daten können Verfahren zum teilüberwachten Lernen [11], die derart inhomogene Daten direkt verarbeiten, genutzt werden. An anderen Fällen ist etwa eine Verwendung unterschiedlicher Importmodule bei zunächst getrennter Verarbeitung der Datenströme möglich. Jene ließen sich dann, z.B. nach einer Featureextraktion, zusammenführen und gemeinsam weiterverarbeiten. Als Alternative böte sich eine völlig separate Bearbeitung der Daten aus unterschiedlichen Quellen mit der Erstellung gegebenenfalls konkurrierender Modelle an. Die damit erstellten Prognosen ließen sich im Anschluß wiederum zusammenfassen.

4.4 Superknoten

Ein bisher für InformationMiner noch nicht realisiertes Konzept besteht in der Nutzung sogenannter Superknoten. Hierunter sind kombinierte Strukturen zu verstehen, die sich zwar der Schnittstelle nach wie einzelne Verarbeitungsknoten verhalten, deren Funktionalität in Wirklichkeit selbst aber mittels einer Kombination anderer Knoten realisiert wurde. Auf diese Weise lassen sich zusätzliche Abstraktionsebenen einführen oder wiederkehrende Teillösungen als funktionale Einheiten zusammenfassen. Zur Konfiguration der internen Knoten könnte Parameter vom Superknoteninterface weitergereicht werden. Dies ist vor allem dann sinnvoll, wenn intern mehrere Knoten mit identischer Parametrisierung verwendet werden sollen (mögliche Implementierung des Baggingverfahrens [3]). Die hierfür erforderliche, der Übertragung von Daten und Modellen analoge Weiterreichung von Parameterwerten ist in InformationMiner vorgesehen, zum gegenwärtigen Zeitpunkt jedoch ebenfalls noch nicht vollständig implementiert.

5 Fazit

Das Data-Mining-System InformationMiner behandelt viele der bisher nur wenig beachteten Probleme bei der Erstellung praxisnaher Data-Mining Lösungen. Durch ein hohes Maß an Flexibilität und die mögliche Integration neuer Algorithmen oder im jeweiligen Problemumfeld erprobter Verfahrensvarianten sollen in kurzer Zeit anwendungspezifische Lösungen unter möglichst guter Ausnutzung des Kontextwissens zusammengestellt werden können. Eine Fokussierung auf spezialisierte Sachbeiche sowie Maßnahmen zur Unterstützung der Nutzer gestatten es, diese Tätigkeiten mit verhältnismäßig geringem Aufwand durchzuführen. Durch das Konzept adaptierbarer Verarbeitungströme kann der Einsatz von Data-Mining nah an die datengenerierenden Prozesse geführt werden. Somit können Analysen unmittelbar dort erfolgen, wo sie benötigt werden.

Auch den bereits erkennbaren Anforderungen an die Lösung künftiger Datenanalyseprobleme wird das Softwarepaket dank geplanter Erweiterungen in absehbarer Zeit genügen können. Kombiniert mit der Einführung neuer Konzepte werden sich somit relevante aktuelle Fragestellungen unter Ausnutzung der spezifischen Informations- und Datensituation effizient beantworten lassen.

Danksagung

Die Entwicklung der in diesem Aufsatz beschriebenen Software wurde durch die DFG gefördert (Förderkennzeichen KR 521/4-1). Die Autoren danken Aljoscha Klose, Christian Borgelt und den zahlreichen weiteren Entwicklern für ihre Arbeit am Grundsystem und ihre Beiträge zum Methodenpool.

Literatur

[1] Agrawal, Rakesh ; Imielinski, Tomasz ; Swami, Arun N.: Mining Association Rules between Sets of Items in Large Databases. In: Buneman, Peter

(Hrsg.) ; Jajodia, Sushil (Hrsg.): *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. Washington, D.C., 26–28 1993, 207–216

[2] Bezdek, James C.: *Pattern Recognition with Fuzzy Objektive Function Algorithms*. New York, NY, USA : Plenum Press, 1981

[3] Breiman, Leo: Bagging Predictors. In: *Machine Learning* 24 (1996), Nr. 2, 123-140. `citeseer.ist.psu.edu/breiman96bagging.html`

[4] Breiman, Leo ; Friedman, Jerome H. ; Olshen, Richard A. ; Stone, Charles J.: *Classification and regression tree*. Belmont, CA : Wadsworth International Group, 1984

[5] DeLine, Robert: Avoiding packaging mismatch with flexible packaging. In: *ICSE '99: Proceedings of the 21st international conference on Software engineering*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1999, S. 97–106

[6] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: From Data Mining to Knowledge Discovery in Databases. In: *AI Magazine* (1996), S. 37–54

[7] Gabriel, Thomas R. ; Berthold, Michael R.: Constructing Hierarchical Rule Systems. In: Berthold, Michael R. (Hrsg.) ; Lenz, Hans-Joachim (Hrsg.) ; Bradley, Elizabeth (Hrsg.) ; Kruse, Rudolf (Hrsg.) ; Borgelt, Christian (Hrsg.): *Proc. 5th International Symposium on Intelligent Data Analysis (IDA 2003)*, Springer Verlag, 2003 (Lecture Notes in Computer Science (LNCS)), S. 76–87

[8] Garlan, David ; Shaw, Mary: An introduction to software architecture. In: Ambriola, Vincenzo (Hrsg.) ; Tortora, Genoveffa (Hrsg.): *Advances in Software Engineering and Knowledge Engineering*. Singapore : World Scientific Publishing Company, 1993, S. 1–39

[9] Gath, Isak ; Geva, Amir B.: Unsupervised optimal fuzzy clustering. In: *IEEE Transactions on pattern Analysis and Machine Intelligence* 11 (1889), S. 773–781

[10] Gustafson, Donald E. ; Kessel, William C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. In: *Proc. of the IEEE Conference on Decision and Control*, 1979, S. 761–766

[11] Klose, Aljoscha ; Kruse, Rudolf: Information Mining with Semi-Supervised Learning. In: *Advances in Soft Computing: Soft Methodology and Random Information Systems*. Berlin, Heidelberg : Springer-Verlag, 2004

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

[13] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, USA : Morgan Kaufman, 1988. – (2nd edition 1992)

[14] Quinlan, J. R.: Induction of Decision Trees. In: Shavlik, Jude W. (Hrsg.) ; Dietterich, Thomas G. (Hrsg.): *Readings in Machine Learning*. Morgan Kaufmann, 1990. – Originally published in *Machine Learning* 1:81–106, 1986

[15] Rügheimer, Frank ; Kruse, Rudolf: Information Miner - a Data Analysis Platform. In: Montseny, Eduard (Hrsg.) ; Sobrevilla, Pilar (Hrsg.): *Proceedings of the Joint EUSFLAT-LFA Conference 2005, September 7-9, Barcelona*. Barcelona, Spain, 2005, S. 1182–1187