Westfälische Wilhelms-Universität Münster

Data Mining: Klassifikations- und Clusteringverfahren

Ausarbeitung
im Rahmen des Projektseminars „CRM für Finanzdienstleister“

im Fachgebiet Wirtschaftsinformatik
am Lehrstuhl für Informatik

Themensteller: Dr. Jens Lechtenbörger
Betreuer: Dr. Jens Lechtenbörger

vorgelegt von: Dirk Loss
Körnerstraße 7
48151 Münster
0251-5395049
dloss@uni-muenster.de

Abgabetermin: 2002-04-15

Inhalt

1 Einleitung

Im Rahmens des Projektseminars „CRM für Finanzdienstleister“ soll festgestellt werden, welche für die Zwecke der Kundenbindung und -pflege nützlichen Informationen ein Kreditinstitut allein durch die Analyse von Zahlungsströmen über seine Kunden ermitteln kann. Diese Aufgabe beinhaltet zwei Fragestellungen: Erstens, welche Informationen lassen sich aus Zahlungsstromdaten überhaupt gewinnen und wie muss dazu vorgegangen werden? Und zweitens, welche dieser Informationen lassen sich für das Customer Relationship Management in einer Bank sinnvoll nutzen?

Das Gewinnen von interessanten, d.h. validen, neuen, nützlichen und verständlichen Informationen aus großen Datenmengen ist Gegenstand und Ziel des *Data Mining* [6]. Im engeren Sinn gehört zum Data Mining nur die eigentliche Analyse der Daten, d.h. das Ermitteln von Mustern und Regeln. Zusammen mit den vorbereitenden Schritten der Bereinigung, Integration, Auswahl und Transformation der Daten sowie den nachfolgenden Schritten der Bewertung und Präsentation der gewonnenen Informationen spricht man dann vom *Knowledge Discovery in Databases* (dem sogenannten KDD-Prozess) [6]. Häufig werden beide Begriffe jedoch synonym verwendet.

Beim Data Mining können verschiedenste Techniken eingesetzt werden: Die zu untersuchenden Daten werden z. B. vorbereitet, zusammengestellt, eingeordnet, geschätzt, gruppiert, assoziiert oder visualisiert [2]. In dieser Arbeit werden zwei dieser Techniken näher beschrieben:

Kapitel 2 stellt das *Clustering* und die grundsätzliche Vorgehensweise bei der Gruppierung von Daten vor. Für das Clustering sind viele verschiedene Verfahren entwickelt worden. Die unterschiedlichen dabei verwendeten Ansätze und ihre Eigenschaften werden im Überblick dargestellt und jeweils im Hinblick auf ihre Vor- und Nachteile beurteilt.

Die Einordnung von Daten ist die Aufgabe von Klassifikationsverfahren. In Kapitel 3 werden die Grundlagen der *Klassifikation* erläutert und mehrere im Bereich des Data Mining gebräuchliche Klassifikationsverfahren vorgestellt.

In Kapitel 4 werden Anwendungsmöglichkeiten von Clustering und Klassifikation im Rahmen des Projektseminars aufgezeigt und besondere Anforderungen genannt, die sich in diesem konkreten Fall an die zu verwendenden Verfahren stellen. Abschließend werden einige Vorschläge gemacht, wie bei der Erprobung der Verfahren sinnvoll vorgegangen werden könnte.

2 Clustering

2.1 Idee, Ziel und Vorgehensweise

Beim Clustering werden Objekte anhand ihrer Merkmale zu Gruppen, sogenannten *Clustern*, zusammengestellt.[1] Die Gruppierung soll dabei so erfolgen, dass die Objekte innerhalb eines Clusters sich möglichst ähnlich, die Cluster untereinander sich aber möglichst unähnlich sind. Andere Bezeichnungen für das Clustering sind: Clusteranalyse, Numerische Taxononmie, Grouping oder Clumping Strategies, Automatische Klassifikation und Q-Analysis [11].

Bei der Anwendung des Clusterings geht man in mehreren Schritten vor [1]:

Zunächst müssen die Ziele und Rahmenbedingungen geklärt werden: In der Regel werden disjunkte Aufteilungen der Datenbasis (*Partitionierungen*) gesucht, d.h. nach Abschluss des Verfahrens soll jedes Objekt zu genau einem Cluster gehören. Kann ein Objekt auch Element mehrerer Cluster sein, spricht man von Fuzzy Clustering [10] oder Clumping Methods [9]. Außerdem ist z. B. festzulegen, ob die Anzahl der zu bildenden Cluster vorgegeben ist oder automatisch ermittelt werden soll, in welcher Größenordnung sich die Anzahl der Objekte voraussichtlich bewegt und wie lang die Rechenzeit des Algorithmus sein darf.

Dann werden die zu clusternden Objekte ermittelt und bereitgestellt. Häufig ist eine Vorbereitung der Daten sinnvoll, z. B. durch Auslassen von redundanten Attributen, Normalisieren von Werten, Entfernen von Ausreißern oder Ermitteln von abgeleiteten Merkmalen [6, Kapitel 3].

Im Anschluss muss festgelegt werden, wie die Ähnlichkeiten zwischen den Objekten sinnvoll gemessen werden können. In der Literatur sind hierzu eine Vielzahl von Ähnlichkeits- und Distanzmaßen definiert worden. Übersichten finden sich z. B. in [1, 9]. Ähnlichkeiten und Unähnlichkeiten bzw. Distanzen sind dabei ineinander transformierbar [9]; allgemein spricht man von Proximitätsmaßen [1]. Distanzen sind Unähnlichkeitsmaße mit speziellen Eigenschaften [10].

Die entscheidende Rolle bei der Auswahl eines Ähnlichkeits- oder Distanzmaßes spielt die Skalierung der Merkmale: Euklidische Distanzen (oder ihre Verallgemeinerungen wie die L_p-Metriken und die Mahalanobis-Distanz) machen nur bei metrischen (d.h. intervall- oder verhältnisskalierten) Daten Sinn. Ordinale Daten können in Form von Rangreihen metrischen Distanzmaßen unterworfen werden [6]. Bei binären Daten verwendet man als Ähnlichkeitsmaß grundsätzlich die Anzahl der Übereinstimmungen, diese kann allerdings auf unterschiedliche Weisen normiert werden (Simple-Matching, Jaccard-Koeffizient, u.

[1] In dieser Arbeit wird generell von *Objekten* gesprochen, die *Attribute* mit bestimmten *Ausprägungen* besitzen. Ein *Merkmal* ist eine bestimmte Kombination von Attribut und Ausprägung. Alternative Bezeichnungen wären zum Beispiel statt Objekt Datensatz oder Tupel, statt Attribut Dimension, statt Ausprägung Wert und statt Merkmal Eigenschaft.

a.). Nominal skalierte Daten lassen sich binär codieren, was allerdings die Anzahl der zu untersuchenden Attribute in der Regel stark erhöht: Aus einem nominal skalierten Attribut mit 2^p Ausprägungen werden dann mindestens p Attribute mit binärer Ausprägung. Für die Behandlung gemischt skalierter Daten gibt es zwei verschiedene Möglichkeiten: die Transformation der Daten auf das niedrigste gemeinsame Skalierungsiveau (was zu hohem Informationsverlust führt) oder die getrennte Berechnung der Ähnlichkeiten pro Skalierungsniveau mit anschließender Aggregation (wobei auf eine sinnvolle Gewichtung zu achten ist) [1, 6].

Nach der Festlegung eines Proximitätsmaßes wird ein konkreter Clusteringalgorithmus ausgewählt. Bei der Beurteilung ist zu berücksichtigen, welche der generellen Anforderungen an Clusteringverfahren (Skalierbarkeit, Fähigkeit mit unterschiedlichen Datentypen umzugehen, Entdeckung von Clustern mit beliebiger Form, geringe Ansprüche an Inputparameter, Umgehen mit Ausreißern, Unabhängigkeit von der Ordnung der Daten, Behandlung hochdimensionaler Daten, Verwendbarkeit von Nebenbedingungen (constraints), Interpretierbarkeit und Anwendbarkeit der Ergebnisse [6]) in der vorliegenden Analysesituation von besonderer Bedeutung sind. Eine Klassifizierung der verschiedenen Clusteringverfahren nach ihren Ansätzen liefert der nächste Abschnitt.

Viele Clusteringverfahren stellen eine Zielfunktion auf, die die Güte einer Partititionierung bewertet, und versuchen, diese Funktion zu maximieren (oder sie versuchen umgekehrt, eine entsprechende Kostenfunktion zu minimieren). Auf diese Weise betrachtet wird das Gruppierungsproblem [11] zu einem Optimierungsproblem [10]. Der naive Ansatz, eine vollständige Enumeration aller möglichen Gruppierungen und die anschließende Auswahl der am besten bewerteten Lösung, ist schon bei einer kleinen Anzahl von Objekten aus Zeitgründen nicht durchführbar [9, 11]. So gibt es beispielsweise schon für die Gruppierung von 100 Objekten in 5 Cluster mehr als 10^{69} Möglichkeiten.[2] Deshalb gehen alle Clusteringverfahren heuristisch vor und suchen nicht nach der besten, sondern nach einer brauchbaren Lösung bei vertretbarem Aufwand [9].

Nach dem Ablauf des implementierten Algorithmus wird die gefundene Partitionierung auf ihre Stabilität bzw. Sensitivität analysiert und interpretiert [1]. Gegebenenfalls muss das Verfahren mit veränderten Parametern wiederholt werden, bis die Ergebnisse brauchbar sind.

Klassische Anwendungsgebiete des Clustering sind z. B. die Kunden- oder Produktsegmentierung im Marketing, die Typisierung von Verhaltensweisen in der Psychologie [9] und die Taxonomierung von Lebewesen in der Biologie [7, 11]. Clustering kann außerdem überall da eingesetzt werden, wo man Daten auf wenige überschaubare Einheiten reduzieren will [11], um diese dann jeweils im Anschluss intensiver untersuchen zu können [9]. Ein spezieller Einsatzzweck ergibt sich in Verbindung mit Klassifikationsverfahren: Hier werden durch das Clustering zunächst die Klassen *gebildet*; danach werden neue Objekte mit Klassifikationsverfahren diesen Klassen *zugeordnet* [1]. Umgekehrt setzen einige Clusteringverfahren Klassifikation zur Performance-Steigerung ein, indem sie zunächst nur eine Teilmenge clustern und die restlichen Elemente mit Klassifikationsverfahren den gebildeten Clustern zuordnen.

[2] Die Anzahl möglicher Partitionierungen von n Objekten in k Cluster ist: $\frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n$ (Stirlingsche Zahlen)

2.2 Unterschiedliche Ansätze im Überblick

Clusteringverfahren können in verschiedene Kategorien eingeteilt werden: [6, 10]

Hierarchische Verfahren

Bei *agglomerativen* hierarchischen Verfahren bildet im Anfgangszustand jedes Objekt ein eigenes Cluster. Alle Cluster werden dann paarweise verglichen. Dazu ist neben dem Ähnlichkeitsmaß für Objekte auch festzulegen, wie Ähnlichkeiten zwischen *Clustern* von Objekten gemessen werden sollen. Übersichten zu solchen Ähnlichkeits- bzw. Distanzmaßen für Cluster finden sich z. B. in [5, 9]. Die beiden ähnlichsten Cluster werden zu einem größeren Cluster fusioniert. Die Schritte Distanzbildung und Fusion werden iterativ wiederholt bis zu einem vorgegebenen Abbruchkriterium (z. B. der vorgegebene Clusteranzahl oder dem Wert einer Gütefunktion für die Partitionierung) oder bis sich schließlich alle Objekte in einem Cluster befinden ('bottom-up').

Divisive hierarchische Verfahren gehen umgekehrt ('top-down') vor: Ausgehend von einem Cluster, in dem sich alle Objekte befinden, werden die Cluster sukzessive immer weiter in kleinere Cluster aufgeteilt. Mit jeder Iteration erhöht sich somit die Anzahl der Cluster um eins. Divisive Verfahren sind rechenaufwändiger und weniger verbreitet als agglomerative[10], können aber bei einer sehr geringen Clusteranzahl vorteilhaft sein.

Der Ablauf von hierarchischen Verfahren lässt sich in Dendrogrammen darstellen. Ein Beispiel mit fünf Objekten ist in Abbildung 2.1 dargestellt. Ein agglomeratives Verfahren würde zunächst die beiden Objekte 5 und 3, die eine Distanz von 0.21 Einheiten aufweisen, zu einem Cluster zusammenfassen, danach die Objekte 1 und 4 (Distanz: 0.40) usw. Im letzten Schritt würden die beiden Cluster (1, 4, 2) und (5, 3) fusioniert (Distanz: 1.13). Ein divisives Verfahren würde hingegen das anfängliche Cluster (1, 2, 3, 4, 5) zunächst in die beiden Cluster (1, 4, 2) und (5, 3) aufteilen und dann mit jedem dieser Cluster genauso weiterverfahren, bis schließlich jedes Cluster nur noch aus einem Objekt besteht. Bei der Auswahl der besten Partitionierung wird das Dendrogramm an einer bestimmten Stelle vertikal durchschnitten: Im Beispiel würde ein Schnitt z. B. beim Distanzniveau 0.50 zu den drei Clustern (1, 4), (2) und (5, 3) führen.

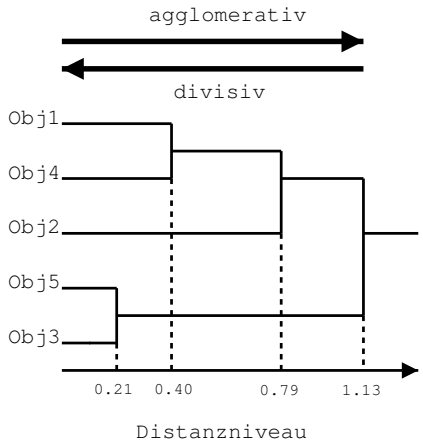

Abbildung 2.1: Beispiel für ein Dendrogramm (nach [5, S. 129])

Algorithmen: AGNES (agglomerativ), DIANA (divisiv), BIRCH (erst hierarchisch, dann partitionierend als Verfeinerung [6], CURE [10], Chameleon [10], ROCK (speziell für kategorielle Daten).

Vorteile: Die Anzahl der Cluster muss nicht vorgegeben werden. Hierarchische Verfahren eignen sich besonders, wenn man an den Verbindungen zwischen den Clustern interessiert ist (z. B. bei einer Taxonomie in der Biologie) [9, 10]. Implementierungen sind weit verbreitet.

Nachteile: Aufgrund der nötigen paarweisen Distanzbildung für alle Objekte sind hierarchische Verfahren schlecht skalierbar und in der Praxis auf wenige tausend Elemente beschränkt [5]. Einmal getroffene Zusammenfassungen von Clustern können nicht wieder rückgängig gemacht werden [5]. Das Clustering selbst orientiert sich an lokalen Kriterien. Gute lokale Fusionsentscheidungen müssen aber nicht zu global guten Partitionierungen führen [10]. Hierarchische Verfahren haben Probleme mit Ausreißern und nicht-konvexen Clustern [10].

Partitionierende Verfahren

Partitionierende Verfahren gehen von einer zufälligen Anfangspartitionierung aus und ordnen die Objekte schrittweise so zwischen den Clustern um, dass die Güte der Gruppierung sich immer weiter verbessert [1]. Dazu wird in jeder Iteration pro Cluster ein Zentroid (durch Mittelwertbildung) oder Repräsentant spezifiziert, und die Objekte werden dann demjenigen Cluster zugeordnet, dessen Zentrum sie am ähnlichsten sind [6]. Das Verfahren endet, wenn sich die Güte der Partitionierung nicht mehr verbessert. Neuere Algorithmen (CLARA, CLARANS) verbessern die Skalierbarkeit klassischer Verfahren wie k-means und PAM, indem sie die Cluster zunächst nur anhand von Teilmengen der Datenbasis bilden und anschließend die restlichen Objekte diesen Clustern zuordnen.

Algorithmen: k-means (arbeitet mit Zentroiden, daher nur für metrische Daten), PAM bzw. k-medoids (arbeitet mit Repräsentanten; ist langsamer, aber robuster als k-means) [6], k-prototypes (für gemischt skalierte Daten und große Datenmengen) [8], CLARA (mehrmaliges PAM auf Stichproben) [6], CLARANS

Vorteile: k-means ist für kleine Clusteranzahlen recht effizient (der Rechenaufwand wächst linear mit den Anzahlen der Objekte, Cluster und Attribute) [10]. Implementierungen sind weit verbreitet.

Nachteile: Die Clusteranzahl muss vorgegeben werden. Die Ergebnisse werden stark beeinflusst von der Wahl der Startgruppierung und dem Umordnungsverfahren [1]. Oft werden nur lokale Optima der Gütefunktion ermittelt. k-means und k-medoids sind nicht für große Datenmengen und nicht bei komplexen Clusterformen geeignet [6].

Dichtebasierte Verfahren

Dichtebasierte Verfahren sind für Raumdaten entwickeln worden und gehen daher grundsätzlich von metrischen Daten aus. Sie ermitteln solche Bereiche im Merkmalsraum, die besonders dicht von Objekten belegt sind. Jedes Objekt in einem Cluster besitzt in seiner Umgebung entweder (a) eine festgelegte Mindestanzahl von anderen Objekten oder (b) zumindest ein anderes Objekt, das zu diesem Cluster gehört – für das also eine der Bedingungen (a) oder (b) erfüllt ist. Objekte, die zu keinem Cluster gehören, weil sie in zu dünn besiedelten Bereichen liegen, werden als Ausreißer angesehen [6].

Algorithmen: DBSCAN, OPTICS, DENCLUE [6], CLIQUE und WaveCluster (beide dichte- und gridbasiert[6]), MAFIA (schnellere, bessere Modifikation von CLIQUE [10]

Vorteile: Dichtebasierte Verfahren können Cluster beliebiger Form erkennen (im Gegensatz z. B. zu Verfahren, die mit metrischen Distanzmaßen arbeiten und nur konvexe Cluster bilden). Sie sind auch für große Datenmengen geeignet.

Nachteile: Die Qualität der gefundenen Partitionierung hängt stark von der Wahl der Inputparameter (Umgebungsgröße, Mindestzahl von Objekten) ab.

Weitere Verfahren

Gridbasierte Verfahren (z. B. STING, WaveCluster, CLIQUE) unterteilen den Merkmalsraum gitterartig in eine endliche Anzahl von Zellen und führen das Clustering ausschließlich auf diesen Zellen aus. Sie wurden für hochdimensionale metrische Daten entwickelt und zeichnen sich besonders durch ihre hohe Verarbeitungsgeschwindigkeit aus, die nicht von der Anzahl der Objekte, sondern nur von der Anzahl der Zellen abhängt [6].

Stochastische Verfahren sehen die Objekte als Realisierungen von Zufallsvariablen. Es wird angenommen, dass jeder Cluster durch eine bestimmte, unbekannte Verteilung gekennzeichnet ist. Diese wird geschätzt, und die Objekte werden anhand der Verteilungen den Clustern zugeordnet [9, 6]. Diese Verfahren (COBWEB, CLASSIT, AutoClass) eignen sich nicht für große Datenmengen.

Außerdem werden zum Clustering auch Neuronale Netze in Form von Self-organizing feature maps (SOMs) eingesetzt [10].

3 Klassifikation

3.1 Idee, Ziel und Vorgehensweise

Klassifikation ist das Einordnen von Objekten in vorgegebene Klassen. Die Frage lautet: In welche Klasse passt ein gegebenes Objekt aufgrund seiner individuellen Merkmalskombination am besten? In der Statistik spricht man meist von Diskriminanzanalyse [4], in der KI von Mustererkennung (engl.: pattern recognition). Manche Autoren (z. B. [4, 7, 11]) verwenden den Begriff Klassifikation in der Bedeutung 'Unterteilung einer Menge von Objekten in Klassen' [1]. In dieser Arbeit wird diese Aufgabenstellung wie in [3, 6] nicht als Klassifikation, sondern als Clustering aufgefasst: Clusteringverfahren bilden Klassen, Klassifikationsverfahren ordnen Objekte in vorgegebene Klassen ein. Klassifikation ist außerdem zu unterscheiden von der Vorhersage: Klassifikationsverfahren sagen Klassenzugehörigkeiten (kategorielle Daten) voraus, Vorhersageverfahren schätzen Ausprägungen von Attributen (kontinuierliche Daten).

Anwendungsbeispiele für die Klassifikation sind z. B. die Erkennung von Schriftzeichen, das Diagnostizieren einer Krankheit oder die Überprüfung der Kreditwürdigkeit [1, 4, 6].

Klassifikationsverfahren laufen in zwei wesentlichen Schritten ab:

1. Lernphase (Erstellung eines Klassifikators): Aus der Datenbasis werden zufällig einige Objekte ausgewählt und zu einer Trainingsmenge (engl.: training data set) zusammengestellt. Zu jedem Trainingsobjekt muss in einem zusätzlichen Attribut die Klasse vorgegeben bzw. vermerkt werden, in die es gehört. Man spricht daher von überwachtem Lernen (engl.: supervised learning) [6]. Anhand der klassifizierten Trainingsdaten wird mittels eines Algorithmus ein Modell (z. B. ein Satz von Regeln) erstellt, das zu Merkmalskombinationen die zugehörige Klasse angeben kann. Dieses Modell bezeichnet man als *Klassifikator*.

2. Klassifikationsphase (Anwendung des Klassifikators): Die zu klassifizierenden Objekte werden dem Modell unterworfen. Als Ergebnis wird zu jedem Objekt seine Klasse ausgegeben.

Wie beim Clustering lässt sich ein Schritt der Datenvorbereitung (Ausreißerentfernung, Normalisierung, Transformation, Konstruktion von neuen Attributen, usw.) vorschalten, um bessere Ergebnisse zu erhalten [6].

Es ist darauf zu achten, dass das ermittelte Modell nicht zu genau an die Trainingsdaten angepasst ist, sondern flexibel genug bleibt, auch neue Daten korrekt zu klassifizieren (Problem des 'overfitting'). Daher sollte die Brauchbarkeit des Klassifikators vor der Anwendung überprüft werden, z. B. anhand von Testdaten [6]. Neben der Vorhersagegenauigkeit sind auch die Geschwindigkeit, die Robustheit bei Ausreißern, die Eignung für große Datenmengen und die Interpretierbarkeit der Ergebnisse von Interesse [6].

3.2 Unterschiedliche Ansätze im Überblick

Entscheidungsbäume

Die Klassifikation eines Objekts mit einem Entscheidungsbaum erfolgt, indem man von der Wurzel ausgehend die sich an den Knoten befindlichen Attribute prüft und je nach vorliegender Ausprägung den entsprechenden Verzweigungen folgt. Das Klassifikationsergebnis steht fest, sobald man an einem Blattknoten angelangt ist: die Beschriftung des Blattknotens gibt dann die Klasse an, in die das Objekt einzuordnen ist.

Abbildung 3.1 zeigt dazu ein Beispiel. Es soll entschieden werden, ob einem Bankkunden ein Kredit gewährt wird oder nicht. Die Entscheidung wird von der Art des Kunden (Altkunde oder Neukunde), seinem Einkommen sowie den vorhandenen Sicherheiten beeinflusst. Als Trainingsdaten wurden sieben Kunden mit unterschiedlichen Merkmalskombinationen ausgewählt, für die bekannt war, ob der Kredit gewährt wurde oder nicht. Anhand dieser Trainingsdaten ist der abgebildete Entscheidungsbaum ermittelt worden. Mit seiner Hilfe kann in Zukunft z. B. ein neuer Kunde mit den Ausprägungen "Neukunde, hohes Einkommen, keine Sicherheiten" in die Klasse der Kunden eingeordnet werden, die einen Kredit bekommen (in diesem Fall aufgrund seines hohen Einkommens).

Kundentyp	Einkommen	Sicherheiten	Kredit
Neukunde	hoch	vorhanden	ja
Neukunde	niedrig	keine	nein
Neukunde	niedrig	vorhanden	nein
Altkunde	hoch	vorhanden	ja
Altkunde	hoch	keine	ja
Altkunde	niedrig	vorhanden	ja
Altkunde	niedrig	keine	nein

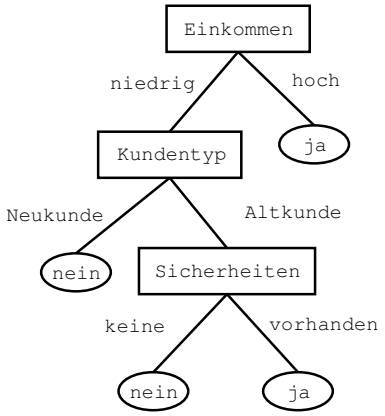

Abbildung 3.1: Trainingsdaten und daraus konstruierter Entscheidungsbaum

Die Konstruktion von Entscheidungsbäumen erfolgt in einem rekursiven Divide-and-Conquer Verfahren anhand der Trainingsdaten: In jedem Knoten wird mit einer informationstheoretischen Kennzahl entschieden, anhand welches Attributs die nächste Verzweigung geschehen soll. Für jede vorkommende Ausprägung dieses Attributs wird eine Verzweigung gebildet und der Algorithmus mit denjenigen Traingsobjekten rekursiv weitergeführt, die diese Ausprägung besitzen. Gehören an einer Verzweigung alle Trainingssobjekte zur gleichen Klasse, wird ein mit diese Klasse beschrifteter Blattknoten erstellt. Wenn schon alle Attribute zum Test verwendet wurden, geschieht die Beschriftung mit der in der Teilmenge häufigsten Klasse [6].

Dieser Basisalgorithmus (ID3) ist in verschiedener Weise weiterentwickelt worden. Neben Unterschieden in dem verwendeten informationstheoretischen Maß (z. B. Gini-Index statt Information-Gain) sind die wesentlichen Verbesserungen die Einsatzmöglichkeit auch bei kontinuierlichen Merkmalen (z. B. Einkommen in Euro) sowie der Einsatz von Pruning-

Verfahren, die durch gezieltes Entfernen von Verzweigungen die Komplexität des Entscheidungsbaumes verringern und damit seine Generalisierbarkeit und Interpretierbarkeit verbessern [6]. Außerdem gibt es spezielle Entscheidungsbaumverfahren für den Einsatz bei großen Datenmengen (SLIQ, SPRINT, RainForest).

Algorithmen: ID3, C4.5, C5.0, CART, CHAID, QUEST, SLIQ, SPRINT. Literaturhinweise auf weitere Algorithmen finden sich in [6].

Vorteile: Entscheidungsbäume können sehr einfach in leicht interpretierbare Wenn-Dann-Regeln konvertiert werden, indem man alle Pfade von der Wurzel bis zu den Blattknoten durchläuft und auflistet. Aufgrund dieser Eigenschaft werden sie auch im Anschluss an Clusteringverfahren eingesetzt, um eine gewonnene Partitionierung besser verstehbar zu machen [8]. Dadurch, dass die Attribute, die am meisten zur Klassifikation beitragen, in die Nähe der Wurzel des Entscheidungsbaums gesetzt werden, können Entscheidungsbaumverfahren auch zur Priorisierung von Attributen dienen [2].

Nachteile: Bei den meisten Verfahren müssen die Trainingsdaten komplett im Hauptspeicher gehalten werden [6].

Bayes-Klassifikation

Bei der Bayes-Klassifikation wird ein Objekt derjenigen Klasse zuordnet, die für seine individuelle Merkmalskombination am wahrscheinlichsten ist.

Gegeben seien k Klassen $C_1, C_2, \ldots, C_k$ und ein zu klassifizierendes Objekt mit m Merkmalen $x_1, x_2, \ldots, x_m$, die im Merkmalsvektor X zusammengefasst werden. $P(C_i|X)$ gibt dann die Wahrscheinlichkeit an, dass das Objekt mit dem gegebenen Merkmalsvektor X zur Klasse C_i gehört. Gesucht ist diejenige Klasse C_i, für die diese Wahrscheinlichkeit am größten ist; in diese wird das Objekt eingeordnet. Nach dem Satz von Bayes gilt:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Da $P(X)$ konstant ist, reicht es zu prüfen, für welche Klasse der Zähler $P(X|C_i)P(C_i)$ maximal wird. Die Auftretenswahrscheinlichkeiten der Klassen $P(C_i)$ werden anhand der relativen Häufigkeiten in den Trainingsdaten geschätzt oder auch vereinfacht als gleichverteilt angenommen. Um die Berechnung von $P(X|C_i)$ zu vereinfachen, wird angenommen, dass bei gegebener Klasse die Ausprägungen der Attribute unabhängig voneinander sind. Da die Unabhängigkeit in der Praxis nicht immer gegeben ist (z. B. wird in der Klasse der Privatkunden das Attribut Einkommen vom Attribut Beruf abhängen), spricht man hier von "naiver" Bayes-Klassifikation. (Mit *Bayesian belief networks* können auch Abhängigkeiten zwischen den Attributen dargestellt werden [6]). Unter der Annahme der Unabhängigkeit ist $P(X|C_i)$ das Produkt aus den bedingten Wahrscheinlichkeiten für die in X vorkommenden Ausprägungen x_j der einzelnen Attribute:

$$P(X|C_i) = \prod_{j=1}^{m} P(x_j|C_i)$$

Diese Einzelwahrscheinlichkeiten $P(x_j|C_i)$ können wiederum anhand der relativen Häufigkeiten in den Trainingsdaten geschätzt werden. Dazu betrachtet man die Trainingsdaten nach Klassen getrennt und setzt die Anzahl der Objekte mit der Ausprägung x_j für das

j-te Attribut ins Verhältnis zur Anzahl aller Objekte dieser Klasse. Bei kontinuierlichen Merkmalen erfolgt die Schätzung anhand einer angenommenen Verteilungsfunktion [6].

Vorteile: Naive Bayes-Klassifikation erzielt bei Anwendung auf großen Datenmengen eine hohe Genauigkeit und eine vergleichbare Geschwindigkeit wie Entscheidungsbaumverfahren und Neuronale Netze [6].

Nachteile: Wenn die Annahmen über Verteilungen und die Unabhängigkeit der Attribute ungerechtfertigt sind, werden die Ergebnisse ungenau.

Neuronale Netze

Neuronale Netze bestehen aus mehreren Knoten (Neuronen) die miteinander verbunden sind und sich gegenseitig aktivieren. In ihrer gebräuchlisten Form als *fully-connected, feed-forward, multilayer perceptrons* sind die Neuronen in mehreren Schichten angeordnet (Eingabeschicht, eine oder mehrere verborgene Schichten, Ausgabeschicht) und jedes Neuron ist mit allen Neuronen der nachfolgenden Schicht verbunden. Die Verbindungen zwischen den Neuronen sind mit anfangs zufälligen Gewichten belegt. Ein Beispielnetz mit drei Eingabeneuronen, zwei verborgenen Neuronen und zwei Ausgabeneuronen findet sich in Abbildung 3.2; Gewichte sind dort mit w_{ij} bezeichnet.

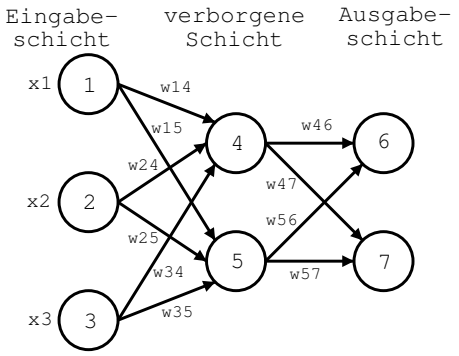

Abbildung 3.2: Beispiel für ein Neuronales Netz (Quelle: nach [6, S. 309])

In der Lernphase werden die Merkmale des Trainingsobjekts (im Beispiel seine drei Merkmale x_1, x_2 und x_3) als numerische Daten an entsprechende Neuronen der Eingabeschicht übergeben und von dort aus gewichtet an die Neuronen der ersten verborgenen Schicht weitergeleitet. Jedes Neuron in der verborgenen Schicht bildet die gewichtete Summe über die erhaltenen Eingabedaten, wendet auf das Zwischenergebnis eine Aktivierungsfunktion an (z. B. eine Schwellenwertfunktion oder eine s-förmige Funktion wie die Sigmoidfunktion $f(x) = \frac{1}{1+e^{-x}}$) und leitet das Ergebnis an die Neuronen der nächsten Schicht weiter. Anhand der Ausgaben der Neuronen in der Ausgabeschicht lässt sich schließlich das Klassifikationsergebnis ablesen. Normalerweise setzt man pro Klasse ein Ausgabeneuron ein, das als einziges aktiviert wird, wenn die zugehörige Klasse als Ergebnis herauskommen soll [6]. Im Beispielnetz hängt also das Klassifikationsergebnis (Klasse 1 oder Klasse 2) davon ab, welches der beiden Neuronen 6 und 7 aktiviert wird.

Das Lernen erfolgt nach dem sogenannten Backpropagation-Ansatz: Die Ausgabe des Netzes wird mit dem erwünschten Ergebnis (der in den Trainingsdaten vermerkten korrekten Klasse) verglichen. Die Differenz (d.h. der Fehler) wird in umgekehrter Richtung

an das Netz zurückgegeben und sorgt für eine langsame Anpassung der Gewichte. Mit der Zeit werden solche Gewichtskombinationen ermittelt, die die Trainingsdaten immer besser klassifizieren. Die Lernphase wird abgebrochen, wenn sich kaum noch Veränderungen ergeben, die Klassifikation gut genug erscheint oder nach einem zeitlichen Kriterium [6]. Die Anwendung von gelernten Klassifkationsregeln geschieht dann sehr schnell.

Vorteile: Neuronale Netze können sehr gut mit Ausreißern umgehen und solchen Objekten, deren Merkmalskombination nicht in der Trainingsmenge vorgekommen ist [6].

Nachteile: Die erlernten Gewichte sind kaum zu interpretieren. Somit lässt sich das Klassifikationsergebnis nicht erklären [2]. Inzwischen gibt es allerdings einige Verfahren, die versuchen, aus den Gewichten Regeln abzuleiten [6]. Die Trainingsphase dauert sehr lange, besonders wenn die Anzahl der Attribute groß ist. In diesem Fall kann es auch sein, dass gar keine gute Lösung gefunden wird [2]. Neuronale Netze erfordern besondere Sorgfalt bei der Datenvorbereitung, z. B. bei der Normalisierung der Daten. Kategorielle Daten müssen vorher sinnvoll in metrische Daten umgewandelt werden, was problematisch sein kann [2]. Die Alternative, für jede mögliche Ausprägung ein eigenes Eingabeneuron einzusetzen, lässt die Trainingszeiten enorm ansteigen und verschlechtert die Qualität der Ergebnisse. Eine dem Problem angepasste Topologie des Neuronalen Netzes (Anzahl der verborgenen Schichten, Anzahl der Neuronen jeder Schicht) ist nicht vorgegeben und muss anhand von Erfahrungswerten festgelegt werden.

k-nächste-Nachbarn-Verfahren

Die Idee dieses Verfahrens besteht darin, ein Objekt in die gleiche Klasse einzuordnen wie ähnliche Objekte aus der Trainingsmenge. Dazu werden diejenigen k Trainingsobjekte ermittelt, welche die größte Ähnlichkeit mit dem zu klassifizierenden Objekt besitzen. Gemessen wird wird dies mit einem festzulegenden Ähnlichkeitsmaß. Die Klasse, die unter diesen k Objekten am häufigsten auftritt, wird als Klassifikationsergebnis ausgegeben.

Vorteile: Das Verfahren ist grundsätzlich sowohl für metrische als auch für kategorielle Merkmale anwendbar, das Ähnlichkeits- bzw. Distanzmaß muss nur entsprechend sinnvoll definiert werden [4]. Die Lernphase entfällt praktisch: alle Trainingsdaten werden nur zwischengespeichert und erst ausgewertet, wenn neue Objekte zu klassifizieren sind ('lazy learning') [6].

Nachteile: Die Klassifikationsphase ist sehr aufwändig. Für jeden einzelnen Klassifikationsvorgang muss die gesamte Trainingsmenge zur Verfügung stehen und nach ähnlichen Objekten durchgearbeitet werden [4]. Die Anzahl der zu berücksichtigenden Nachbarn k muss von außen festlegt werden. Für größere Werte von k nimmt der Aufwand noch zu [6].

Weitere Verfahren

Genetische Algorithmen kodieren Klassifikationsregeln in Form von Bitstrings und verwenden die genetischen Operatoren Rekombination und Mutation auf Populationen solcher Strings, um sie zu verändern und die Klassifikationsgenauigkeit bezogen auf einen Testdatensatz zu erhöhen.

Andere Verfahren wie Assoziationsbasierte Klassifikation, Fallbasiertes Schließen (Case-based Reasoning) sowie FuzzySet- und RoughSet-Techniken sind in [6] beschrieben.

4 Anwendung im Rahmen des Projektseminars

Im Projektseminar soll festgestellt werden, welche für die Zwecke der Kundenbindung und -pflege nützlichen Informationen ein Kreditinstitut allein durch die Analyse von Zahlungsströmen über seine Kunden ermitteln kann, wie dabei vorzugehen ist und welche Verfahren dabei einsetzbar sind. Die Umsetzbarkeit der Überlegungen soll anhand einer prototypisch realisierten Mining-Software demonstriert werden.

Ausgangsdatenbasis für die Analysen sind die einzelnen Zahlungsvorgänge (Überweisungen, Ein-/Auszahlungen, Lastschriften, Gutschriften, Scheck- und Kartenbelastungen, Scheckeinreichungen usw.) jedes Kunden des Kreditinstituts.

4.1 Mögliche Analysen

Die Suche nach für den CRM-Prozess verwendbaren Informationen lässt sich sowohl direkt auf den Ausgangsdaten, d.h. den Zahlungsströmen, als auch auf abgeleiteten Daten durchführen.

Clustering und Klassifikation auf Basis der Ausgangsdaten

Denkbar wären z. B. folgende Analysen:

- Clustering von Zahlungsvorgängen über alle Kunden hinweg. Dieser Ansatz könnte im Rahmen einer eher explorativen Voranalyse Sinn machen, um die Typen von Zahlungsvorgängen kennenzulernen. Dazu ist jedoch ein besonders skalierbares Verfahren notwendig (s. u.).
- Clustering von Zahlungsvorgängen eines Kunden, zur Erkennung von Buchungstypen (oder periodischen Zahlungen) oder in Verbindung mit einer Ausreißer-Analyse zur Erkennung von Unregelmäßigkeiten (z. B. Änderungen des Lebensstils).
- Klassifikation von Zahlungsvorgängen über alle Kunden hinweg, z. B. die Zuordnung zu Clustern, die mit Hilfe eines zufällig ausgewählten Teils der Zahlungsvorgänge ermittelt wurden (zur Performance-Steigerung des Clusterings im Rahmen einer explorativen Voranalyse)
- Klassifikation von Zahlungsvorgängen eines Kunden, um z. B. interessante von uninteressanten Buchungen des Kunden zu trennen (Datenreduktion).

Clustering und Klassifikation auf Basis von abgeleiteten Daten

In diesem Fall würde man zunächst mit anderen Verfahren zu jedem Kunden seine Merkmale ermitteln und zusammenstellen. Clustering und Klassifikation könnten dann auf diesem „Kundenobjekten“ arbeiten:

- Clustering von Kunden, zur Ermittlung von Kundengruppen. Die Gruppe (z. B. junger, lediger Privatkunde mit Vermögen) könnte dem Kunden als weiteres Merkmal zugeordnet werden und stünde für den CRM-Prozess zur Verfügung.
- Klassifikation von Kunden, zur Einordnung in Kategorien: Privatkunde/Geschäftskunde; Lebensphase; ansprechbar auf Versicherungen, Bausparverträge, Wertpapiere (ja/nein)

4.2 Besondere Anforderungen an die Verfahren

Eignung für große Datenmengen

Die mögliche Datenmenge bewegt sich in einem sehr breiten Spektrum: Für die Anzahl der Kunden sind Werte von 10.000 (regionale Kleinbank) bis zu einigen hunderttausend (führende Online-Broker) oder mehreren Millionen (Großbanken) möglich. Je nach Art des Kunden kann dieser pro Monat größenordnungsmäßig zwischen 10 Buchungen (kleiner Privatkunde) und 50.000 Buchungen (großer Geschäftskunde) durchführen. Somit ist die Skalierbarkeit des Analyseverfahrens ein wichtiges Auswahlkriterium. Dies gilt vor allen Dingen für Clusteringverfahren. Bei Klassifikationsverfahren könnte die aufwändige Trainingsphase möglicherweise bereits vor dem praktischen Einsatz erfolgen.

Sinnvolle Behandlung gemischt-skalierter Daten

Die Attribute der Ausgangsdaten weisen unterschiedliche Skalierungen auf. Am Beispiel eines Überweisungsvorgangs wird dies deutlich: Name, Kontonummer, Verwendungszweck, Währung: nominal skaliert. Bankleitzahl: je nach Betrachtungsweise ordinal skaliert oder nominal skaliert. Datum: intervallskaliert. Betrag: verhältnisskaliert. Die Namen von Kreditinstituten lassen sich anhand der Bankleitzahl eindeutig bestimmen, sie sind also redundant und sollten bei der Analyse nicht berücksichtigt werden.

Bei kategoriellen Daten ist zu beachten, dass sie unter Umständen zu einer hohen Dimensionalität des Merkmalsraums führen können (wenn sie z. B. binär codiert werden). Ursprünglich liegt jedoch keine Hochdimensionalität vor (10-20 Attribute pro Objekt/Zahlungsvorgang).

Verfügbarkeit von komfortablen Implementierungen

Spezielle Clustering- und Klassifikationsverfahren, die sich besonders gut für das Data Mining eignen, sind erst in den letzten Jahren entwickelt worden. Viele der leistungsfähigsten Algorithmen stehen daher zur Zeit nur als Fortschungsprototypen zur Verfügung. Eine Integration der Algorithmen in komfortable Analyse-Pakete mit einheitlicher Benutzeroberfläche sowie leistungsfähigen Datenimport- und -aufbereitungsfunktionen wäre jedoch

wünschenswert, um eine schnelle und unproblematische Erprobung verschiedener unterschiedlicher Ansätze in der Anfangsphase des Projektseminars zu ermöglichen.

4.3 Vorschläge zur weiteren Vorgehensweise

Bei der Anwendung sollten zunächst nur ausgewählte Teilmengen der Ausgangsdaten, d.h. der Zahlungsvorgänge, untersucht werden, mit dem Ziel, sich mit den grundsätzlichen Analyseschritten vertraut zu machen. Die Beschränkung auf kleine Datenmengen hält den Aufwand für die Datenvorbereitung klein, ermöglicht schnelle Durchläufe der Algorithmen und vereinfacht die Interpretation der Analyseergebnisse. Zu Anfang sollten Verfahren verwendet werden, die in integrierten Statistik- oder Data-Mining-Paketen (wie SPSS, Clementine oder IBM Intelligent Miner for Data) verfügbar sind, um nicht für jedes einzelne Verfahren Softwarebeschaffungs-, Installations- und Einarbeitungsaufwand zu haben. Von den Klassifikationsverfahren könnten dies Entscheidungsbaumverfahren wie C5.0 sein, für das Clustering partitionierende Verfahren wie z. B. k-medoids/PAM. Bezogen auf das Clustering sollte in dieser Phase bereits versucht werden, den Einfluss verschiedener Distanzmaße beim Clustering zu beurteilen und ein geeignetes Ähnlichkeitsmaß für Zahlungsvorgänge vorläufig festzulegen.

Später kann überprüft werden, welche der eingesetzten Verfahren auch auf großen Datenmengen ausreichend schnell arbeiten und verwertbare Ergebnisse erzielen. Gegebenenfalls müssen dann speziellere Verfahren (wie z. B. k-prototypes oder QUEST) hinzugezogen werden.

Literatur

[1] Backhaus, Klaus ; Erichson, Bernd ; Plinke, Wulff ; Weiber, Rolf: *Multivariate Analysemethoden: eine anwendungsorientierte Einführung.* 9., überarb. u. erw. Aufl. Berlin : Springer, 2000

[2] Berry, Michael J. A. ; Linoff, Gordon S.: *Mastering Data Mining: The Art and Science of Customer Relationship Management.* New York : John Wiley & Sons, 1999

[3] Bortz, Jürgen: *Statistik für Sozialwissenschaftler.* 5., vollst. überarb. Aufl. Berlin : Springer, 1999

[4] Fahrmeir, Ludwig ; Häussler, Walter ; Tutz, Gerhard: Diskriminanzanalyse. In: Fahrmeir, Ludwig (Hrsg.) ; Hamerle, Alfred (Hrsg.) ; Tutz, Gerhard (Hrsg.): *Multivariate statistische Verfahren.* Berlin : de Gruyter, 1996, S. 357–435

[5] Grimmer, Udo ; Mucha, Hans-Joachim: Datensegmentierung mittels Clusteranalyse. In: Nakhaeizadeh, Gholamreza (Hrsg.): *Data Mining. Theoretische Aspekte und Anwendungen.* Heidelberg : Physika-Verlag, 1998, S. 109–141

[6] Han, Jiawei ; Kamber, Micheline: *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000

[7] Hartung, Joachim ; Elpelt, Bärbel: *Multivariate Statistik. Lehr- und Handbuch der angewandten Statistik.* 5., durchges. Aufl. München, Wien : R. Oldenbourg Verlag, 1995

[8] Huang, Zhexue: Clustering large data sets with mixed numeric and categorical values. In: Lu, Hongjun (Hrsg.) ; Liu, Huan (Hrsg.) ; Motoda, Hiroshi (Hrsg.): *Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, World Scientific, 1997, S. 21–34. – URL `http://www.act.cmis.csiro.au/gjw/papers/apkdd.pdf`. – Zugriffsdatum: 2002-04-14

[9] Kaufmann, Heinz ; Pape, Heinz: Clusteranalyse. In: Fahrmeir, Ludwig (Hrsg.) ; Hamerle, Alfred (Hrsg.) ; Tutz, Gerhard (Hrsg.): *Multivariate statistische Verfahren.* Berlin : de Gruyter, 1996, S. 437–536

[10] Steinbach, Michael: *An introduction to Cluster Analysis for Data Mining.* – URL `http://www.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf`. – Zugriffsdatum: 2002-04-14

[11] Steinhausen, Detlef ; Langer, Klaus: *Clusteranalyse: Einführung in Methoden und Verfahren der automatischen Klassifikation.* Walter de Gruyter Verlag, 1977