Diplomarbeit

Häufigkeitsbasierte Merkmalsgenerierung für die Wissensentdeckung in Datenbanken

Hanna Köpcke

Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

Dienstag, 9. Dezember 2003

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inform. Martin Scholz

Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?
Where is the information we have lost in data?
Where is the data we have lost in libraries (computers, the Internet)?

T. S. Elliot

Danksagung

Ein herzliches Dankeschön an alle, die mir die vorliegende Arbeit ermöglichten.
Mein besonderer Dank gilt meinen beiden Betreuern Frau Prof. Dr. Katharina Morik und Herrn Dipl.-Inform. Martin Scholz. Ihre hilfreichen Anregungen, besonderers aber ihre konstruktive Kritik waren für mich sehr wertvoll.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Die Extraktion von verwertbarem Wissen aus Daten ist ein Thema, das angesichts der Menge des zur Zeit verfügbaren Datenmaterials mehr und mehr an Aktualität gewinnt. In vielen Unternehmen und wissenschaftlichen Institutionen existieren sehr große Datenbestände, deren Analyse nutzbare Erkenntnisse verspricht. Seit einigen Jahren werden unter dem Begriff *Wissensentdeckung in Datenbanken* Methoden und Systeme zur Entdeckung neuartigen, verborgenen Wissens aus großen Datenbeständen entwickelt. Data Mining-Verfahren verbinden Methoden aus den Bereichen Statistik, Maschinelles Lernen, Datenbanken und Visualisierung, um den Prozess des Auffindens verborgener und interessanter Informationen in großen Datenbeständen umfassend zu unterstützen. Ob die Wissensentdeckung auf einem Datensatz erfolgreich ist und tatsächlich zu neuen Erkenntnissen führt, hängt von vielen Faktoren ab. Dazu gehören unter anderem die Qualität und Vollständigkeit der Daten, die Klarheit der Aufgabe und die Verfügbarkeit effizienter Algorithmen. Während die Forschung nach effizienten Lernverfahren auf eine lange Tradition zurückblicken kann, finden die vorverarbeitenden Schritte erst seit kurzem Beachtung. Dabei kommt gerade ihnen im Prozess der Wissensentdeckung eine wichtige Rolle zu. Zur Vorverarbeitung zählt die Inspektion der Datencharakteristika (wie z.B. die Klassenverteilung), die Verbesserung der Qualität der Daten (z.B. die Behandlung von fehlenden Werten), die Auswahl eines Lernverfahrens und falls erforderlich die Transformation in das vom Lernverfahren benötigte Format. Zur Erzielung optimaler Ergebnisse durch das Lernverfahren ist eine geeignete Datenrepräsentation erforderlich. Durch Merkmalsauswahl und Merkmalsgenerierung kann ein zunächst ungeeignet repräsentierter Datensatz transformiert werden.

In dieser Diplomarbeit wird untersucht, wie TF/IDF zur häufigkeitsbasierten Merkmalsgenerierung eingesetzt werden kann. Die Anwendbarkeit wird an einem Datensatz der Versicherungsgesellschaft Swiss Life untersucht. Dieser Datensatz umfasst Kunden- und Versicherungsvertragsdaten in anonymisierter Form. Die Aufgabe besteht darin, den Rückkauf eines Vertrages vorherzusagen. Von einem Rückkauf spricht man, wenn der Versicherungsnehmer die Versicherung vorzeitig durch Rücktritt oder Kündigung beendet. Die Versicherungsgesellschaft ist in diesem Fall verpflichtet, dem Versicherungsnehmer den aktuellen Zeitwert der Versicherung auszuzahlen. Rückkaufe sind für die Versicherungsgesellschaft Verlustgeschäfte, da Rücklagen zu jedem Zeitpunkt bereitstehen müssen. Die frühzeitige Identifizierung von rückkaufgefährdeten Verträgen ermöglicht einer

Versicherungsgesellschaft die Entwicklung von Strategien, den Rückkauf zu verhindern.

1.1. Gliederung der Diplomarbeit

In Kapitel 2 wird eine Einführung in die Wissensentdeckung in Datenbanken (engl. Knowledge Discovery in Databases, KDD) gegeben. Der Begriff *Wissensentdeckung in Datenbanken* wird definiert und der KDD-Prozess vorgestellt. Die im Rahmen der Wissensentdeckung zu lösenden Lernaufgaben werden erklärt. Klassifikation ist eine mögliche Lernaufgabe. Es werden Verfahren vorgestellt, die diese Lernaufgabe lösen.

Der KDD-Prozess umfasst mehrere Phasen. Der Vorverarbeitung kommt in diesem Prozess eine besondere Bedeutung zu. Sie besteht darin, die zu analysierenden Daten für den anschließenden Data Minig-Schritt geeignet zu repräsentieren. Merkmalsauswahl und Merkmalsgenerierung transformieren die Daten in einen geeigneten Merkmalsraum. In Kapitel 3 werden Ansätze zur Merkmalsauswahl, zur Merkmalsgenerierung und zur Kombination beider Methoden vorgestellt. Danach wird TF/IDF zur häufigkeitsbasierten Merkmalsgenerierung vorgeschlagen. TF/IDF ist ein Maß zur Gewichtung von Wortern. Das Maß hat seinen Ursprung im Information Retrieval und wird im Maschinellen Lernen bei der Textklassifikation eingesetzt.

Kapitel 4 widmet sich der Beschreibung des Mining Mart Systems. Das System unterstützt den Prozess der Wissensentdeckung, insbesondere die Vorverarbeitung der Daten. Es folgt eine Schilderung der Realisierung von TF/IDF als Operator für dieses System.

Die Anwendbarkeit von TF/IDF zur Merkmalsgenerierung wird an einem Fallbeispiel untersucht. Es handelt sich um temporale Daten aus Versicherungsverträgen der Rentenanstalt Swiss Life. In Kapitel 5 wird zunächst die zu lösende Aufgabe erläutert. Danach werden die Daten detailliert beschrieben und Ergebnisse anderer Arbeiten zur Analyse der Swiss Life-Daten vorgestellt. Abschließend wird die Durchführung und das Ergebnis der Experimente geschildert.

In Kapitel 6 wird anhand des von Thorsten Joachims entwickelten statistischen Lernmodells für die Textklassifikation [Joachims 2002] das gute Lernergebnis der SVM auf den mit Hilfe von TF/IDF repräsentierten Daten erklärt. Joachims Theorie wird vorgestellt und auf die verwendeten Daten angewandt.

Abschließend werden in Kapitel 7 die Ergebnisse der Arbeit zusammengefasst und mögliche Erweiterungen des TF/IDF Ansatzes sowie neue Perspektiven diskutiert.

2. Wissensentdeckung in Datenbanken

In den verschiedensten informationsverarbeitenden Aufgabengebieten entstehen enorme Mengen von Daten, z. B. in der Wissenschaft oder der Wirtschaft. Viele dieser Daten sind jedoch nicht unmittelbar von Nutzen, da das eigentlich enthaltene Wissen von uninteressanten Daten „verdeckt“ wird, so dass Zusammenhänge oder Strukturen nicht direkt erkennbar sind. Diese Erkenntnis war die Motivation für die Entwicklung effizienter Konzepte zur automatischen Datenanalyse mit dem Ziel, das in den Daten verborgene implizite Wissen aufzufinden und explizit darzustellen. Viele dieser Verfahren werden seit Beginn der neunziger Jahre unter dem Begriff *Wissensentdeckung in Datenbanken* (engl. *Knowledge Discovery in Databases, KDD*) zusammengefaßt. Es handelt sich hierbei nicht nur um neue Methoden, im Gegenteil: Viele sind auch in den Gebieten Statistik, Mustererkennung, künstliche Intelligenz, maschinelles Lernen, Datenvisualisierung und Datenbanken bekannt. Neu ist vielmehr ihre konsequente Ausrichtung auf die effiziente Verarbeitung sehr großer Datenmengen.

In diesem Kapitel soll eine kurze Einführung in die Wissensentdeckung gegeben werden, indem zunächst der Begriff Wissensentdeckung in Datenbanken definiert wird und anschließend die grundlegenden Schritte im KDD-Prozess erläutert werden. Es folgt eine Vorstellung von Lernaufgaben, dabei wird insbesondere auf die Klassifikation eingangen. Es werden Verfahren vorgestellt, die diese Lernaufgabe lösen.

2.1. Begriffsbestimmung

Die in der Literatur allgemein anerkannte Festlegung des Begriffs *Wissensentdeckung in Datenbanken* stammt von Fayyad, Piatetsky-Shapiro und Smyth [Fayyad et al. 1996].

Definition 1 (Wissensentdeckung in Datenbanken (KDD))
Wissensentdeckung in Datenbanken ist der nichttriviale Prozess der Identifikation gültiger, neuer, potenziell nützlicher und schlussendlich verständlicher Muster in Daten.

Die Daten D bestehen dabei im Allgemeinen aus einer Menge von Objekten, z.B. Tupel einer Tabelle in einer Datenbank, die durch bestimmte Merkmale oder Variablen charakterisiert sind. Die gesuchten Muster P sind Ausdrücke in einer Hypothesensprache $\mathcal{L}_H$, die eine Teilmenge $D_p \subseteq D$ abdecken. Sie sollen bestimmten Eigenschaften genügen. Zum einen sollen sie gültig sein, d.h. die zugrundeliegenden Daten und darin enthaltenen Zusammenhänge und Phänomene zutreffend beschreiben. Zum anderen werden Neuartigkeit, Nützlichkeit und Verständlichkeit der identifizierten Muster gefordert, wobei diese Eigenschaften formal nur schwer zu erfassen sind.

2.2. Der KDD-Prozess

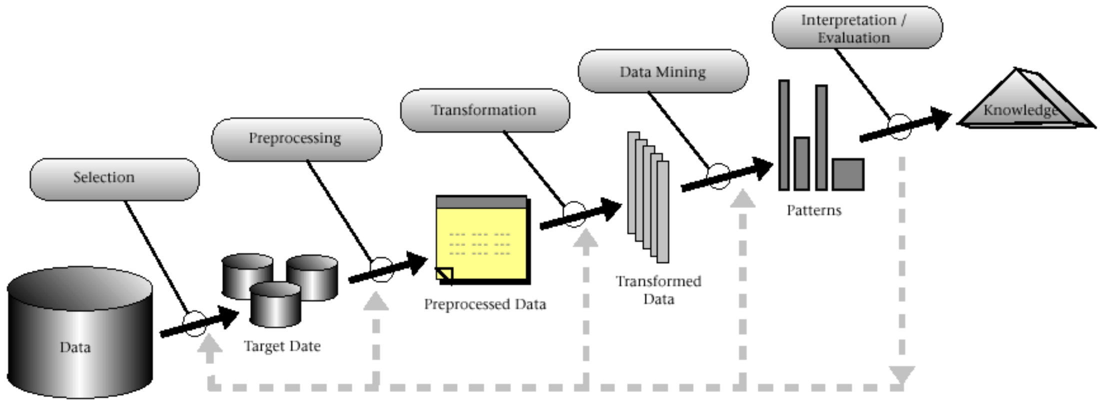


Abbildung 2.1.: Der KDD-Prozes nach [Fayyad et al. 1996]

Die Betonung des Prozessaspektes in der Definition verweist auf die umfassende Sichtweise, die KDD auf den Prozess der Datenanalyse hat: es werden alle Schritte von der ersten Beschäftigung mit einer Domäne bis hin zur Verwendung der Ergebnisse in Reports oder installierten Softwaresystemen betrachtet. Abbildung 2.1 zeigt die grundlegenden Schritte dieses Prozesses (nach [Fayyad et al. 1996]).

- **Auswahl eines Zieldatensatzes**
  Häufig sind die benötigten Datenquellen verteilt; insbesondere im kommerziellen Bereich geschieht die Datenerhebung und -speicherung an unterschiedlichen Stellen des betrieblichen Transformationsprozesses. In solchen Fällen muss eine geeignete Datenselektion und -zusammenführung stattfinden.

- **Vorverarbeitung**
  Diesem Schritt kommt in bezug auf Anwendbarkeit und Effizienz des anschließenden Data-Mining-Schritts eine besondere Bedeutung zu. Gerade bei großen Daten-

banken tritt oft das Problem auf, dass fehlende, mehrdeutige oder widersprüchliche Informationen vorhanden sind. Diese Datenbestände müssen dann im Hinblick auf ihre Qualität überarbeitet werden, indem z. B. statistische Ausreißer oder inkonsistente Datensätze entfernt oder fehlende Merkmalswerte ergänzt werden.

- **Transformation**
  Aufgabe dieses Schrittes ist es, geeignete Merkmale für die Repräsentation der Daten im Hinblick auf die Zielsetzung zu finden. Hierzu kann z. B. innerhalb der Datenbasis die Zusammenfassung mehrerer Merkmale, die Ermittlung und Eliminierung überflüssiger Merkmale, die Diskretisierung und Gruppierung von Merkmalswerten oder eine auf den zu verwendenden Algorithmus abgestimmte Transformation des Eingaberaums gehören.

- **Data Mining**
  In diesem Schritt geschieht die eigentliche Wissensentdeckung durch Anwendung eines Data Mining-Verfahren auf den aufbereiteten Datensatz.

- **Interpretation der Ergebnisse**
  Nach dem Data Mining erfolgt die Auswertung und Interpretation der Ergebnisse. Der ganze Wissensentdeckungsprozess wird, wenn nötig, mehrmals wiederholt. Am Ende werden die Ergebnisse in einem Bericht zusammengefasst und evtl. in einem Computerprogramm oder auf eine andere Art und Weise weiterverwendet.

Der Wissensentdeckungsprozess ist iterativ und interaktiv. Die einzelnen Schritte werden in der Regel nicht in linearer Abfolge durchlaufen, sondern es können sich Schleifen bzw. Rücksprünge ergeben. So kommen zum Beispiel Situationen vor, bei denen zu Beginn der Datenerforschung die verfolgten Ziele noch nicht exakt festgelegt werden können oder bei denen sich während des Prozesses vorher nicht bedachte interessante Unterziele ergeben. Auch kann die Art der entdeckten Muster überraschen und das Augenmerk in eine neue Richtung lenken. Durch Interaktion mit dem Benutzer müssen insbesondere die Datenaufbereitungs- und analysearbeiten auf den jeweiligen Anwendungszweck abgestimmt werden.

2.3. Lernaufgaben und Lernverfahren

Im vorangegangenen Abschnitt wurde der Prozess der Wissensentdeckung in Datenbanken vorgestellt. Data Mining ist ein Schritt dieses Prozesses. In diesem Schritt werden Data Mining Verfahren zur Lösung von Lernaufgaben eingesetzt. In [Wrobel et al. 2000] wird definiert, was unter einer Lernaufgabe zu verstehen ist.

Definition 2 (Lernaufgabe)
Eine Lernaufgabe wird definiert durch

- *Eine Beschreibung der dem lernenden System zur Verfügung stehenden Eingaben.*
- *Die vom lernenden System erwarteten Ausgaben.*
- *Den Randbedingungen des Lernsystems selbst (z.B. maximale Laufzeiten oder Speicherverbrauch)*

Die Lernaufgabe wird genau dann erfolgreich vom System gelöst, wenn das System in der Lage ist, bei Eingaben, die den Spezifikationen entsprechen, unter den geforderten Randbedingungen, Ausgaben mit den gewünschten Eigenschaften zu erzeugen.

Grundsätzlich werden prädiktive und deskriptive Lernaufgaben unterschieden. Bei prädiktiven Lernaufgaben ist es das Ziel, eine unbekannte Funktion möglichst gut zu approximieren, um so für alle möglichen zukünftigen Instanzen aus dem Instanzenraum den Funktionswert möglichst gut vorhersagen zu können. Hierzu ist ein globales Modell erforderlich, das es gestattet, für jede mögliche Instanz auch tatsächliche eine Vorhersage zu treffen. Das Ziel einer deskriptiven Lernaufgabe besteht hingegen darin, durch Hypothesen beschriebene Teilbereiche des Instanzenraumes zu identifizieren, über die lokal interssante Aussagen gemacht werden können.

Die Klassifikation gehört zu den prädiktiven Lernaufgaben. Bei der Klassifikation geht es darum, eine Menge von Objekten in endlich viele disjunkte Klassen $K_1, K_2, \ldots, K_m$ einzuteilen. Die Objekte werden durch eine Menge von Attributen $A_1, A_2, \ldots, A_n$ beschrieben. Formal besteht das Ziel darin, eine Funktion f mit

$$f : A_1 \times A_2 \times \ldots \times A_n \rightarrow \{K_1, K_2, \ldots, K_m\} \tag{2.1}$$

zu bestimmen.

Es gibt eine Reihe von Verfahren, die Klassifikationsaufgaben lösen können. Im Rahmen dieser Arbeit kommen die Stützvektorvektormethode, Naive Bayes, Entscheidungsbäume und Apriori zum Einsatz. Diese Verfahren werden nachfolgend detailliert vorgestellt.

2.3.1. Die Stützvektormethode

Die Stützvektormethode (Support Vector Machine, SVM) ist ein Lernverfahren, das aus der statistischen Lerntheorie hervorgegangen ist [Vapnik 1998]. Der Algorithmus beruht auf der Idee der strukturellen Risikominimierung [Vapnik 1982]. Die folgende Darstellung orientiert sich an [Joachims 2001, Wrobel et al. 2000]. Für eine detaillierte Diskussion der Theorie der strukturellen Risikominimierung und eine ausführlichere Darstellung nicht-linearer Erweiterungen wird auf [Burges 1998, Vapnik 1998, Christianini und Shawe-Taylor 2000] verwiesen.

Die Eingabe der SVM besteht aus Trainingsbeispielen E

$$(\vec{x_1}, y_1), \ldots, (\vec{x_l}, y_l) \quad \text{mit} \quad (\vec{x_i}, y_i) \in \mathbb{R}^p \times \{+1, -1\} \tag{2.2}$$

Ein einzelnes Beispiel $(\overrightarrow{x_i}, y_i)$ ist durch einen sog. *Merkmalsvektor* $\overrightarrow{x_i}$ und einer *Kennzeichnung* (oder *Klassifikation*) y_i beschrieben.

Das Ziel des Algorithmus' ist es, aus den gegebenen Beispielen eine Klassifikationsregel abzuleiten, die neue Beispiele gleichen Typs mit größtmöglicher Wahrscheinlichkeit korrekt klassifiziert. Lineare Klassifikationsregeln h können durch folgende Formel dargestellt werden:

$$h(\vec{x}) = \text{sign}\left(b + \sum_{i=1}^{n} w_i x_i\right) = \text{sign}(\vec{w} \cdot \vec{x} + b) \tag{2.3}$$

Die Klassifikationsregel besteht aus zwei Komponenten $\vec{w}$ und b. $\vec{w}$ ist der Gewichtsvektor, er ordnet jedem Merkmal ein gewisses Gewicht zu. b ist ein Schwellwert. In der Trainingsphase legt die SVM den Gewichtsvektor $\vec{w}$ und den Schwellwert b fest. Nach dieser Festlegung kann ein unbekanntes Beispiel klassifiziert werden, indem $h(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) = y$ durch Bildung des Skalarproduktes $\vec{w} \cdot \vec{x}$, der Addition von b und der Anwendung der Funktion $\text{sign} : \mathbb{R} \to \{+1, -1\}$ berechnet wird. Die Funktion $\text{sign}(a)$ hat den Wert $+1$, wenn $a > 0$ ist, sonst nimmt sie den Wert -1 an.

Die Gleichung $\text{sign}(\vec{w} \cdot \vec{x} + b) = 0$ beschreibt eine sogenannte Hyperebene. Im zweidimensionalen Raum ist die Hyperebene eine Gerade. Alle Beispiele auf der einen Seite der Hyperebene werden als positive, alle Merkmalsvektoren auf der anderen Seite werden als negative Beispiele klassifiziert. Wie im linken Teil der Abbildung 2.2 zu sehen, gibt es viele Geraden, die positive und negative Beispiele trennen. Die SVM wählt die Hyperebene, welche die positiven und negativen Beispiele mit maximalen Abstand trennt. Konkret bedeutet dies, dass der euklidische Abstand der Merkmalsvektoren $\vec{x_i}$ zur Hyperebene maximiert werden muss. Zu jedem Beispiel muss die Hyperebene mindestens einen Abstand von δ haben. Formal kann man das durch das folgende Optimierungsproblem beschreiben:

$$\begin{aligned} \text{Berechne} \quad & \vec{w}, b && (2.4) \\ \text{so dass} \quad & \delta \text{ maximal} && (2.5) \\ \text{und es gilt} \quad & y_1 \frac{1}{||\vec{w}||} [\vec{w} \cdot \vec{x_1} + b] \geq \delta, \ldots, y_\ell \frac{1}{||\vec{w}||} [\vec{w} \cdot \vec{x_\ell} + b] \geq \delta && (2.6) \end{aligned}$$

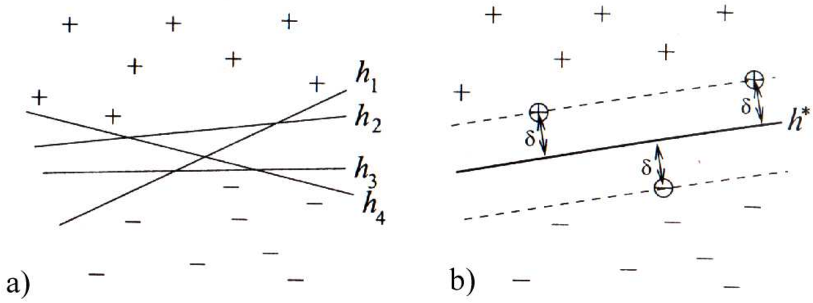


Abbildung 2.2.: a) einige der möglichen Trennebenen, b) optimale Hyperebene, eingekreiste Stützvektoren (aus [WROBEL et al. 2000])

Die Beispiele, die der Hyperebene am nächsten liegen, werden Stützvektoren (engl. Support Vectors) genannt. Ihr Abstand ist genau δ, d.h. die zugehörige Ungleichung ist per Gleichheit erfüllt.

Um die optimale Hyperebene zu berechnen, wird das Optimierungsproblem in eine einfachere Form gebracht. Die Länge des Gewichtsvektors $||\vec{w}||$ beeinflusst die Lage der Hyperebene nicht. Diese Tatsache erlaubt es, die Hyperebene zu fixieren, indem $\delta = \frac{1}{||\vec{w}||}$ gesetzt wird. Alle Vorkommen von δ werden substituiert. Es ergibt sich dann das folgende Optimierungsproblem:

$$\text{finde} \quad \vec{w}, b \tag{2.7}$$

$$\text{so dass} \quad \vec{w} \cdot \vec{w} \text{ minimal} \tag{2.8}$$

$$\text{und es gilt} \quad y_1 [\vec{w} \cdot \vec{x_1} + b] \geq 1, \ldots, y_\ell [\vec{w} \cdot \vec{x_\ell} + b] \geq 1 \tag{2.9}$$

Dieses quadratische Optimierungsproblem läßt sich schlecht direkt mit numerischen Methoden lösen. Deshalb wird es in das folgende, äquivalente Optimierungsproblem umgeformt.

$$\text{finde} \quad \alpha_1, \ldots, \alpha_\ell \tag{2.10}$$

$$\text{so dass} \quad -\sum_{i=1}^{\ell} \alpha_i + \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} y_i y_j \alpha_i \alpha_j (\vec{x_i} \cdot \vec{x_j}) \text{ minimal} \tag{2.11}$$

$$\text{und es gilt} \quad \sum_{i=1}^{\ell} y_i \alpha_i = 0 \text{ und } \forall i : 0 \leq \alpha_i \tag{2.12}$$

Es gilt eine Belegung für die Variablen $\alpha_1, \ldots, \alpha_\ell$ zu finden, so dass der Ausdruck in 2.11 minimal ist. Dabei müssen alle α_i größer oder gleich null sein. Gleichzeitig muss die Gleichung 2.12 erfüllt sein. Nach der Berechnung der Lösung $\alpha_1^*, \ldots, \alpha_\ell^*$ des Optimierungsproblems ergibt sich folgende Klassifikationsregel:

$$h(x) = \text{sign}\left(b + \sum_{i=0}^{\ell} \alpha_i^* y_i \vec{x_i} \cdot \vec{x}\right) \tag{2.13}$$

Der Gewichtsvektor $\vec{w}$ wird berechnet durch

$$\vec{w} = \sum_{i=0}^{\ell} \alpha_i^* y_i \vec{x_i} \tag{2.14}$$

wobei nur die Stützvektoren mit $\alpha_i^* > 0$ in die Berechnung eingehen. Der Schwellwert $b = y_i - \vec{w} \cdot \vec{x_i}$ lässt sich mit Hilfe eines beliebigen Stützvektors durch Einsetzen bestimmen.

SVMs mit weicher Trennung

Manchmal ist es nicht möglich, die Trainingsbeispiele so zu trennen, dass alle Beispiele korrekt klassifiziert werden. Die Lösung besteht darin, Trainingsfehler zuzulassen. Man versucht nicht nur den Abstand zur optimalen Hyperebene, sondern auch die Summe der in Kauf genommenen Fehler ξ_i zu minimieren. Dies wird als *weiche Trennung* bezeichnet.

$$\begin{aligned}
\text{Berechne} \quad & \vec{w}, b && (2.15)\\
\text{so dass} \quad & w * w + C \sum_{i=1}^{l} \xi_i \text{ minimal} && (2.16)\\
\text{und es gilt} \quad & y_1[\vec{w} * \vec{x_1} + b] \geq 1 - \xi_1, \ldots, y_l[\vec{w} * \vec{x_l} + b] \geq 1 - \xi_l && (2.17)
\end{aligned}$$

Jedes ξ_i misst, wie weit das entsprechende Beispiel im oder jenseits des Separationsbereichs liegt. C ist ein Parameter, der ein Abwägen von Separationsweite und Fehler erlaubt. Das duale quadratische Optimierungsproblem in Standardform für SVMs mit weicher Trennung gleicht dem für SVMs mit harter Trennung, wenn man in Gleichung 2.12 $\forall : 0 \leq \alpha_i$ durch $\forall : 0 \leq \alpha_i \leq C$ ersetzt.

2.3.2. Naive Bayes

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. $\Pr(C_i|X)$ gibt 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 maximal ist. Die Bayes'sche Regel [James 1985] besagt, dass ein Klassifikator optimal ist, wenn er ein Objekt mit dem Merkmalsvektor X der Klasse zuweist, für die $\Pr(C_i|X)$ maximal ist.

$$H_{BAYES}(X) = \underset{C_i \in C}{\operatorname{argmax}} \Pr(C_i|X) \tag{2.18}$$

Bayes' Theorem [James 1985] kann dazu benutzt werden, die Wahrscheinlichkeit $\Pr(C_i|X)$ in zwei Teile aufzuspalten.

$$\Pr(C_i|X) = \frac{\Pr(X|C_i) * \Pr(C_i)}{\sum_{C_i \in C} \Pr(X|C_i) * \Pr(C_i)} \tag{2.19}$$

$\Pr(C_i)$ ist die a priori Wahrscheinlichkeit, dass ein Objekt in Klasse C_i ist. $\Pr(X|C_i)$ ist die Wahrscheinlichkeit, dass in Klasse C_i das Objekt mit dem Merkmalsvektor X beobachtet wird.

$\hat{\Pr}(C_i)$, der Schätzwert für $\Pr(C_i)$, kann von dem Anteil der Trainingsbeispiele berechnet werden, die sich in der entsprechenden Klasse befinden.

$$\hat{\Pr}(C_i) = \frac{|C_i|}{\sum_{C' \in C} |C'|} \tag{2.20}$$

Dabei ist $|C_i|$ die Anzahl der Trainingsbeispiele in Klasse C_i.

Das Schätzen der Wahrscheinlichkeit $\Pr(X|C_i)$ ist weniger einfach. Einen Schätzwert zu bestimmen, wird erst durch die vereinfachende Annahme der konditionalen Unabhängigkeit möglich. Diese besagt, dass Merkmale unabhängig sind, gegeben die Klasse, aus der ein Objekt stammt. Unter dieser Annahme kann $\Pr(X|C_i)$ als Produkt aus den bedingten Wahrscheinlichkeiten für die in X vorkommenden Merkmale x_j berechnet werden:

$$\Pr(X|C_i) = \prod_{j=1}^{|C_i|} \Pr(x_j|C_i) \tag{2.21}$$

2.3.3. Der Entscheidungsbaumlerner C4.5

Das induktive Lernsystem C4.5 [Quinlan 1993] wurde auf der Basis von ID3 [Quinlan 1986] entwickelt. Wie ID3 generiert auch C4.5 Klassifikatoren in Form von Entscheidungsbäumen. Ein Entscheidungsbaum ist ein gerichteter, azyklischer Graph. Jeder Knoten des Graphen ist entweder ein Entscheidungsknoten oder ein Blatt. Jedes Blatt ist mit einer Klasse beschriftet. An einem Entscheidungsknoten wird ein Attribut-Wert-Test über ein einzelnes Attribut ausgeführt. Für jeden Ausgang dieses Attributtests hat der Entscheidungsknoten einen Unterbaum, bei dem es sich wiederum um einen Entscheidungsbaum handelt.

Die Konstruktion eines Entscheidungsbaumes erfolgt bei C4.5 über eine rekursive „Teile-und-Herrsche“ -Strategie. Seien $C = \{C_1, C_2, \dots\}$ die Menge der Klassen und T eine Menge von Trainingsbeispielen. Es gibt in einem Schritt während der Konstruktion eines Entscheidungsbaumes drei mögliche Situationen:

- T enthält ein oder mehrere Beispiele, die alle zu einer Klasse C_j gehören:
  Der Entscheidungsbaum für T ist ein Blatt, das die Klasse C_j vorhersagt.
- T ist leer, enthält also keine Beispiele;
  Der Entscheidungsbaum ist ein Blatt, das die häufigste Klasse in seinem Elternknoten vorhersagt.
- T enthält Beispiele aus verschiedenen Klassen:
  Gemäß des Gain-Ratio-Kriterium wird der beste Attributtest ausgewählt. Wenn dieser Test die disjunkten Ausgänge $\{O_1, O_2, \dots, O_n\}$ hat, wird T so in disjunkte Teilmengen $T_1, T_2, \dots, T_n$ partitioniert, dass Teilmenge T_i alle Beispiele aus T enthält, die bei dem gewählten Attributtest den Ausgang O_i haben. Der Entscheidungsbaum für T ist dann ein Entscheidungsknoten, an dem der ausgewählte Test auszuführen ist und der für jeden Ausgang des Tests genau einen Unterbaum hat. Diese Prozedur wird rekursiv auf jede der Teilmengen von T angewandt, so dass der i-te Zweig des konstruierten Entscheidungsknotens zu einem Entscheidungsbaum für die Teilmenge T_i führt.

Zur Auswahl des jeweils besten Attributes an einem bestimmten Knoten verwendet C4.5 das Maß *GainRatio*. Gemäß der Informationstheorie gilt, dass die von einer Nachricht übertragene Information von der Wahrscheinlichkeit des Auftretens dieser Nachricht $\Pr(m)$ abhängt und in Bits gemessen werden kann:

$$-\log_2 \Pr(m) \quad \text{bits} \tag{2.22}$$

Sei X ein möglicher Attributtest mit n Ausgängen, der die Trainingsmenge T in die Teilmengen $T_1, T_2, \dots, T_n$ partitioniert. Sei $freq(C_j, T)$ die Anzahl der Trainingsbeispiele in T, die zur Klasse C_j gehören und sei $|T|$ die Anzahl der Beispiele in T. Die Wahrscheinlichkeit, dass ein zufällig ausgewähltes Beispiel in die Klasse C_j gehört beträgt:

$$\frac{freq(C_j, T)}{|T|} \tag{2.23}$$

Diese Nachricht liefert eine Information von:

$$-\log_2 \frac{freq(C_j, T)}{|T|} \quad \text{bits} \tag{2.24}$$

Um die erwartete Information einer solchen Nachricht über die Klassenzugehörigkeit eines zufällig aus der Menge T gewählten Beispiels zu berechnen, bildet man die über die Klassenhäufigkeiten gewichtete Summe der Informationsmengen von Nachrichten über die Zugehörigkeit zu einzelnen Klassen:

$$Info(T) = -\sum_{C_j \in C} \frac{freq(C_j, T)}{|T|} \cdot \log_2 \frac{freq(C_j, T)}{|T|} \quad \text{bits} \tag{2.25}$$

Diese Informationsmenge wird auch als *Entropie* der Menge T bezeichnet. Nachdem die Beispielmenge T entsprechend der n Ausgänge des Attributtests C partitioniert worden ist, ergibt sich die erwartungsgemäß benötigte Information wie folgt:

$$Info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot Info(T_i) \tag{2.26}$$

Die durch die Partitionierung von T durch den Attributtest gewonnene Information ist wie folgt messbar:

$$IG(X) = Info(T) - Info_X(T) \tag{2.27}$$

Dieses Maß erfasst den Informationsgewinn, der durch die Verwendung von X erzielt wird. Es wird als *Information Gain* bezeichnet. Information Gain bevorzugt Attributtests mit sehr vielen Ausgängen. In Extremfall, wenn ein Attribut die Menge T so partitioniert, dass jedes einzelne Beispiel aus T eine eigene Teilmenge bildet, ist der Informationsgewinn gemäß dem Information Gain maximal. Um diesen Fall zu vermeiden, wird das Maß $SplitInfo(T)$ eingeführt. Es misst die Information, die durch Partitionierung er Menge T in n Teilmengen gemäß X entsteht:

$$SplitInfo(T) = -\sum_{C_j \in C} \frac{|T_i|}{|T|} \cdot \log_2 \frac{|T_i|}{|T|} \quad \text{bits} \tag{2.28}$$

Das Maß Gain Ration berechnet den Informationsgewinn eines Attributes als Verhältnis zwischen dem Information Gain und der SplitInfo:

$$GainRatio(X) = \frac{IG(X)}{SplitInfo(X)} \tag{2.29}$$

2.3.4. Apriori zur Klassifikation

Das Entdecken von Assoziationsregeln ist eine der bekanntesten deskriptiven Lernaufgaben. Die Lernaufgabe kann folgendermaßen definiert werden [Wrobel et al. 2000]: Sei $\mathcal{I}$ eine Menge von Objekten (items) und $\mathcal{T} = \{t \mid t \subseteq \mathcal{I}\}$ eine Menge von Transaktionen. Sei weiterhin $supp_{min} \in [0,1]$ eine benutzerdefinierte Minimalhäufigkeit und $conf_{min} \in [0,1]$ eine benutzerdefinierte Minimalkonfidenz. Dann sind alle Regeln der Form $X \to Y$ mit $X \subseteq \mathcal{I}$, $Y \subseteq \mathcal{I}$ und $X \cap Y = \emptyset$ zu finden, für die gilt:

$$supp(X \to Y) := \frac{|\{t \in \mathcal{T} \mid X \cup Y \subseteq t\}|}{|\mathcal{T}|} \geq supp_{min} \tag{2.30}$$

und

$$conf(X \to Y) := \frac{|\{t \in \mathcal{T} \mid X \cup Y \subseteq t\}|}{|\{t \in \mathcal{T} \mid X \subseteq t\}|} \geq conf_{min} \tag{2.31}$$

Einer der bekanntesten Algorithmen zum Bestimmen von Assoziationsregeln ist der Apriori Algorithmus (siehe [Agrawal et al. 1993], [Agrawal und Srikant 1994b], [Agrawal und Srikant 1994a] und [Agrawal et al. 1996]).

In [Liu et al. 1998] wird ein Algorithmus präsentiert, der den Apriori Algorithmus zur Erzeugung von Klassifikationsregeln benutzt. Der vorgeschlagene Algorithmus heißt CBA (Classification Based on Associations). Er besteht aus zwei Teilen, einem Regelgenerator, der auf dem Apriori Algorithmus beruht und einem Algorithmus, der aus den erzeugten Regeln einen Klassifikator bildet. Um einen Klassifikator zu erzeugen, wird zunächst eine totale Ordnung auf der Menge der generierten Regeln gebildet.

Definition 3
Eine Regel r_i hat einen höheren Rang als eine Regel r_j, $r_i \succ r_j$, wenn

1. *die Konfidenz von r_i größer ist als von r_j oder*
2. *beide Regeln die gleiche Konfidenz haben, aber der Support von r_i größer ist als von r_j oder*
3. *beide Regeln die gleiche Konfidenz und den gleichen Support haben, aber r_i vor r_j erzeugt wird.*

Mit R die Menge der generierten Regeln und mit D die Menge der Trainingsdaten bezeichnet. Es sollen diejenigen Regeln aus R gewählt werden, die D überdecken. Der resultierende Klassifikator hat folgendes Format:

$$< r_1, r_2, \ldots, r_n, default\_class > \tag{2.32}$$

Dabei gilt $r_i \in R, r_a \succ r_b$ falls $b > a$. $default\_class$ ist die Standardklasse. Ein ungesehenes Beispiel ist klassifiziert durch die erste Regel, die das Beispiel erfüllt. Wenn es keine Regel gibt, die auf das Beispiel zutrifft, wird die Standardklasse gewählt. In [Liu et al. 1998] wird folgender naiver Algorithmus vorgestellt, der in drei Schritten vorgeht:

- Schritt 1: Die Menge der generierten Regeln R wird gemäß der Relation $\succ$ geordnet. Dadurch wird sichergestellt, dass stets die Regel mit dem höchsten Rang gewählt wird.
- Schritt 2: Für jede Regel r, wird Menge der Trainingsbeispiele D durchgegangen, um alle Beispiele zu finden, die von r überdeckt werden. Die Regel r wird markiert, wenn sie ein Trainingsbeispiel d richtig klassifiziert. Alle Trainingsbeispiele, die von r überdeckt werden, werden aus D entfernt.
- Schritt 3: Alle Regeln in C, die die Genauigkeit des Klassifikators nicht verbessern, werden verworfen. Als Schnittpunkt wird die erste Regel gewählt, die die geringste Anzahl an Fehlern aufweist. Alle Regeln, die auf diese Regel folgen, können verworfen werden, weil sie nur mehr Fehler produzieren. Die Regeln, die nicht verworfen wurden und die Standardklasse der letzten Regeln in C bilden den Klassifikator.

Dieser Algorithmus ist einfach, jedoch ineffizient. Insbesondere wenn die Daten nicht in den Hauptspeicher passen. Der Algorithmus muss die Menge der Trainingsbeispiele mehrmals durchlaufen. In [Liu et al. 1998] wird deshalb zusätzlich eine verbesserte Version des Algorithmus vorgestellt. Sie benötigt maximal zwei Durchgänge durch die Menge der Trainingsbeispiele.

3. Merkmalserzeugung für die Wissensentdeckung

Die Repräsentation von Beispielen und Hypothesen hat einen entscheidenden Einfluss auf die Generalisierungsfähigkeit von Lernalgorithmen. Die von einem Lernalgorithmus lernbaren Hypothesen werden zum einen durch den Algorithmus bestimmt, zum anderen aber auch ganz wesentlich von der Repräsentation eines Datensatzes beeinflusst. Ein Datensatz muss für die Wissensentdeckung in eine Repräsentation transformiert werden, die sich sowohl für den Lernalgorithmus als auch für die zu lösende Lernaufgabe eignet.

Bevor in diesem Kapitel TF/IDF zur häufigkeisbasierten Merkmalsgenerierung vorstellt wird, werden existierende Ansätze zur Merkmalsauswahl, zur Merkmalsgenerierung und zur Kombination beider Methoden beschrieben (siehe auch [Liu und Motoda 1998]).

3.1. Merkmalsauswahl

Das Hauptziel der *Merkmalsauswahl*, auch *Merkmalsselektion* genannt, ist eine *Dimensionsreduktion* des Repräsentationsraumes möglichst ohne Verlust wichtiger Information. Durch die Auswahl eines guten Attributsatzes kann oftmals die Qualität des Lernergebnisses optimiert werden. Ein guter Attributsatz zeichnet sich dadurch aus, dass er zum einen genug Information zum Lösen des Lernproblems bereitstellt, zum anderen aber keine irrelevanten Attribute enthält.

Viele Algorithmen fassen das Problem der Merkmalsauswahl als Suche auf. Eine Möglichkeit zur Kategorisierung dieser Algorithmen zur Merkmalsauswahl ist die Einteilung nach den beiden Dimensionen Suchstrategie und Evaluationsstrategie, wie sie in [Blum und Langley 1997] vorgeschlagen wird.

Die einfachste Suchstrategie ist die *erschöpfende Suche*. Diese Strategie garantiert, dass die beste Merkmalsmenge gefunden wird. Sie benötigt jedoch exponentielle Laufzeit in bezug auf die Anzahl der Merkmale und ist daher in den meisten Fällen nicht anwendbar.

Heuristische Suchmethoden verwenden eine Evaluierungsfunktion für die Suche. Eine Unterklasse dieser Algorithmen bilden die so genannten *hill climbing* Methoden. Diese Methoden wählen inkrementell die Untermengen der Merkmale aus, die in einer Iteration die Qualität des Lernergebnisses maximal verbessert. Zwei Instanzen der hill climbing Methoden sind *forward selection* und *backward elimination* [Aha und L. 1996].

Forward selection beginnt mit einer leeren Menge von Merkmalen. Der Algorithmus fügt iterativ Merkmale der Menge hinzu. Es wird jeweils das Merkmal ausgewählt, das die Qualität des Lernergebnisses maximal verbessert. Der Algorithmus stoppt, wenn die Qualität nicht weiter verbessert werden kann.

Ausgangspunkt bei der *backward elimination* ist die gesamte Merkmalsmenge. Bei dieser Strategie werden iterativ die Merkmale entfernt, deren Entfernung eine maximale Verbesserung der Qualität des Lernergebnisses bewirkt.

Ein großer Nachteil dieser sequentiellen *hill climbing* Methoden ist ihr Hang in lokalen Maxima stecken zu bleiben. Die Ursache liegt in der mangelnden Berücksichtung von Merkmalsinteraktion. Mit Merkmalsinteraktion wird die Situation bezeichnet, bei der die Auswirkung eines bestimmten Merkmals von Werten anderer Merkmale abhängt [Freitas 2001, Vafaie und Jong 1993]. Eine probabilistische Suchmethode, die mit Merkmalsinteraktionen umgehen kann, sind die *genetischen Algorithmen* [T. Baeck und Michalewicz 2000]. Hybride Ansätze werden z.B. in [Bala et al. 1995], [Punch et al. 1993] und [Yang und Honavar 1998] beschrieben. Die Ansätze verwenden evolutionäre Methoden für die Merkmalsauswahl. Zur Evaluation der Merkmalsmengen kommen unterschiedliche Lernmethoden zum Einsatz, z.B. neuronale Netze und Entscheidungsbäume.

3.2. Merkmalsgenerierung

Im Gegensatz zur Auswahl von Merkmalen wird bei ihrer Generierung der Merkmalsraum mit zusätzlichen Merkmalen angereichert, die aus den vorhandenen konstruiert oder abgeleitet werden. Ein Spezialfall der Merkmalsgenerierung ist die konstruktive Induktion (constructive induction) [Michalski 1983]. Bei dieser Methode wird induktive Generalisierung zur Erzeugung neuer Merkmale verwendet.

Konjunktion, Disjunktion und Negation werden für nominale Merkmalswerte als Operatoren verwendet, um neue Merkmale zu erzeugen. M-aus-N und X-aus-N sind weitere Möglichkeiten, um aus vorhandenen Merkmalen neue zu generieren. Die folgende Definition von M-aus-N und X-aus-N stammt aus [Zheng 1998] .

Definition 4
Sei $\{F_i \mid 1 \leq i \leq MaxFea\}$ eine Menge von Merkmalen aus einer Domäne und sei für jedes F_i $\{V_{ij} \mid 1 \leq j \leq MaxFeaVal_i\}$ seine Wertemenge, wobei $MaxFea$ die Anzahl der Merkmale ist und $MaxFeaVal_i$ die Anzahl der verschiedenen Werte für F_i ist.

- *X-aus-N besteht aus einer Menge von Paaren. Die Paare setzen sich zusammen aus einem Merkmal und einem Merkmalswert. X-aus-N wird notiert als:*

$$\begin{aligned} X\text{-}aus\text{-}\{\, FV_k \quad | \quad & FV_k \textit{ ist ein Paar aus einem Merkmal und einem Wert} \\ & \textit{notiert als } F_i = V_{ij}, \\ & 1 \leq k \leq N_+, N \leq N_+, 1 \leq j \leq MaxFea\} \end{aligned}$$

  Eine X-aus-N Darstellung kann als Wert jede Zahl zwischen 0 und N annehmen. Der Wert ist x, wenn x der FV_k wahr sind.

- *M-aus-N besteht aus einer Menge von Paaren, die sich aus einem Merkmal und einem Merkmalswert zusammensetzen, und einer Zahl M. M-aus-N wird notiert als:*

$$\begin{aligned} M\text{-}aus\text{-}\{\, FV_k \quad | \quad & FV_k \textit{ ist ein Paar aus einem Merkmal und einem Wert} \\ & \textit{notiert als } F_i = V_{ij}, \\ & 1 \leq k \leq N_+, N \leq N_+, 1 \leq j \leq MaxFea, 1 \leq M \leq N\} \end{aligned}$$

  Eine M-aus-N Darstellung ist entweder wahr oder falsch. Sie ist wahr, wenn mindestens M der FV_k wahr sind, ansonsten ist sie falsch.

N_+ ist die Anzahl der Paare aus Merkmal und Merkmalswert in einer Repräsentaion und wird als Größe der Repräsentation bezeichnet. N ist die Anzahl der unterschiedlichen Merkmale, die in der Repräsentation vorkommen. Ein Paar aus Merkmal und Merkmalswert $FV_k(F_i = V_{ij})$ ist wahr, wenn das Merkmal F_i der Instanz den Wert V_{ij} hat.

In [Pazzani 1998] wird das Kartesische Produkt als Operator vorgeschlagen. Für numerische Attribute werden beispielsweise von [Vafaie und Jong 1998] und von [Bloedorn und Michalski 1998] einfache algebraische Operatoren wie Gleichheit, Addition, Subtraktion oder Division und mathematische Funktionen wie Maximum, Minimum oder Durchschnitt eingesetzt.

3.3. Kombinierte Ansätze

Merkmalsauswahl und Merkmalsgenerierung sind stark miteinander verbunden. Wenn die gegebene Repräsentation der Daten ungeeignet ist für die Lösung der Lernaufgabe, kann die ursprüngliche Merkmalsmenge mittels einer Generierung um geeignete Merkmale erweitert werden. Merkmalsauswahl hilt die Merkmalsmenge zu vereinfachen, wenn die Ausgangsrepräsentation der Daten irrelevante oder redundante Merkmale enthält. Auch konstruierte Merkmale können irrelevant oder redundant sein. Es ist daher zweckmäßig diese mit Selektionsmethoden aus der Merkmalsmenge zu entfernen.

Es existieren verschiedene hybride Ansätze, die Merkmalsgenerierung und Merkmalsauswahl miteinander kombinieren, z.b [Bloedorn und Michalski 1998, Lavrac et al. 1998, Markovitch und Rosenstein 2002]. Einige Verfahren verwenden eine Kombination aus Merkmalsgenerierung und probalistischen Suchstrategien zur Merkmalsauswahl. Beispiele für Anwendung dieser Verfahren finden sich in [Bensusan und Kuscu 1996], [Chang und Lippmann 1991] und [Moore et al. 2001]. In [Ritthoff et al. 2002] wird ein modifizierter genetischer Algorithmus für die Merkmalstransformation und die SVM als Induktionsalgorithmus für die Merkmalsevaluation eingesetzt.

3.4. Häufigkeitsbasierte Merkmalsgenerierung mit Hilfe von TF/IDF

In den beiden vorangegangenen Abschnitten wurde ein Überblick über verschiedene existierende Ansätze zu Merkmalsauswahl und Merkmalsgenerierung gegeben. Im vorliegenden Abschnitt wird eine neue Methode vorgestellt, für temporale Daten in Form von Zeitreihen oder Ereignissequenzen Merkmale zu erzeugen. Diese Methode beruht auf dem TF/IDF Maß. Das TF/IDF Maß ist aus dem Information Retrieval bekannt. Zunächst wird TF/IDF definiert, anschließend wird gezeigt, wie es zur Merkmalsgenerierung eingesetzt werden kann.

3.4.1. Definition von TF/IDF

TF/IDF hat seinen Ursprung im Information Retrieval. Im Maschinellen Lernen kommt es häufig bei der Textklassifikation zum Einsatz. Das Problem Textklassifikation besteht darin, Dokumente in eine vorgegebene Menge von Klassen einzuteilen. Bei der Textklassifikation wird meist die vereinfachende Annahme gemacht, dass die Reihenfolge der Worte im Text vernachlässigt werden kann. Dokumente werden nicht als Sequenzen von Worten, sondern als *Multimengen ("bags") von Worten* dargestellt. Diese Repräsentationsform auch wird auch als Bag-of-Words-Ansatz oder als Vektorraumdarstellung bezeichnet.

TF/IDF ist ein Maß, um Worte in der Vektorraumdarstellung zu gewichten und zwar mittels der *Termfrequenz (term frequency)* und der *Dokumentfrequenz (document frequency)* [Salton und Buckley 1988]. Die *Termfrequenz* $TF(w, d)$ gibt an, wie oft das Wort w in Dokument d auftritt. Die *Dokumentfrequenz* $DF(w)$ ist die Anzahl der Dokumente, die das Wort w mindestens einmal enthalten. Man geht davon, dass Worte, die in wenigen Dokumenten vorkommen, aber häufig im aktuellen Dokument auftreten, sehr gute Deskriptoren für den Inhalt dieses Dokumentes sind. Ein hohes Gewicht sollen also intuitiv Worte mit einem hohen $tf(w, d)$-Wert und einem geringen $df(w)$- Wert erhalten. Die Definition der sogenannten *inversen Dokumentfrequenz (Inverse Document Frequency)* bringt diese Annahme zum Ausdruck:

$$idf(w) = \log \frac{|D|}{df(w)} \tag{3.1}$$

Dabei ist D die Dokumentenkollektion, die der Gewichtung zugrunde liegt (Trainingsmenge), also $|D|$ die Anzahl der Dokumente in dieser Kollektion. Das TFIDF-Gewicht eines Wortes w im Dokumentvektor eines Dokumentes d kombiniert die beiden geschilderten Anforderungen wie folgt:

$$tfidf(w, d) = tf(w, d) * idf(w) \tag{3.2}$$

Die Repräsentation eines Dokumentes d ergibt sich entsprechend aus der elementweisen Multiplikation des Vektors der Termfrequenz-Werte des Dokumentes mit dem IDF-Vektor der zugrundeliegenden Dokumentenkollektion:

$$\vec{d} = \vec{tf}(d) * \vec{idf} \tag{3.3}$$

3.4.2. TF/IDF als Operator zur Merkmalsgenerierung

Mit Hilfe von TF/IDF können aus temporalen Daten in Form von Zeitreihen oder Ereignissequenzen Merkmale generiert werden. In diesem Abschnitt werden zwei Möglichkeiten vorgestellt.

Eine Möglichkeit besteht darin, die Änderung von Attributwerten in Zeitreihen oder Ereignissequenzen zu betrachten. In diesem Fall beschreibt die Termfrequenz (*term frequency*), wie oft der Wert eines bestimmten Attributes a_i sich innerhalb einer Zeitreihe c_j ändert. Für eine Zeitreihe mit Zeitpunkten $t_j, j = 1, \ldots, n$ berechnet sich die Termfrequenz eines Attributes a_i dann wie folgt:

$$tf(a_i; c_j) = \sum_{j=2}^{n} I_j \quad \text{mit} \quad I_j = \begin{cases} 0: & a_i(t_j) = a_i(t_{j-1}) \\ 1: & a_i(t_j) \neq a_i(t_{j-1}) \end{cases} \tag{3.4}$$

Dabei bezeichnet $a_i(t_j)$ den Wert des Attributes a_i zum Zeitpunkt t_j.

Eine andere Möglichkeit betrachtet das Auftreten von Merkmalswerten. In diesem Fall beschreibt die Termfrequenz *term frequency*, wie oft ein bestimmtes Attribut a_i innerhalb einer Zeitreihe c_j vorkommt, d.h. wie oft $w(a_i) \neq 0$ ist. Für eine Zeitreihe mit Zeitpunkten $t_j, j = 1, \ldots, n$ berechnet sich die Termfrequenz eines Attributes a_i dann folgendermaßen:

$$tf(a_i, c_j) = \sum_{j=1}^{n} I_j \quad \text{mit} \quad I_j = \begin{cases} 0: & a_i(t_j) = 0 \\ 1: & a_i(t_j) \neq 0 \end{cases} \tag{3.5}$$

Dabei bezeichnet $a_i(t_j)$ den Wert des Attributes a_i zum Zeitpunkt t_j.

Die *document frequency* entspricht bei beiden Defintionen der Anzahl der Zeitreihen, in denen der Wert des Attributes a_i mindestens einmal geändert wurde.

$$df(a_i) = \parallel \{c_j \in C \mid tf(a_i, c_j) > 0\} \parallel \tag{3.6}$$

Der TFIDF-Wert für ein Attribut a_i in eine Zeitreihe c_j berechnet sich folglich als:

$$tfidf(a_i) = tf(a_i, c_j) log \frac{\parallel C \parallel}{df(a_i)} \tag{3.7}$$

4. Mining Mart

Im Kapitel 2 wurden im Abschnitt 2.2 die Phasen des KDD-Prozesses vorgestellt. In der Praxis erweisen sich die vorbereitenden Phasen, insbesondere die Auswahl des Zieldatensatzes, die Vorverarbeitung und die Transformation, am zeit- und arbeitsintensivsten. Die clevere Vorverarbeitung der Daten benötigt 50-80% des Arbeitsaufwands des gesamten Prozess [Pyle 1999]. Gründe dafür sind, dass viele Vorverarbeitungsschritte noch immer selbst programmiert werden müssen und die Auswahl geeigneter Algorithmen für diese Vorverarbeitung das Ergebnis eines „trial-and-error“ Prozesses darstellt.

Das Mining Mart Projekt hat es sich zum Ziel gesetzt, die Vorverarbeitung und die Algorithmenauswahl zu vereinfachen und zu verbessern. Die Idee hinter dem Minig Mart Ansatz beruht auf dem fallbasierten Schließen. Das fallbasierte Schließen geht davon aus, dass es einfacher ist, eine Aufgabe zu lösen, wenn die Lösung einer ähnlichen Aufgabe bekannt ist. Mining Mart bietet eine Umgebung an, die es erlaubt, die Vorverarbeitung der Daten durchzuführen. Die dazu benötigten Operationen werden in Form von Vorverarbeitungsketten modelliert. Diese Vorverarbeitungketten bezeichnet Mining Mart mit dem Begriff Fälle. Diese Fälle werden ebenso wie die eigentlichen Daten in einem Metamodell modelliert. Ein Compiler ermöglicht die Ausführung des alles. Fallmodelle können auf einer Webplattform veröffentlicht werden. Dies ermöglicht anderen Benutzer, sie für ihre eigenen Anwendungen zu modifizieren.

Unterschiedliche Quellen stellen Informationen zum Mining Mart Projekt und dem Mining Mart System zur Verfügung. Ein guter Startpunkt ist die Mining Mart Webseite unter `http://mmart.cs.uni-dortmund.de`. Der Mining Mart Ansatz wird in [Morik und Scholz 2002], [Morik und Scholz 2003], [Kietz et al. 2001], [Kietz et al. 2000] und [Zücker und Kietz 2000] beschrieben.

In diesem Kapitel soll ein Überblick über das Mining Mart System gegeben werden. Die einzelnen Komponenten der Systemarchitektur von Mining Mart und die Umsetzung des im vorherigen Kapitels vorgeschlagenen TF/IDF Operators werden beschrieben.

4.1. Die Systemarchitektur

Abbildung 4.1 gibt einen Überblick über die Systemarchitektur von Mining Mart. Kernbestandteil der Architektur ist ein Metamodell für Metadaten, das sogenannte M4-Modell. Das System besteht aus einem Compiler und einer grafischen Benutzeroberfläche, die verschiedene Editoren zur Verfügung stellt. Mit Hilfe der Editoren können die Metadaten angelegt werden. Weiterer Bestandteil ist eine Webplattform. Aufgabe und Funktion des Metamodells, des Compilers und der Webplattform werden im folgenden näher erläutert.

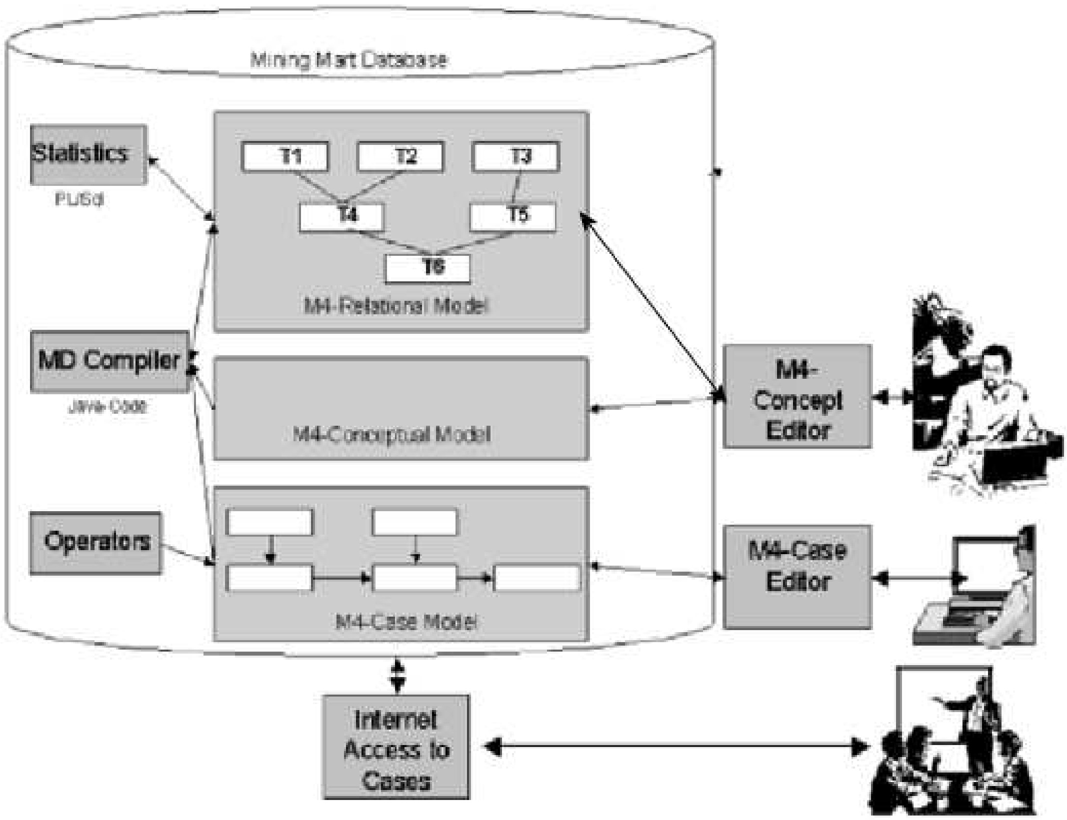


Abbildung 4.1.: Überblick über das Mining Mart System

4.1.1. M4, das Metadatenmodell der Metadaten

Das Metadatenmodell M4 dient dazu, Informationen sowohl über die zu analysierenden Daten als auch über den Prozess der Wissensentdeckung zu speichern. Entsprechend werden zwei Arten von Daten unterschieden: Zum einen die zu analysierenden Daten, sie werden in Mining Mart als Geschäftsdaten (*business data*) bezeichnet. Zum anderen Informationen über die Schritte, die durchgeführt werden, um die Geschäftsdaten in eine für das Lernen geeignete Repräsentation zu überführen. Diese Informationen werden in einem Fall (case) festgelegt. Neben der Unterscheidung der beiden Datentypen wird zwischen einer konzeptionellen und einer relationalen Abstraktionsebene unterschieden. Zur Beschreibung der Geschäftsdaten auf den beiden Abstraktionsebenen existiert das konzeptionelle und das relationale Datenmodell. Der Fall wird auf der konzeptuellen Ebene in Form des Fallmodells modelliert. Alle drei Modelle werden im folgenden näher erläutert.

Das konzeptionelle Datenmodell

Das konzeptionelle Datenmodell (*conceptual data model*) beruht auf einer Ontologie. Mit Hilfe der Ontologie werden die Geschäftsdaten in Form von Konzepten (*concepts*) und Beziehungen (*relationships*) zwischen den Konzepten modelliert. Diese Modellierung abstrahiert von der technischen Repräsentation der Geschäftsdaten in der Datenbank. Statt abstrakter Datenbanknamen können im konzeptuellen Datenmodell verständliche Begriffe zur Bezeichnung der Konzepte, Attribute und Beziehungen verwendet werden. Beispiele für Konzepte sind z.B. `Kunde` und `Produkt`. `Kauft` wäre dann ein Beispiel für eine Beziehung zwischen diesen beiden Konzepten. Für das Anlegen und Bearbeiten des konzeptuellen Datenmodells steht im Mining Mart System der Konzepteditor zur Verfügung (siehe Abbildung 4.2).

Das relationale Modell

Die konzeptuelle Ebene muss auf die eigentlichen Daten, wie sie im relationalen Datenmodell (*relational data model*) beschrieben werden, abgebildet werden. Das relationale Modell speichert Informationen, wie die Daten in der Datenbank gehalten werden. Ein graphischer Editor unterstützt die Abbildung der im konzeptuellen Datenmodell modellierten Konzepte mit ihren Attributen auf die entsprechenden Datenbanktabellen.

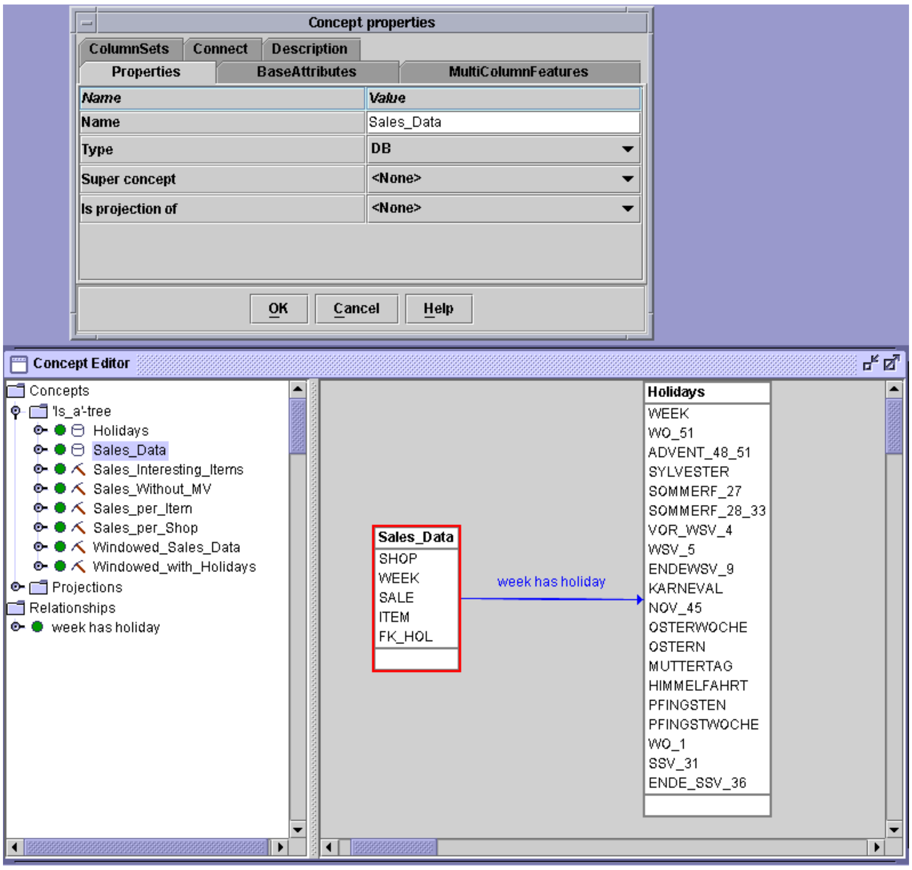


Abbildung 4.2.: Der Konzepteditor

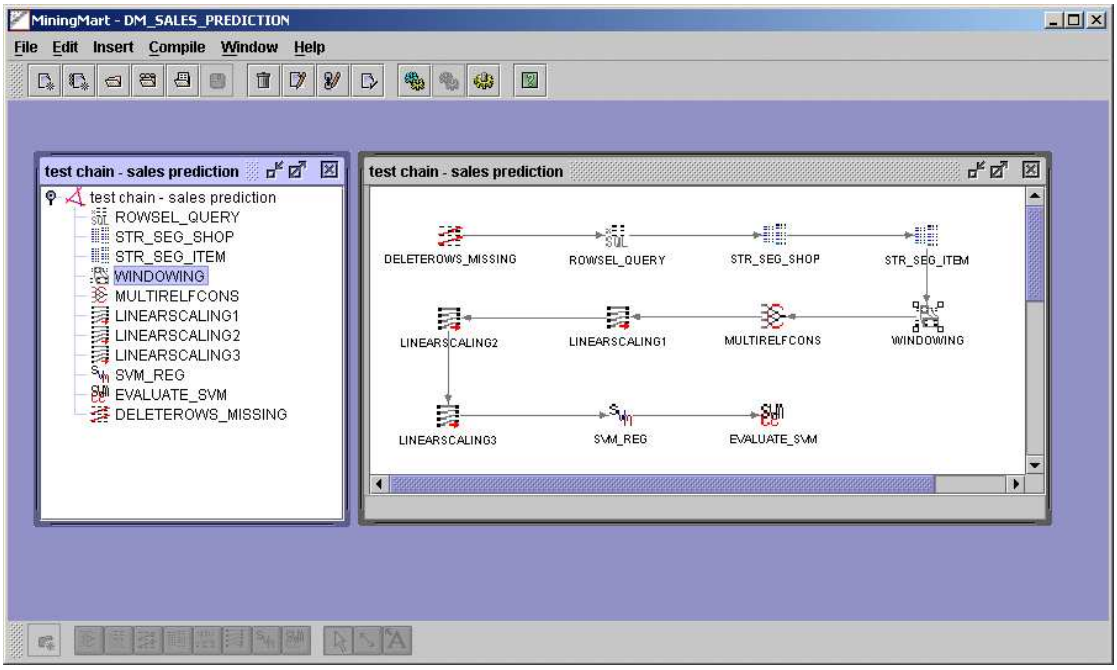


Abbildung 4.3.: Der Falleditor

Das Fallmodell

Ein Fall ist in der Terminolgie von M4 eine Abfolge von Schritten. Ein Schritt besteht in der Anwendung eines Operators. Operatoren führen Datentransformationen wie z.B. Diskretisierung, Behandlung von fehlenden Werten, Aggregation von Attributen oder die Erzeugung von Sequenzen durch. Nach ihrer Ausgabe auf der konzeptuellen Ebene werden zwei Arten von Operatoren unterschieden: Operatoren, die ein Konzept als Ausgabe erzeugen und solche, die ein Attribut als Ausgabe erzeugen. Alle Operatoren haben Parameter, wie z.B. das Eingabekonzept oder das Ausgabe Attribute. Abbildung 4.4 gibt einen Überblick über die Operatoren, die im Mining Mart System zur Verfügung stehen. Im Fallmodell wird die Abfolge der Schritte spezifiziert. Ein grafischer Editor unterstützt das Erstellen und das Anpassen von Fällen. Automatisch überprüfte Anwendbarkeitsbeschränkungen existieren für alle Operatoren. Sie gewährleisten, dass nur gültige Operatorenketten erzeugt werden.

Concept Operators

Joining concepts
MultiRelationalFeatureConstruction
JoinByKey
UnionByKey

Row Selection
RowSelectionByQuery
RowSelectionByRandomSampling
DeleteRecordsWithMissingValues

Segmentation
SegmentationStratified
SegmentationByPartitioning
SegmentationWithKMean
UnSegment

Time series
Windowing
SimpleMovingFunction
WeightedMovingFunction
ExponentialMovingFunction
SignalToSymbolProcessing

Misc
SpecifiedStatistics
Apriori

Feature Selection Operators

FeatureSelectionByAttributes
StochasticFeatureSelection
GeneticFeatureSelection
SGFeatureSelection

Feature Construction Operators

Missing Value
AssignAverageValue
AssignModalValue
AssignMedianValue
AssignDefaultValue
AssignStochasticValue
MissingValuesWithDecisionTree
MissingValuesWithRegressionSVM
MissingValueWithDecisionRules
MissingValueWithDecisionTree
AssignPredictedValueCategorial

Scaling
LinearScaling
LogScaling

Data Mining
SupportVectorMachineForRegression
SupportVectorMachineForClassification
ComputeSVMError
PredictionWithDecisionRules
PredictionWithDecisionTree

Misc
GenericFeatureConstruction

Discretization
TimeIntervalManualDiscretization
NumericIntervalManualDiscretization

Abbildung 4.4.: Überblick über die Operatoren von Mining Mart

4.1.2. Der Compiler

Der M4 Compiler hat die Aufgabe, die verschiedenen Operatoren des Fallmodells in ausführbares SQL zu übersetzen. Unterstützt wird diese Aufgabe durch Spezifikationen, die Ein- und Ausgabe eines Operators definieren. Zum Beispiel erhält ein Operator, der für ein bestimmtes Attribut alle Tupel mit fehlenden Werten entfernt, als Eingabe den Namen des Attributes. Als Ausgabe liefert er ein neues Konzept. Die Datenbanktabelle zu diesem Konzept enthält nur Tupel ohne fehlende Werte für das spezifizierte Attribut.

4.1.3. Die Webplattform

Zum Mining Mart System gehört eine zentrale Webplattform. Diese Webplattform ermöglicht den öffentlichen Austausch und die Dokumentation von exportierten Fällen. Zur Veröffentlichung eines Falles auf der Webplatform werden nur die konzeptionellen Metadaten des Falles übertragen. Benutzer, die im Repository nach relevanten Fällen suchen, werden durch eine navigierbare Repräsentation der konzeptuellen Modelle unterstützt.

4.2. Realisierung des TF/IDF-Operators in Mining Mart

Beide in Kapitel 3 im Abschnitt 3.4.2 vorgestellten Möglichkeiten zur Merkmalsgenerierung mit Hilfe von TF/IDF wurden als Konzeptoperatoren in Mining Mart realisiert. Konzeptoperatoren erhalten alle als Eingabe ein Konzept und erzeugen mindestens ein mit dem Ausgabekonzept verknüpftes ColumnSet. Die M4-Tabelle OP_PARAMS_T

Parametername	Objekttyp	Typ
TheInputConcept	Konzept	Eingabe
TheSelectedAttributes	Attributliste	Eingabe
TheTimeStamp	Attribut	Eingabe
TheKey	Attribut	Eingabe
TheOutputConcept	Konzept	Ausgabe

Tabelle 4.1.: Parameterspezifikationen für die TF/IDF Operatoren in Mining Mart

enthält die Parameterspezifikation für jeden Operator. Tabelle 4.1 gibt einen Überblick über die Parameter, die die neuen Operatoren als Eingabe erwarten bzw. als Ausgabe erzeugen. Der Parameter *TheSelectedAttributes* enthält die Liste der Attribute, für die

TF/IDF Merkmale erzeugt werden sollen. *TheKey* ist der Schlüssel, der eine Zeitreihe identifiziert. *TheTimeStamp* ist das Attribut, das die Zeitpunkte der Zeitreihe angibt.

5. Fallbeispiel: Die Swiss Life-Daten

Die Anwendbarkeit von TF/IDF als Operator zur Merkmalsgenerierung für Zeit-/Ereignisreihen wird an einem Fallbeispiel untersucht. Es handelt sich um temporale Daten über Versicherungsverträge und Partner der Rentenanstalt Swiss Life.

Bevor die Daten detailliert beschrieben werden, wird zunächst die Fragestellung vorgestellt. Es folgt eine kurze Darstellung von anderen Arbeiten zur Analyse der Swiss Life-Daten.

Danach werden die durchgeführten Experimente erläutert. Die notwendige Vorverarbeitung der Daten wurde mit Hilfe des in Kapitel 4 vorgestellten Mining Mart-Systems vorgenommen. Die einzelnen Schritte werden in Abschnitt 5.4.1 beschrieben. Die Anwendung der Lernverfahren auf die vorverarbeiteten Daten erfolgte mit Hilfe von YALE [Ritthoff et al. 2001, Fischer et al. 2002]. Auf die Details wird in Abschnitt 5.4.2 eingegangen. In Abschnitt 5.4.3 werden die erzielten Ergebnisse vorgestellt.

5.1. Die Aufgabe

Die gestellte Aufgabe besteht darin, den möglichen Rückkauf einer Versicherung aus den Versicherungsdaten vorherzusagen. Unter einem Rückkauf versteht man die vorzeitige Beendigung einer kapitalbildenden Lebensversicherung wegen Rücktritt oder Kündigung. Dem Versicherungsnehmer ist in einem solchen Fall der aktuelle Zeitwert der Versicherung auszuzahlen. Für eine Versicherungsgesellschaft bedeuten Rückkäufe immer ein Verlustgeschäft, da kurzfristig Kapital beschafft werden muss bzw. entsprechende Finanzmittel nicht längerfristig und damit gewinnbringend angelegt werden können.

Durch Klassifikation der Versicherungsverträge in zwei Gruppen wird versucht, die gestellte Aufgabe zu lösen. Die eine Gruppe beinhaltet die Verträge, die bereits zurückgekauft wurden, die andere Gruppe beinhaltet die Verträge, die regelrecht fortgeführt wurden. Das durch die erfolgte Klassifikation gewonnene Wissen, kann dazu verwendet werden, Kunden zu identifizieren, deren Verträge rückkaufgefährdet sind. Dies ermöglicht es, Marketingstrategien zu entwickeln, den Rückkauf zu verhindern und die im Bedarfsfall notwendigen finanziellen Rücklagen zu berechnen.

5.2. Die Daten

Die zur Verfügung gestellten Daten sind in einer ORACLE-Datenbank gespeichert. Sie stellt einen Auszug aus dem Datawarehouse von Swiss Life dar. Zur Datenbank gehören insgesamt 12 Tabellen, die über 15 Relationen miteinander verknüpft sind. Abbildung 5.1 gibt einen Überblick über das Datenbankschema. Das abgebildete Datenbankschema ist vergleichbar mit dem in [Staudt et al. 1998] beschriebenen Schema. Dort wird das Datawarehouse von Swiss Life vorgestellt.

In der Tabelle MO_VVERT sind die Verträge gespeichert. Die Tabelle hat 23 Spalten und 1 469 978 Zeilen. Tabelle 5.1 gibt einen Überblick über die Attribute dieser Tabelle und erläutert ihre Bedeutung.

89.5% aller Verträge sind Kapitallebensversicherungen, weitere Versicherungsarten sind: Rentenversicherungen (3.8%), Krankenversicherungen (0.6%), Erwerbsunfähigkeitsversicherungen (2.5%) und Fondsgebundene Lebensversicherungen (3.6%).

Ein einzelner Vertrag besteht aus einem oder mehreren Vertragsteilen, den sogenannten Tarifkomponenten. In den Tarifkomponenten werden die konkreten Vertragskonditionen festgelegt. Zu einem Vertrag gehören durchschnittlich zwei Tarifkomponenten. Sie sind in der Datenbank in der Tabelle MO_TFKOMP abgelegt. Diese Tabelle hat 31 Spalten und 2 194 825 Reihen.

Die Tabelle MO_PART enthält Informationen über Versicherungsteilnehmer. Swiss Life bezeichnet alle Versicherungsteilnehmer als Partner. Die Partner werden über Rollen den Versicherungsverträgen zugeordnet. Es wird zwischen vier Arten von Rollen unterschieden:

- Versicherungsnehmer
  Der Versicherungsnehmer ist der Partner, der mit dem Versicherungsunternehmen den Versicherungsvertrag abschlisst.
- Versicherte Person
  Die Versicherte Person ist der Partner, auf dessen Risiko die Versicherung abgeschlossen wurde.
- Prämienzahler
  Der Prämienzahler ist der Partner, der die Prämien bezahlt.

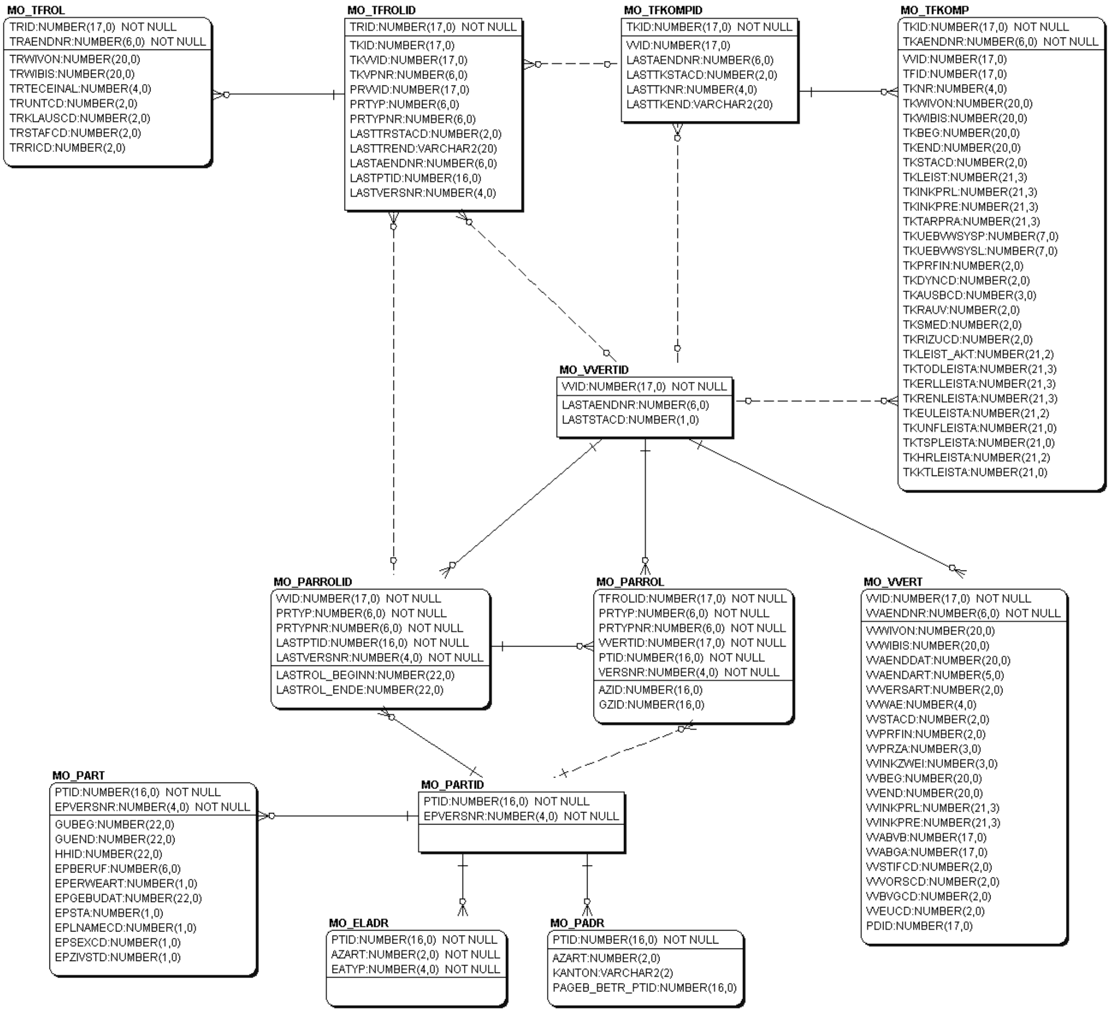


Abbildung 5.1.: Datenbankschema

Attributname	Beschreibung
VVID	Eindeutige Vertragsidentifikation
VVAENDNR	Laufende (aufsteigend) Vorgangsänderungsnummer
VVWIVON	Ab wann ist diese Version wirksam. Datum, das die linke Intervallgrenze eines Wirksamkeitszeitraums bei Zeitraumbearbeitungen angibt
VVWIBIS	Bis wann ist diese Version wirksam. Datum, das die rechte (bis) Intervallgrenye eines Wirksamkeitszeitraums bei Zeitraumsbearbeitungen angibt
VVAENDDAT	Datum, an dem eine Mutation durchgeführt wurde.
VVAENDART	Mutationscode der Bearbeitung eines Vertrages, welcher die Art der Mutation wiedergibt.
VVVERSART	Angabe, ob es sich um eine Kapital-, Renten-, Kranken- oder Erwerbsunfähigkeitsversicherung handelt.
VVWAE	Währung, in welcher die vertraglichen Leistungen und Prämien bezahlt werden
VVSTACD	Versicherungsvertragsstatus; Angabe, in welchem Zustand (auf welcher Stufe) sich ein Versicherungsvertrag bzgl.seines Lebenszyklus befindet (2 = Antrag, 4 = Vertrag; 5 = abgegangener Vertrag).
VVPRFIN	Art der Prämienfinanzierung (0 = keine Angabe, 1 = prämienpflichtig, 3 = prämienfrei wegen Leistungsbezug, 4 = prämienfrei aus EE, 5 = prämienfreigestellt)
VVPRZA	Technisches Feld, original VVINKZWEI aus Transformation.
VVINKZWEI	Inkassozahlweise. Angabe, wieviele Prämienzahlungen innerhalb eines Versicherungsjahres fällig werden bzw. ob es sich um eine Einmalprämie handelt. Diese Angabe gilt für den gesamten Versicherungsvertag. (0 = keine Angabe, 1 = jährlich, 2 = halbjährlich, 3 = vierteljährlich, 4 = monatlich, 5 = Einmaleinlage).
VVBEG	Technischer Versicherungsvertragsbeginn. Datum, ab welchem die Gültigkeit eines Versicherungsvertrages aus versicherungstechnischer Sicht beginnt
VVEND	Datum an dem der gesamte Vertrag abläuft. Ist im Normalfall das Datum an welchem der letzte Versicherungsteil abläuft
VVINKPRL	Jährliche aktuell zu zahlende Pramie (ink. aller Zuschläge) des gesamten Versicherungsvertrages.
VVINKPRE	Geleistete reine Einmaleinlage (Einmalbeitrag) auf Vertragsebene. Bei einem Vertrag gegen Einmalprämie wird hier die einmalig für den Gesamtvertrag zu zahlende Inkassoprämie (= Einmaleinlage) ausgewiesen.
VVABVB	PTID des abschliessenden Vorsorgeberaters. Bei mehreren Beteiligten ist es der wichtigste nach folgender Priorität: 1. RA-Mitarbeiter, 2. der höherer Provisionssatz, 3. sonst der erstgenannte
VVABGA	PTID der Abschluss-Generalagentur, Fremdschlüssel aus PART.
VVSTIFCD	Vorsorgestiftung; Angabe, ob der Versicherungsvertrag eines/einer RA/SL-Mitarbeiters/Mitarbeiterin bei einer Vorsorgestiftung (VSI/VSA) der Rentenanstalt hinterlegt ist. (0 = keine Angabe, 1 = Vorsorgestiftung Innendienst (VSI), 2 = Vorsorgestiftung Aussendienst (VSA), 3 = Gemeinschaftsstiftungen)
VVVORSCD	Vorsorgetyp; (0 = keine Angabe, 1 = Waadtlaender, 2 = Vorsorgepolice, 3 = keine Vorsorgepolice)
VVBVGCD	Kennzeichen, ob der Versicherte einer beruflichen Vorsorge der 2. Säule (BVG versichert) angehört oder nicht. (0 = keine Angabe, 1 = BVG versichert, 2 = nicht BVG versichert)
VVEUCD	Leistungscode von fälligen Invalitätsleistungen
PDID	Eindeutige Indentifikation eines Produkts.

Tabelle 5.1.: Beschreibung der Attribute in der Tablle MO_VVERT

VVID	VVAENDNR	VVWIVON	VVWIBIS	VVAENDDAT	VVAENDART	...
16423	1	1946	1998	1946	1000	
16423	2	1998	1998	1998	27	
16423	3	1998	1998	1998	4	
16423	4	1998	1998	1998	54	
16423	5	1998	1998	1998	4	
16423	6	1998	999	1998	61	
5016	1	1997	1999	1997	33	
5016	2	1999	2001	1999	33	
5016	3	2001	2001	2001	33	
5016	4	2001	2001	2001	33	
5016	5	2001	2002	2001	81	
5016	6	2002	9999	2001	94	
...						

Abbildung 5.2.: Auszug der Tabelle MO_VVERT

- Begünstigter
  Der Begünstigte ist der Partner, der bei Fälligkeit die Versicherungsleistung erhält.

Verschiedene Partner können im Rahmen eines Vertrages die gleiche Rolle ausüben, so kann es z.B. zwei Begünstigte geben. Ebenso kann ein und derselbe Partner mehrere Rollen innehaben. Eine Person kann z.B. Versicherungsnehmer und Prämienzahler sein. Die Zuordnung zwischen den Partnern, den Rollen und den Verträgen geschieht mit Hilfe der Tabelle MO_PARROL. Handelt es sich bei einem Partner um die Versicherte Person, so enthält die Tabelle MO_TFROL Informationen über die ihr zugeteilten Tarifkomponenten.

Die Attribute von Verträgen, Tarifkomponenten und Rollen können während der Vertragslaufzeit geändert werden. Um mögliche Änderungen nachhalten zu können, sind die Tupel der Tabellen MO_TFKOMP, MO_VVERT und MO_PARROL zeitlich gestempelt. In allen drei Tabellen gibt es jeweils zwei Attribute, die den Gültigkeitszeitraum eines Tupels angeben. Jedes Tupel in MO_TFKOMP, MO_VVERT bzw. MO_PARROL repräsentiert somit einen Zustand einer Tarifkomponente, eines Vertrages bzw. einer Rolle. Zu einem Vertrag, einer Tarifkomponente oder einer Rolle gehören mehrere Tupel in der entsprechenden Tabelle. Tabelle MO_VVERT enthält für jeden Vertrag durchschnittlich sechs Tupel und Tabelle MO_TFKOMP enthält für jede Tarifkomponente im Durchschnitt vier Tupel.

Abbildung 5.2 verdeutlicht das Prinzip der Tupelzeitstempelung anhand eines Auszugs der Tabelle MO_VVERT. Die Attribute VVWIVON und VVWIBIS geben den Gültigkeitszeitraum eines Tupels an. Darüber hinaus dokumentiert das Attribut den Transaktionszeitpunkt, d.h. den Zeitpunkt zu dem eine Änderung (Mutation) durchgeführt wurde. Das Attribut VVAENDART erläutert den Grund für jede durchgeführte Änderung. Die Gründe sind durch Nummern kodiert. Anhand der Abbildung wird die Mutationsgeschichte zweier Verträge dargestellt. Um die Tupel der beiden Verträge besser voneinander unterscheiden zu können, sind die Tupel in der Abbildung unterschiedlich farblich markiert. Alle Tupel für den Vertrag mit der Versicherungsnummer 16423 sind türkis hinterlegt und alle Tupel für den Vertrag mit der Versicherungsnummer 5016 sind violett hinterlegt. Für beide Verträge existieren sechs Einträge in der Tabelle, d.h. beide Verträge wurden fünfmal geändert.

5.2.1. Statistische Eigenschaften der Daten

Die Daten sind durch folgende statistische Eigenschaften gekennzeichnet:

- **Verzerrte Verteilung (skewed data)**
  Rückkauf betrifft nur 7.7% aller Verträge. Die Verteilung der rückgekauften Verträge kann man im Sinne von [Bi et al. 2001] als verzerrte Verteilung bezeichnen.

- **Hochdimensionaler Eigenschaftsraum**
  Insgesamt gibt es 118 Attribute. Wenn man die Werte der nominalen Attribute in zusätzliche binäre Attribute transformieren würde, würde man 2 181 401 Attribute erhalten. In so einem hochdimensionalen Eigenschaftsraum versagen Methoden der visuellen Dateninspektion und statistische Methoden.

- **Spärlich besetzte Eigenschaftsvektoren**
  Würde man die Ausprägungen aller Attribute in binäre Attribute transformieren, erhielte man spärlich besetzte Eigenschaftsvektoren.

Diese Eigenschaften machen die Komplexität der Daten deutlich. Die Zeitstemplung der Daten erhöht die Komplexität weiter. Für die Lösung der Aufgabe müssen die Rohdaten in einen Merkmalsraum übertragen werden, auf den Lernverfahren anwendbar sind.

5.3. Bisherige Untersuchungen

Die beschriebenen Swiss Life-Daten waren bereits zweimal Gegenstand von Analysen. Sowohl die Projektgruppe 402 [Bauschulte et al. 2002], die im Wintersemester 2001/2002 und im Sommersemester 2002 am Lehrstuhl für Künstliche Intelligenz des Fachbereichs Informatik der Universität Dortmund stattfand, als auch Jens Fisseler in seiner Diplomarbeit „Anwendung eines Data Mining-Verfahres auf Versicherungsdaten" [Fisseler 2003] befassten sich mit diesen Daten. Beide Arbeiten berücksichtigen die zeitliche Dimension der Daten unterschiedlich. Während die Projektgruppe in ihren Untersuchungen die zeitliche Dimension nicht explizit verwendet hat, wurden in der Diplomarbeit die Daten nach häufigen Sequenzen untersucht. Im folgenden werden beide Ansätze kurz vorgestellt.

5.3.1. Vorhersage von Rückkauf ohne Berücksichtigung der Zeit

Ein erster Ansatz der Projektgruppe bestand darin, aus den Daten der Partner Aufschluss über die Rückkaufwahrscheinlichkeit eines Vertrages zu erhalten. Aus der Partnertabelle wurden zehn Attribute ausgewählt. Zusätzlich wurde ein binäres Attribut *Rückkauf* aus den Rohdaten generiert. Als Lernverfahren kamen ein Entscheidungsbaumlerner und eine SVM zum Einsatz. Mit dem Entscheidungsbaumlerner wurde eine Precision von 57% und ein Recall von 80% erreicht. Mit der SVM wurde bei der besten Parametereinstellung eine Precision von 11% und ein Recall von 57% erzielt. Die Daten wurden außerdem mit Hilfe von Assoziationsregeln analysiert. Die entstehenden Regeln wurden nach dem Attribut Rückkauf in der Konklusion gefiltert. Es ergaben sich Korrelationen zwischen einigen Attributen. Diese Korrelationen waren jedoch nicht signifikant in bezug auf Rückkauf.

Aus den Ergebnissen aller Untersuchung kann der Schluss gezogen werden, dass die Partnerdaten keine relevanten Informationen zur Vorhersage von Rückkauf enthalten.

Änderungen in der persönlichen Situation eines Partners, wie z.B. der Kauf eines Hauses, Heirat oder die Geburt eines Kindes sind in der Datenbank nicht gespeichert. Diese Ereignisse können nur indirekt über die Änderung eines Vertrages oder eines seiner Tarifkomponenten beobachtet werden. Deshalb betrachtete die Projektgruppe in einem zweiten Ansatz die Änderungen der Tarifkomponenten. Mit Hilfe von Apriori wurde versucht, typische Veränderungen von Vertragsmerkmalen in zurückgekauften Versicherungsverträgen zu entdecken. Die Beendigung der Prämienzahlung war eine charakteristische Änderung, die ermittelt werden konnte. Sie kann allerdings nicht als Kriterium zur Vorhersage von Rückkauf verwendet werden.

Die Ergebnisse können folgendermaßen interpretiert werden:

1. Die Daten enthalten keine relevanten Informationen zur Vorhersage von Rückkauf oder
2. die Repräsentation der Daten wurde ungeeignet gewählt.

Aus den durchgeführten Analysen der Partnerdaten kann daher gefolgert werden, dass sie für die Rückkaufvorhersage ungeeignet sind. Eine Analyse der Versicherungsdaten verspricht in dieser Hinsicht mehr Erfolg.

5.3.2. Vorhersage von Rückkauf auf der Basis von Intervallsequenzen

Jens Fisseler untersuchte in seiner Diplomarbeit [Fisseler 2003] die Swiss Life-Daten nach häufigen Änderungsmustern. Aus diesen Mustern versuchte er, Regeln zu erzeugen, um mit ihrer Hilfe die zeitliche Entwicklung von Intervallsequenzen vorhersagen zu können. In Anlehnung an den Ansatz von Höppner [Höppner 2002] entwickelte Jens Fisseler einen Algorithmus zur Auffindung von häufigen Intervallmustern. Getrennt nach den fünf verschiedenen Versicherungsarten, wendete Fisseler seinen Algorithmus auf die Tarifkomponentendaten an. Die Intervalle erzeugte er aus den Zeitmarken der Änderungen. Mit Hilfe von Allens zeitlicher Intervalllogik [Allen 1984] formulierte er Relationen zwischen den erzeugten Intervallen.

Durch Filterung nach Weltzeit und Laufzeit überprüfte er die gefundenen Regeln auf zeitliche Veränderungen. Die Filterung nach der Weltzeit ergab unter anderem, dass Mitte der 90er Jahre viele Verträge zurückgekauft wurden. Dies ist auf die Einführung der sogenannten Stempelsteuer im April 1998 zurückzuführen. Die Stempelsteuer kann auf Wertpapiere, auf Quittungen von Versicherungsprämien und auf andere Urkunden des Handelsverkehrs erhoben werden. Die Einführung der Stempelsteuer veranlasste viele Kunden, ihre Versicherungen aufzulösen.

Jens Fisseler versuchte eine Rückkaufsvorhersage sowohl unter alleiniger Verwendung der Tarifkomponentendaten als auch in Kombination mit Daten aus den Versicherungsverträgen. Bei beiden Untersuchung konnte er keine Regeln zur Vorhersage von Rückkauf abgeleiten.

5.4. Experimente

Zusammenfassend lässt sich sagen, dass es in den vorgestellten Untersuchungen nicht gelungen ist, einen Ansatz zur Vorhersage von Rückkauf zu finden. Die bisher erzielten

Ergebnisse lassen zwei mögliche Rückschlüsse zu:

1. Die Daten sind nicht adäquat repräsentiert worden, d.h. in den Daten stecken relevante Informationen, die in den durchgeführten Analysen nicht erkannt wurden.

2. Die Daten enthalten keine Informationen zur Vorhersage von Rückkauf.

Bisher nicht untersucht wurde die Häufigkeit von Vertragsänderungen. In der Annahme, dass häufige Änderungen eines Vertrages Ausdruck von Unzufriedenheit des Kunden sind, könnte die Berücksichtigung von Vertragsänderungen, die Vorsage von Rückkauf ermöglichen. Eine TF/IDF Darstellung berücksichtigt die Häufigkeit von Änderungen. Daher werden in der vorliegenden Diplomarbeiten die Versicherungsdaten in einer TF/IDF Darstellung neu analysiert.

5.4.1. Vorverarbeitung

Die notwendige Vorverarbeitung der Versicherungsdaten wurde mit Hilfe des Mining Mart Systems durchgeführt. Zunächst wird das konzeptionelle Datenmodell erstellt. Danach können die Vorverarbeitungsschritte im Fallmodell modelliert werden.

Erstellen des konzeptionellen Datenmodells

Die beiden Datenbanktabellen MO_VVERT und MO_VVERTID werden als Konzepte modelliert. Zur Modellierung der Tabelle MO_VVERT wird das Konzept *Vertraege* angelegt. Die Tabelle MO_VVERTID wird durch das Konzept *VertragsID* repräsentiert. Für jedes Attribute der beiden Tabellen wird ein entsprechendes Attribut zu den Konzepten angelegt. Abbildung 5.3 zeigt das Konzept *Vertraege*. Die Konzepte mit den Attributen werden auf die Datenbanktabellen abgebildet. Abbildung 5.4 zeigt die Abbildung der Attribute für das Konzept *Vertraege*.

Erstellen des Fallmodells

Zunächst wird ein binäres Attribut *Rückkauf* erzeugt. Dieses Attribut nimmt den Wert 1 an, wenn der Vertrag zurückgekauft wurde und hat den Wert 0 bei nicht erfolgtem Rückkauf.

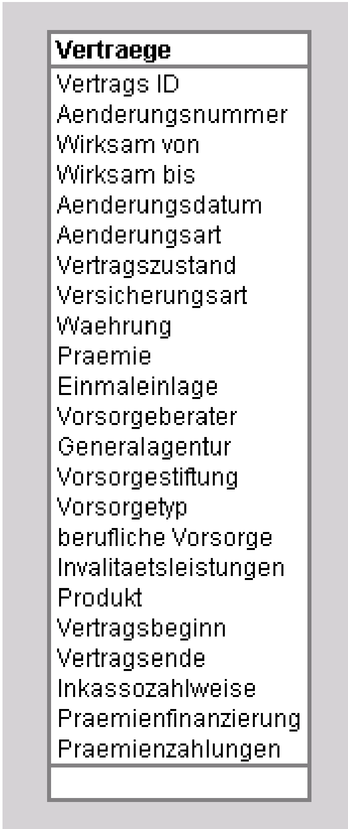
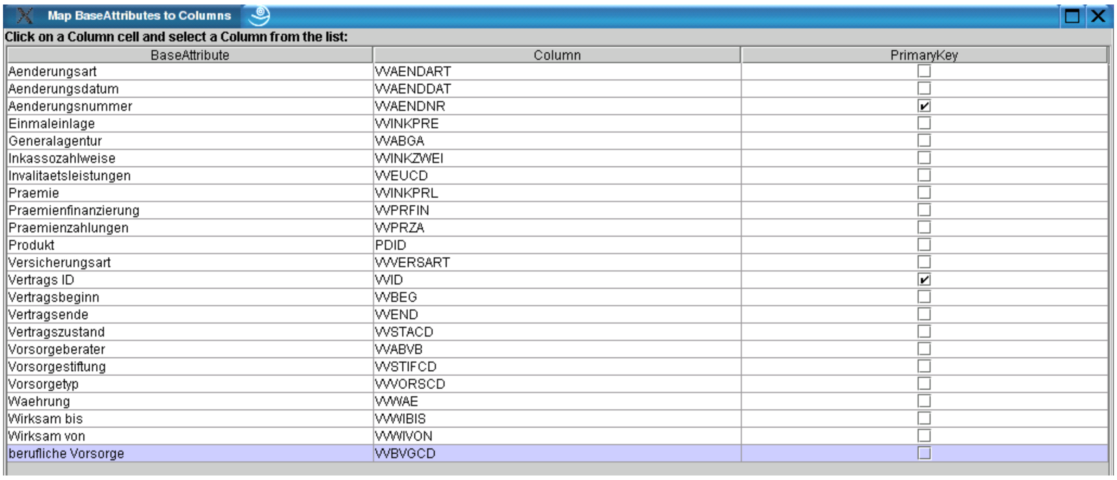


Abbildung 5.3.: Konzept *Vertraege*

Erstellen eines binären Attributes Rückkauf Abbildung 5.5 gibt einen Überblick über die notwendigen Operatoren zur Erstellung des binären Attributs *Rückkauf*. Mit Hilfe des Operators *UserDefinedFeatureSelection* werden alle Attribute ausgewählt, die Informationen über die Änderungen von Verträgen erfassen. Es handelt sich dabei um die Attribute *Aenderungsnummer*, *Aenderungsdatum* und *Aenderungsart*. *Aenderungsnummer* ist die fortlaufende Änderungsnummer, *Aenderungsdatum* ist das Datum, an dem die Änderung erfolgte und *Aenderungsart* gibt den Grund für die durchgeführte Änderung an. Alle Gründe sind durch Nummern kodiert.

Mit Hilfe des Operators *RowSelByQuery* werden anhand des Attributs *Aenderungsart* alle rückgekauften Verträge selektiert (Schritt *Selektion der rueckgekauften Vertraege*).

Ein Vertrag kann jedoch nach einem Rückkauf reaktiviert und danach erneut zurückkauft werden. Für solche Verträge muss das Tupel mit der höchsten Änderungsnummer


Map BaseAttributes to Columns

Click on a Column cell and select a Column from the list:

BaseAttribute	Column	PrimaryKey
Aenderungsart	WAENDART	☐
Aenderungsdatum	WAENDDAT	☐
Aenderungsnummer	WAENDNR	☑
Einmaleinlage	WINKPRE	☐
Generalagentur	WABGA	☐
Inkassozahlweise	WINKZWEI	☐
Invalitaetsleistungen	WEUCD	☐
Praemie	WINKPRL	☐
Praemienfinanzierung	WPRFIN	☐
Praemienzahlungen	WPRZA	☐
Produkt	PDID	☐
Versicherungsart	WVERSART	☐
Vertrags ID	WID	☑
Vertragsbeginn	WBEG	☐
Vertragsende	WEND	☐
Vertragszustand	WSTACD	☐
Vorsorgeberater	WABVB	☐
Vorsorgestiftung	WSTIFCD	☐
Vorsorgetyp	WVORSCD	☐
Waehrung	WWAE	☐
Wirksam bis	WWIBIS	☐
Wirksam von	WWIVON	☐
berufliche Vorsorge	WBVGCD	☐

Abbildung 5.4.: Abbildung der Attribute des Konzeptes Vertraege auf die Datenbankattribute der Tabelle MO_VVERT

für Rückkauf bestimmt werden. Dies gelingt mit Hilfe des Operators *SpecifiedStatitics* (Schritt *Bestimmung der maximalen Aenderungsnummer*).

Alle Verträge, die nach einem Rückkauf reaktiviert und nicht erneut zurückgekauft wurden, werden entfernt. Dazu werden die Operatoren *JoinByKey* (Schritt *Verbund(2)*), *GenericFeatureConstruction* (Schritt *Differenzbildung (2)*) und *RowSelByQuery* (Schritt *Auswahl der rueckgekauften Vertraege mit Reaktivierung*) benötigt. Nach der Anwendung dieser Operatoren sind alle Verträge bestimmt, die nach einer Reaktivierung erneut zurückgekauft wurden.

Im nächsten Schritt werden alle rückgekauften Verträge ermittelt, bei denen keine Reaktivierung stattgefunden hat. Dies geschieht mittels der Operatoren *UnionByKey* (Schritt *Vereinigung (1)*) und *RowSelByQuery* (Schritt *Auswahl der rueckgekauften Vertraege ohne Reaktivierung*).

Die Menge aller rückgekauften Verträge erhält man durch die Vereinigung der rückgekauften Verträge mit und ohne Reaktivierung (Schritt *Vereinigung (2)*). Der Operator *GenericFeatureConstruction* erzeugt das binäre Attribut *Rückkauf* (Schritt *Konstruktion eines binären Attributes Rueckkauf*). Das Attribut *Rückkauf* hat für alle rückgekauften Verträge den Wert 1.

Zur Identifizierung aller nicht zurückgekauften Verträge wird zunächst mittels des Operators *UserDefinedFeatureConstruction* das Attribut *VertragsID* des Konzeptes *VertragsIDs* ausgewählt. Das erstellte Konzept wird mit den rückgekauften Verträgen vereinigt

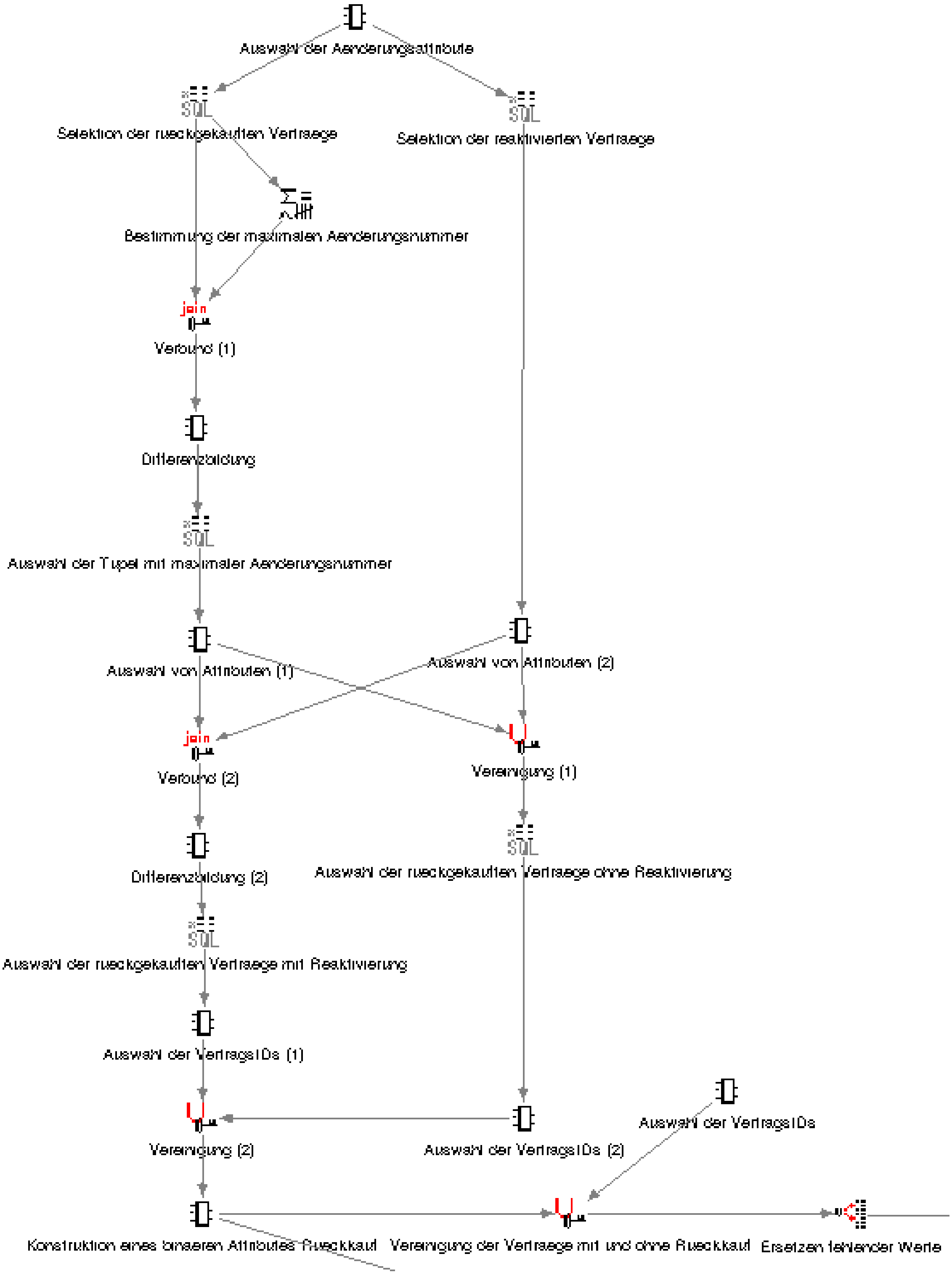


Abbildung 5.5.: Überblick über die Schritte in MiningMart zur Erstellung eines binären Attributes Rückkauf

(Schritt *Vereinigung der Vertraege mit und ohne Rueckkauf*). Für alle nicht zurückgekauften Verträge hat das Attribut *Rückkauf* nach der Vereinigung fehlende Werte. Der Operator *AssignDefaultValue* ersetzt alle fehlenden Werte mit dem Wert 0 (Schritt *Ersetzen fehlender Werte*).

Attributauswahl Nicht alle 23 Merkmale des Konzepts *Vertraege* können zur Generierung von TF/IDF Merkmalen herangezogen werden. Es können nur Merkmale berücksichtigt werden, die keine fehlende Werte aufweisen und die sich innerhalb der vertraglichen Laufzeit ändern. Die Attribute *Versicherungsart* und *Währung* ändern sich nicht über die Zeit. Die Attribute *Generalagentur* und *Vorsorgeberater* weisen fehlende Werte auf. Die erwähnten vier Attribute werden folglich nicht zur Generierung verwendet. Die Attribute *VertragsID*, *Aenderungsnummer*, *Aenderungsdatum*, *Wirksam von* und *Wirksam bis* können ebenfalls nicht berücksichtigt werden. Die verbliebenen 14 Attribute werden unterschiedlich weiterverwendet: Eines der Attribute wird vor der TF/IDF Transformation in binäre Merkmale überführt; die anderen 13 werden unmittelbar in TF/IDF Merkmale transformiert.

Generierung von binären Merkmalen Das Attribut *Aenderungsart* gibt den Grund für die Änderung eines Vertrages an. Insgesamt existieren 123 Gründe. Zwei von ihnen bedeuten Rückkauf, weitere drei treten immer im Zusammenhang mit Rückkauf auf. Diese fünf Gründe müssen bei der Anwendung der Lernverfahren ausgeschlossen werden. Die restlichen 118 Werte des Attributs *Aenderungsart* werden in binäre Attribute transformiert (siehe Abbildung 5.6). Für jeden Wert wird ein neues Merkmal generiert. Die so erhaltenen Attribute werden mit dem Namen MUTx bezeichnet, wobei x für die Codenummer steht. Für ein Tupel nimmt MUTx den Wert 1 an, wenn das Attribut *Aenderungsart* den Wert x hat. Durch diese Vorgehensweise entsteht ein Merkmalsraum mit 131 Merkmalen.

Merkmalsgenerierung mit Hilfe von TF/IDF Um TF/IDF Merkmale zu generieren, wird die Änderungsgeschichte jedes Vertrages betrachtet. Alle Tupel eines Vertrages werden gemäß ihres Zeitstempels aufsteigend sortiert. Für die 13 unveränderten Attribute werden TF/IDF Merkmale nach der ersten der in Kapitel 3 im Abschnitt 3.4.2 beschriebenen Variante erzeugt. Diese Variante erfasst die Änderung von Attributwerten. Der Termfrequenz entspricht hier die Anzahl der Änderungen, die ein Attribut a_i während der Laufzeit eines Vertrags c_j erfahren hat.

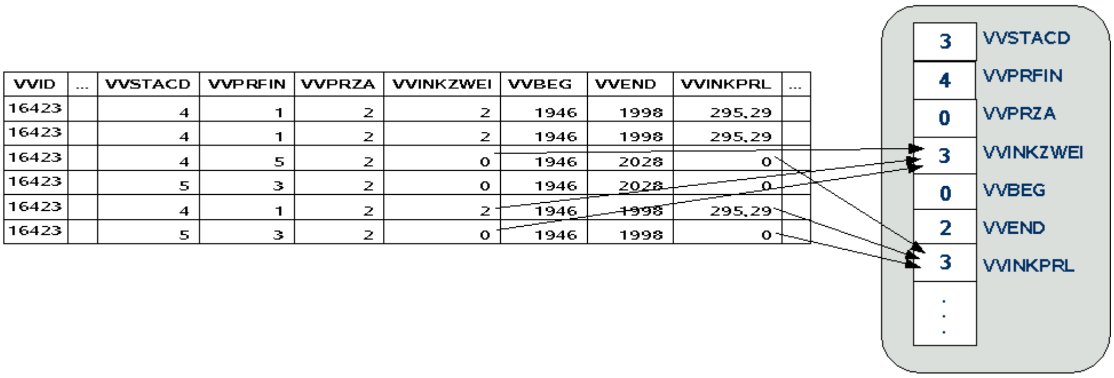
$$tf(a_i, c_j) = \| \{x \in \text{Zeitpunkte} \mid a_i \,\text{von}\, c_j \,\text{wurde geändert}\} \| \tag{5.1}$$

Abbildung 5.7 illustriert die Berechnung der Termfrequenz für die genannten 13 Merkmale.

VVID	VVAENDNR	VVAENDART	VVSTACD	VVPRFIN	VVPRZA	...
16423	1	1000	4	1	2	
16423	2	27	4	1	2	
16423	3	4	4	5	2	
16423	4	54	5	3	2	
16423	5	4	4	1	2	
16423	6	61	5	3	2	
...						

VVID	MUT1	...	MUT27	...	MUT54	...	MUT1000	...	VVSTACD	VVPRFIN	...
16423	0		0		0		1		4	1	
16423	0		1		0		0		4	1	
16423	0		0		0		0		4	5	
16423	0		0		1		0		5	3	
16423	0		0		0		0		4	1	
16423	0		0		0		0		5	3	
...											

Abbildung 5.6.: Transformation des Attributes *Aenderungsart* in binäre Attribute



VVID	...	VVSTACD	VVPRFIN	VVPRZA	VVINKZWEI	VVBEG	VVEND	VVINKPRL	...
16423		4	1	2	2	1946	1998	295.29	
16423		4	1	2	2	1946	1998	295.29	
16423		4	5	2	0	1946	2028	0	
16423		5	3	2	0	1946	2028	0	
16423		4	1	2	2	1946	1998	295.29	
16423		5	3	2	0	1946	1998	0	

Abbildung 5.7.: Berechnung der Termfrequenz für die originalen Merkmale

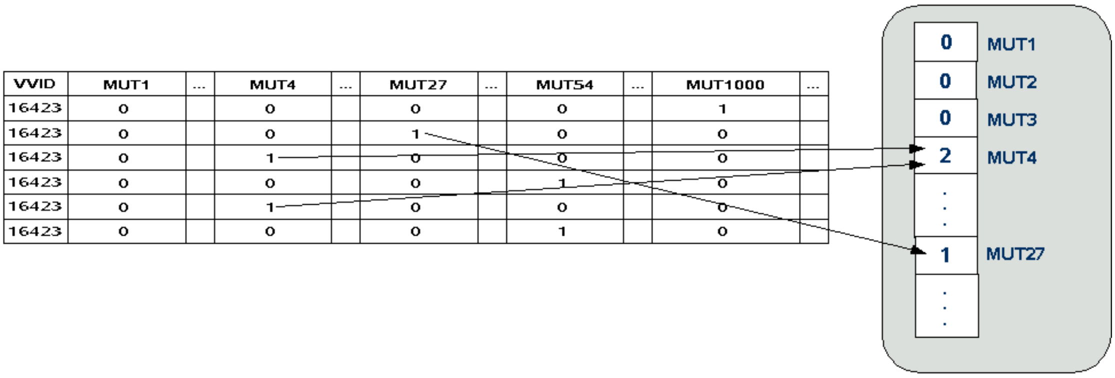


Abbildung 5.8.: Berechnung der Termfrequenz für die binären Merkmale

Für die neu erzeugten binären Merkmale wird die zweite Variante zur Generierung von TF/IDF Merkmalen eingesetzt (siehe Kapitel 3 Abschnitt 3.4.2). Diese Variante betrachtet das Vorkommen von Attributwerten. Die Termfrequenz erfasst hier, wie oft ein Änderungsgrund auftritt.

$$tf(a_i, c_j) = \parallel \{x \in \text{Zeitpunkte} \mid a_i = 1\} \parallel \tag{5.2}$$

Abbildung 5.8 verdeutlicht die Berechnung der Termfrequenz.

Die *Dokumentfrequenz* eines Attributes a_i enspricht der Anzahl der Verträge mit einer *Termfrequenz* größer als 0.

$$df(a_i) = \parallel \{c_j \in C \mid tf(a_i, c_j) > 0\} \parallel \tag{5.3}$$

Erzeugung von Vergleichsrepräsentationen Ein Vergleich der TF/IDF Repräsentation mit anderen Repräsentationen soll den Vorteil der TF/IDF Repräsentation beweisen. Es werden zwei Vergleichsrepräsentationen erzeugt.

Für die erste Repräsentation werden die 14 Merkmalen verwendet, die im Abschnitt Attributauswahl beschrieben wurden. Die zeitliche Dimension der Daten wird nicht berücksichtigt. Bei Verträgen ohne Rückkauf wird das zuletzt gültige Tupel verwendet. Bei Verträgen mit erfolgtem Rückkauf wird das letzte vor dem Rückkauf gültige Tupel ausgewählt. Abbildung 5.9 verdeutlicht die Auswahl der Tupel.

Als zweite Repräsentation wird eine binäre Darstellung gewählt. Im Gegensatz zur TF/IDF Repräsentation erfasst diese Darstellung nur das Auftreten von Änderungen bzw. Ände-

VVID	VVAENDNR	VVAENDART	VVSTACD	VVPRFIN	VVPRZA	...	Rueckkauf
16423	1	1000	4	1	2		1
16423	2	27	4	1	2		1
16423	3	4	4	5	2		1
16423	4	54	5	3	2		1
16423	5	4	4	1	2		1
16423	6	61	5	3	2		1
5016	1	33	4	1	1		-1
5016	2	33	4	1	1		-1
5016	3	33	4	1	1		-1
5016	4	33	4	1	1		-1
5016	5	81	4	1	1		-1
5016	6	94	5	3	1		-1
...							

VVID	VVAENDART	VVSTACD	VVPRFIN	VVPRZA	...	Rueckkauf
16423	4	4	1	2		1
5016	94	5	3	1		-1
...						

Abbildung 5.9.: Erzeugung der Repräsentation mit den originalen Merkmalen

VVID	VVSTACD	VVPRFIN	...	MUT4	...	MUT33	...
16423	4	1		0		0	
16423	4	1		0		0	
16423	4	5		1		0	
16423	5	3		0		0	
16423	4	1		1		0	
16423	5	3		0		0	
5016	4	1		0		1	
5016	4	1		0		1	
5016	4	1		0		1	
5016	4	1		0		1	
5016	4	1		0		0	
5016	5	3		0		0	
...							

VVID	VVSTACD	VVPRFIN	...	MUT4	...	MUT33	...
16423	1	1		1		0	
5016	1	1		0		1	
...							

Abbildung 5.10.: Erzeugung der binären Repräsentation

rungsgründen und nicht die Häufigkeit. Mathematisch läßt sich das wie folgt ausdrücken:

$$bin(a_i; c_j) = \begin{cases} 1: & a_i \text{ wurde in } c_j \text{ geändert bzw. } a_i \text{ tritt in } c_j \text{ auf} \\ 0: & \text{sonst} \end{cases} \tag{5.4}$$

Abbildung 5.10 zeigt die Erzeugung der binären Repräsentation.

5.4.2. Durchführung der Lernläufe

Gütekriterien

In diesem Abschnitt werden die Gütekriterien *Accuracy*, *Precision*, *Recall* und *F-Measure* vorgestellt. Die genannten Kriterien werden zur Bewertung der durchgeführten Experimente eingesetzt. Sie werden anhand der Kontingenztafel in Abbildung 5.2 erläutert. *A*

	T(b) = +	T(b) = -
H(b) = +	A	B
H(b) = -	C	D

Tabelle 5.2.: Kontingenztafel für ein Klassifikationsproblem mit zwei Klassen. T(b) entspricht der korrekten Klassifikation, H(b) ist die Klassifikation des Lernalgorithmus (Hypothese).

bezeichnet die Anzahl der positiven Beispiele, die vom Klassifikator als positiv erkannt wurden. B ist die Anzahl der vom Lernalgorithmus fälschlicherweise als positiv klassifizierten, in Wirklichkeit negativen Beispiele. C gibt die Anzahl positiver Beispiele an, die vom Lernalgorithmus fälschlicherweise als negativ klassifiziert wurden, und D ist die Anzahl der korrekt als negativ erkannten Beispiele.

Die *Accuracy* gibt die Wahrscheinlichkeit an, dass ein zufällig aus der Verteilung der Beispiele gezogenes Beispiel richtig klassifiziert wird. Eine Abschätzung der *Accuracy* kann erfolgen mittels Division der Anzahl der korrekten Klassifikationen durch die Anzahl aller Klassifikationen. Es ergibt sich folgende Formel:

$$Accuracy = \frac{A + D}{A + B + C + D} \tag{5.5}$$

Unter *Precision* versteht man die Wahrscheinlichkeit, dass ein als positiv klassifiziertes Beispiel wirklich positiv ist.

$$Precision = \frac{A}{A + B} \tag{5.6}$$

Recall ist die Wahrscheinlichkeit, dass ein positives Beispiel auch als solches erkannt wird. Dieses Maß läßt sich wie folgt abschätzen:

$$Recall = \frac{A}{A + C} \tag{5.7}$$

Sollen verschiedene Experimente miteinander verglichen werden, ist es schwierig, jeweils Precision und Recall aus den einzelnen Experimenten miteinander zu vergleichen. Aus diesem Grund werden Precision und Recall oft zu einem einzigen Wert kombiniert.

Ein gebräuchliches Maß zur Kombination von Precision und Recall ist das von Lewis [Lewis 1995] definitierte *F-Measure*. Es berechnet das gewichtete harmonische Mittel aus Recall und Precision.

$$F_\beta = \frac{(\beta^2 + 1) Prec(h) Rec(h)}{\beta^2 Prec(h) + Rec(h)} \tag{5.8}$$

β gibt die relative Gewichtung zwischen Precision und Recall an. In der Regel wird $\beta = 1$ gesetzt, so dass beide Werte gleich gewichtet werden.

Eingesetzte Lernfahren

Zur Klassifikation wurden die in Kapitel 2 beschriebenen Lernverfahren eingesetzt: Apriori zur Klassifikation (siehe 2.3.4), J4.8[1] (siehe 2.3.3), die SVM (siehe 2.3.1) und Naive Bayes (siehe 2.3.2). Diese Verfahren sind im Rahmen des WEKA [Witten und Frank 2000] Paketes in YALE integriert. YALE stellt darüber hinaus Operatoren zur automatischen Durchführung von Kreuzvalidierungen und zur Bestimmung von unterschiedlichen Gütekriterien zur Verfügung. Zur Durchführung der Experimente können mit Mining Mart vorverarbeitete Daten mittels eines Operators direkt aus der Datenbank extrahiert werden. Abbildung 5.11 zeigt den Aufbau eines Experimentes mit YALE. Apriori wurde zur Klassifikation verwendet, indem zunächst auf allen Trainingsbeispielen Assoziationsregeln gelernt wurden. Die Regelgenerierung erfolgte für einen Support von 1% und einen Konfidenzwert von 50%. Die erzeugten Regeln wurden anschließend auf Rückkauf in der Konklusion gefiltert. Die gefilterten Regeln wurden auf die Testdaten angewandt, um sie zu klassifizieren. Bei der SVM wurde ein linearer Kernel mit C = 0,01 verwendet.

Validierungsstrategien

Die Wahl geeigneter Testdaten ist entscheidend für exakte Abschätzungen zukünftiger Fehlklassifikationen.

Für die Lernverfahren sind rückgekaufte Verträge positive Beispiele. Negative Beispiele sind Verträge ohne Rückkauf. Rückkauf tritt nur bei 7.7% aller Verträge auf. Es überwiegt die Anzahl der negativen Beispiele. Diese ungleiche Verteilung von positiven und negativen Beispielen bereitet beim Erzeugen der Trainingsbeispiele Probleme. Werden bei dieser Verteilung die Trainingsbeispiele zufällig aus der Gesamtmenge aller Verträge gezogen, können die Lernverfahren aufgund der geringen Anzahl der positiven Beispiele kein Modell zur zuverlässigen Vorhersage von Rückkauf lernen. Um diesem Problem zu begegnen, wurden zunächst verzerrte Stichproben (biased samples) für das Training verwendet. Sie wurden erzeugt, indem Mengen von 1000 und 2000 Beispielen aus der Gesamtmenge mit jeweils gleicher Anzahl von positiven und negativen Beispielen entnommen wurden. Als Testdatensatz dienten alle übrigen Verträge. Bei dieser Methode besteht die Gefahr, dass die Stichprobe nicht repräsentativ ist. Das hat zur Folge, dass bei zwei unterschiedlichen Stichproben weit auseinander liegende Lernergebnisse erzielt werden können. So wurde in einem Lernlauf auf der TF/IDF Repräsentation mit der SVM bei einer Trainingsmenge von 1000 Beispielen eine Precision von 77,76%, ein Recall

[1] J4.8 ist eine jüngere und leicht verbesserte Version von C4.5 Revision 8, der letzten veröffentlichten Version der C4.5-Algorithmenfamilie. Nach C4.5 wurde C5.0 als kommerzielle Implementierung freigegeben [Witten und Frank 2000].

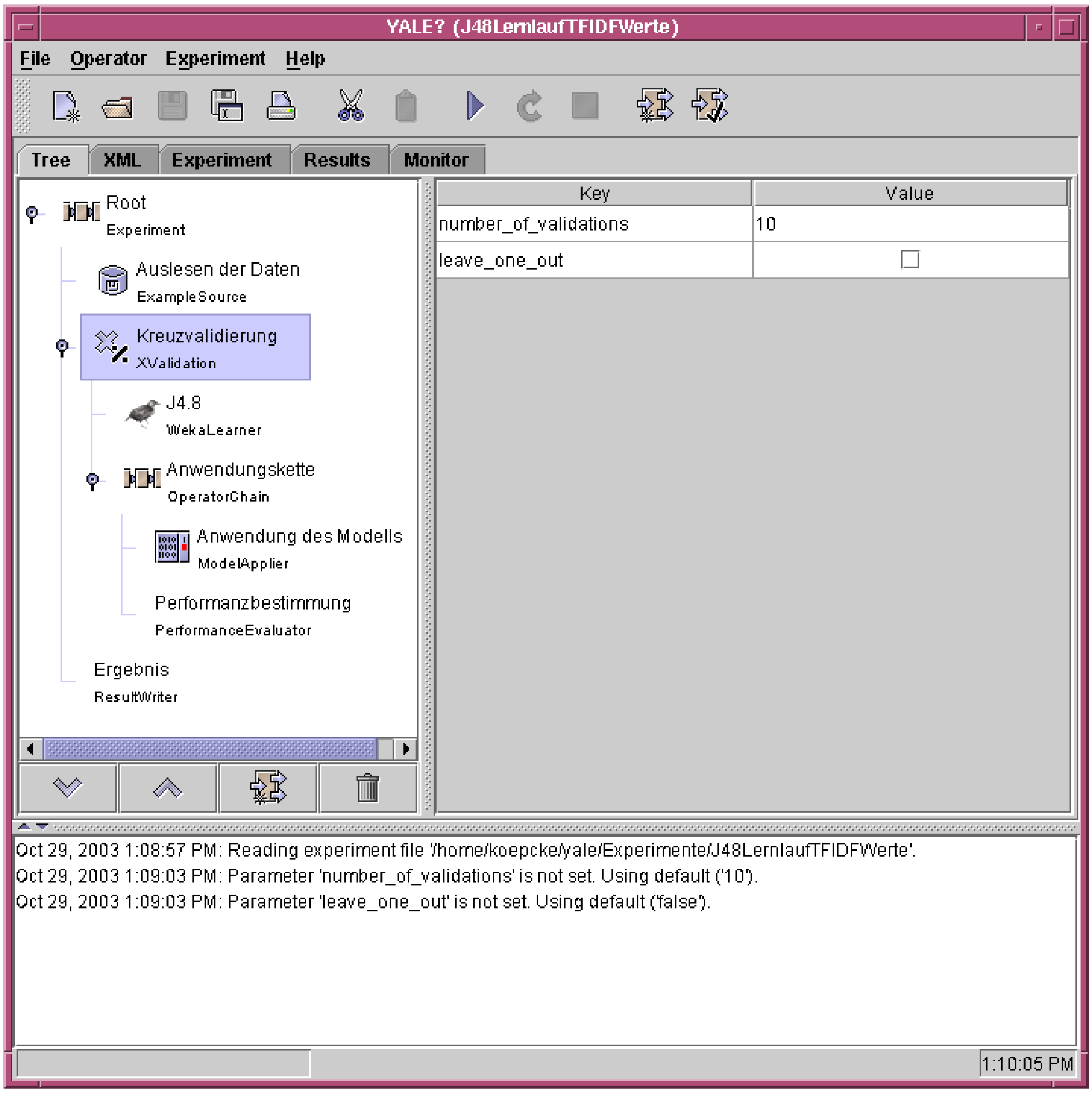


Abbildung 5.11.: Ein Experiment in YALE

von 99,9% und eine Accuracy von 97,78%, in einem anderen Lernlauf wurde dagegen nur eine Precision von 47,21%, ein Recall von 80,49% und eine Accuracy von 91,52% erreicht.

Verzerrte Stichproben sind folglich keine gute Validierungsstrategie. Eine bessere Validierungsstrategie ist die Kreuzvalidierung. Bei der Kreuzvalidierung wird die Gesamtmenge der Beispiele in eine vorgegebene Anzahl von Blöcken partitioniert. Anschließend werden wiederholt Lernläufe durchgeführt. In jedem Lernlauf wird ein anderer Block als Trainingsmenge verwendet, alle übrigen Blöcken dienen als Testmengen. Die Fehlerrate wird über alle Durchläufe gemittelt, um eine stabilere Einschätzung zukünftiger Fehler zu ermitteln.

Bei allen im folgenden Abschnitt dokumentierten Lernläufen wurde eine zehnfache Kreuzvalidierung angewendet.

5.4.3. Ergebnisse

Zunächst wurden die Lernverfahren auf alle Merkmale (131) und alle Verträge (217586) angewendet. Tabelle 5.3 gibt einen Überblick über die Lernergebnisse. Sie wurden für die drei beschriebenen Repräsentationen bei vier eingesetzten Lernverfahren erzielt. Anhand der Ergebnisse lassen sich folgende Beobachtungen machen:

- Die TF/IDF und die binäre Repräsentation erzielen bei allen Lernverfahren deutlich bessere Ergebnisse als die Repräsentation mit den originalen Merkmalen.
- Die Verfahren J4.8 und SVM liefern herausragend gute Ergebnisse sowohl für die TF/IDF als auch für die binäre Repräsentation. Die SVM erzielt auf der TF/IDF Repräsentation ein F-Measure von 97,95% und auf der binären Repräsentation ein F-Measure von 98,72%. Bei J4.8 wird ein F-Measure von 99,19% auf der TF/IDF und von 96,38% auf der binären Repräsentation erreicht.
- Die Lernergebnisse aller Verfahren zeigen nur geringfügige Unterschiede zwischen der binären und der TF/IDF Repräsentation. Bei Apriori beträgt das F-Measure für die TF/IDF-Darstellung 63,35% während es für die binäre Darstellung 62,96% beträgt. Bei der SVM und Naive Bayes schneidet die TF/IDF Darstellung sogar schlechter ab. Bei der SVM beträgt das F-Measure für die TF/IDF Darstellung 97,95%, dagegen beträgt es für die binäre Darstellung 98,72%.

In einer zweiten Versuchsreihe wurde der Frage nachgegangen, ob die Einführung der Stempelsteuer 1998 einen Einfluss auf die Änderungen der Verträge hatte. Die Stempelsteuer wird unter anderem auf Quittungen von Versicherungsprämien erhoben und hat

Apriori			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	93,48%	93,03%	94,3%
Precision	56,07%	53,5%	84,97%
Recall	72,8%	76,49%	18,39%
F-Measure	63,35%	62,96%	30,24%
J4.8			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	99,87%	99,52%	82,52%
Precision	98,61%	98,38%	24,17%
Recall	99,78%	95,32%	74,85%
F-Measure	99,19%	96,83%	36,54%
SVM			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	99,71%	99,82%	26,65%
Precision	97,06%	98,86%	8,73%
Recall	98,86%	98,58%	100%
F-Measure	97,95%	98,72%	16,06%
Naive Bayes			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	88,32%	89,66%	87,44%
Precision	37,82%	41,72%	32,08%
Recall	78,93%	84,37%	77,72%
F-Measure	51,14%	55,83%	45,41%

Tabelle 5.3.: Überblick über die Lernergebnisse

Apriori			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	93,48%	93,03%	94,3%
Precision	56,07%	53,5%	84,97%
Recall	72,8%	76,49%	18,39%
F-Measure	63,35%	62,96%	30,24%
J4.8			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	99,87%	99,8%	97,89%
Precision	98,29%	98,05%	96,21%
Recall	99,68%	98,74%	60,85%
F-Measure	98,98%	98,39%	74,55%
SVM			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	99,71%	99,82%	26,65%
Precision	97,06%	98,86%	8,73%
Recall	98,86%	98,58%	100%
F-Measure	97,95%	98,72%	16,06%
Naive Bayes			
	TF/IDF Merkmale	Binäre Merkmale	Originale Merkmale
Accuracy	92,23%	93,25%	88,04%
Precision	42,44%	47%	25,58%
Recall	75,11%	80,22%	71,19%
F-Measure	54,24%	59,27%	37,64%

Tabelle 5.4.: Überblick über die Lernergebnisse bei Ausschluss der 1998 zurückgekauften Verträge

dazu geführt, dass viele Kunden ihre Versicherungsverträge kündigten. Alle Verträge, die im Jahr 1998 zurückgekauft wurden, wurden daher vom Lernprozess ausgeschlossen. Tabelle 5.4 zeigt die Ergebnisse dieser Lernläufe. Man sieht, dass die Herausnahme der 1998 zurückgekauften Verträge keinen signifikanten Einfluss auf die Qualität der Lernergebnisse nimmt.

Wie bereits beschrieben, unterscheiden sich die Lernergebnisse aller Verfahren für die TF/IDF und die binäre Repräsentation nur geringfügig. Ein Grund dafür kann sein, dass die bei TF/IDF untersuchte Häufigkeit der Mermale nicht von dem binär erfassten Vorhandensein der Merkmale unterschieden werden kann. Um das zu überprüfen, wurde für jeden Vertrag bestimmt, in wievielen Attributen sich die TF/IDF Repräsentation von

Anzahl der Attribute	Anzahl der Verträge
1	64740
2	72689
3	10801
4	7552
5	1598
6	1103
7	1066
8	1277
9	386
10	80
11	27
12	4

Tabelle 5.5.: Ergebnis des Vergleichs zwischen der TF/IDF und der binären Repräsentation

der binären Repräsentation unterscheidet. Bei 161 323 Verträgen zeigt sich ein Unterschied zwischen der TF/IDF Repräsentation und der binären Repräsentation. Allerdings existieren nur wenige Verträge, bei denen sich die Häufigkeit und das Vorkommen bei mehr Attributen unterscheidet (siehe Tabelle 5.5). Dies ist eine mögliche Erklärung für den geringfügigen Unterschied der Lernergebnisse bei den beiden Repräsentationen.

Wie schon beschrieben zeigen die Verfahren J4.8 und SVM herausragende Ergebnisse sowohl für die TF/IDF als auch für die binäre Repräsentation. Eine Erklärung für beide Verfahren kann nicht gegeben werden, da für die meisten Lernverfahren keine Theorien zur Erkennung ihrer Anwendbarkeit auf einem Datensatz existieren.

Für die SVM und Textklassifikationssaufgaben wurde von Joachims [Joachims 2002] ein statistisches Lernmodell entwickelt. Im nächsten Kapitel wird dieses Modell zur Erklärung des guten Lernergebnisses der SVM auf der TF/IDF Repräsentation angewandt. Da dieses Modell auf der Basis der Häufigkeitsverteilung von Wörtern entwickelt wurde, kann es nicht auf die binäre Repräsentation angewendet werden. Weitere Experimente mit anderen Datensätzen sind notwendig, um die Überlegenheit der TF/IDF Repräsentation zu anderen Repräsentationen zu untersuchen.

6. Ein Erklärungsmodell für die Ergebnisse

In diesem Kapitel wird das gute Lernergebnis der SVM auf der TF/IDF Repräsentation anhand des statistischen Lernmodells von Joachims [JOACHIMS 2002] untersucht. Sein Modell beweist die Anwendbarkeit von SVMs auf Textklassifikationsaufgaben. Auf der Basis von statistischen Eigenschaften entwickelte Joachims TCat-Konzepte als Modell für Textklassifikationsaufgaben.

Nach der Erläuterung des statistischen Lernmodells erfolgt die Anwendung auf die Versicherungsdaten. Eine Anwendung des Modells kann jedoch erst nach erfolgter Transformation in TF/IDF Merkmale erfolgen. Zur Abschätzung der Eignung eines Datensatzes für eine TF/IDF Transformation wird eine Heuristik angegeben. Abschließend wird die Diplomarbeit zu einer verwandten Arbeit abgegrenzt.

6.1. Joachims statistisches Lernmodell für die Textklassifikation

Die Funktionsweise der SVM wurde in Kapitel 2 im Abschnitt 2.3.1 erläutert. Es wurde gesagt, dass die SVM genau die Hyperebene wählt, die alle positiven und negativen Beispiele mit maximalem Abstand trennt. Diese Hyperebene hat zu jedem Beispiel mindestens einen Abstand von δ. δ wird Rand (engl. *margin*) der Hyperebene genannt.

Das folgende Theorem zeigt, dass die Kombination von einem großen Rand mit einem niedrigen Trainingsfehler zu einer hohen Generalisierungsgenauigkeit führt. Für den erwarteten Fehler wird eine Schranke angegeben.

Theorem 1 (Schranke für den erwarteten Fehler)
Die erwartete Fehlerrate $\varepsilon(Err^n(h_{SVM}))$ einer SVM mit weicher Trennung basierend auf

n Traningsbeispielen mit $c \leq \kappa(\vec{x_i}, \vec{x_j}) \leq c + R^2$ für eine Konstante c, ist beschränkt durch

$$C \geq \frac{1}{\rho R^2} \quad : \varepsilon(Err^n(h_{SVM})) \quad \leq \frac{\rho\varepsilon\left(\frac{R^2}{\delta^2}\right) + C\rho R^2 \varepsilon\left(\sum_{i=1}^{n+1} \xi_i\right)}{n+1} \tag{6.1}$$

$$C < \frac{1}{\rho R^2} \quad : \varepsilon(Err^n(h_{SVM})) \quad \leq \frac{\rho\varepsilon\left(\frac{R^2}{\delta^2}\right) + \rho(CR^2+1)\varepsilon\left(\sum_{i=1}^{n+1} \xi_i\right)}{n+1} \tag{6.2}$$

Für unverzerrte Hyperebenen ist ρ gleich 1 und für stabile Hyperebenen ist ρ gleich 2.

Diese Schranke zeigt, dass die Schlüsselgrößen der Rand δ, die Länge der Dokumentvektoren R und der Trainingsfehler ξ sind.

Um Aussagen über Textklassifikationsaufgaben treffen zu können, führte Joachims die TCat-Konzepte als generelles Modell ein. Es abstrahiert von einzelnen Textklassifikationsaufgaben und beruht auf fünf statistischen Eigenschaften:

- Hochdimensionaler Merkmalsraum
  Textklassifikationsprobleme haben einen hochdimensionalen Merkmalsraum. Wenn jedes Wort, das in einem Dokument erscheint, als Merkmal betrachtet wird, haben Textklassifikationsprobleme bei einigen Tausend Dokumenten 10000 und mehr Dimensionen.

- Spärlich besetzte Dokumentvektoren
  Die Anzahl der möglichen Merkmale ist groß. Jedes Dokument enthält jedoch nur eine kleine Anzahl unterschiedlicher Worte. Das impliziert, dass die Dokumentvektoren spärlich besetzt sind.

- Heterogener Gebrauch von Worten
  Verwandte Dokumente enthalten Schlüsselwörter, aber es gibt kein Wort, das allen verwandten Dokumenten gemeinsam ist. Dokumente aus der gleichen Kategorie können aus verschiedenen Worten bestehen.

- Hoher Redundanzgrand
  Die meisten Dokumente enthalten mehr als ein Wort zur Bestimmung der Klassenzugehörigkeit. Entfernt man die charakteristischen Merkmale, können die verbliebenen Wörter immer noch zu einem gewissen Grad den Inhalt beschreiben.

- Zipf'sches Gesetz
  Das Zipf'sche Gesetz modelliert die Häufigkeitsverteilung von Worten in natürlichsprachlichen Texten [G.K.Zipf 1949]. Das Gesetz besagt: Werden Worte nach

ihrer Worthäufigkeit geordnet, dann gilt für die Termfrequenz tf_r des r-häufigsten Wortes:

$$tf_r = \frac{1}{r} \cdot tf_{r_{max}} \tag{6.3}$$

$tf_{r_{max}}$ ist die Termfrequenz des häufigsten Wortes.

Mandelbrotverteilungen [Mandelbrot 1959] zeigen eine noch genauere Anpassung. Sie werden durch folgende Formel angegeben:

$$tf_i = \frac{c}{(k+r)^{\phi}} \tag{6.4}$$

Die Gleichung beschreibt den Zusammenhang zwischen dem Häufigkeitsrang r und der Termhäufigkeit tf_r. Sie wird deshalb auch (generalisiertes) Zipf'sches Gesetz genannt.

Definition 5 (Homogene TCat-Konzepte)
Das TCat-Konzept

$$TCat([p_1 : n_1 : f_1], ..., [p_s : n_s : f_s])$$

beschreibt eine binäre Klassifikationsaufgabe mit s disjunkten Mengen von Merkmalen. Die i-te Menge enthält f_i Merkmale. Jedes positive Beispiel enthält p_i Merkmale aus der jeweiligen Menge und jedes negative Beispiel enthält n_i Merkmale. Das gleiche Merkmal kann mehrmals in einem Dokument vorkommen.

Joachims hat gezeigt, dass Klassifikationsaufgaben, generell seperarierbar sind, wenn sie sich als TCat-Konzepte lasssen. Für den Rand δ der trennenden Hyperebene hat er eine untere Schranke bewiesen.

Lemma 1 (Untere Schranke für den Rand von TCat-Konzepte)
Für TCat($[p_1 : n_1 : f_1], ..., [p_s : n_s : f_s]$)-Konzepte existiert immer eine Hyperebene durch den Ursprung. Der Rand δ dieser Hyperebene ist beschränkt durch

$$\delta^2 \geq \frac{ac - b^2}{a + 2b + c} \quad mit \quad \begin{aligned} a &= \sum_{i=1}^{s} \frac{p_i^2}{f_i} \\ b &= \sum_{i=1}^{s} \frac{p_i n_i}{f_i} \\ c &= \sum_{i=1}^{s} \frac{n_i^2}{f_i} \end{aligned}$$

Separierbarkeit impliziert, dass der Trainingsfehler null ist. Voraussetzung für die Anwendbarkeit von Theorem 1 ist eine Schranke für die maximale Euklidische Länge R der Dokumentvektoren. Die Euklidische Länge des Dokumentvektors eines Dokumente mit l Wörtern kann nicht größer als l sein. Für reale Dokumentvektoren ist diese Schranke nicht präzise genug. Das Zipf'sche Gesetz wird angewandt, um eine genauere Schranke für R angeben zu können.

Die Annahme, dass die Termfrequenzen in jedem Dokument dem generalisierten Zipf'schen Gesetz gehorchen, impliziert nicht, dass ein bestimmtes Wort mit einer gewissen Häufigkeit in jedem Dokument vertreten ist. Das Zipf'sche Gesetz sagt nur aus, dass das r-häufigste Wort mit einer gewissen Häufigkeit vorkommt. In unterschiedlichen Dokumenten kann das r-häufigste Wort verschieden sein. Das folgende Lemma verbindet die Länge der Dokumentvektoren mit dem Zipf'schen Gesetz.

Lemma 2 (Euklidische Länge der Dokumentvektoren)
Wenn die gerankten Termfrequenzen tf_r in einem Dokument mit l Termen dem generalisierten Zipf Gesetz gehorchen

$$tf_i = \frac{c}{(k+r)^{\phi}} \tag{6.5}$$

dann ist die quadratische Euklidische Länge des Dokumentenvektors $\vec{x}$ der Termfrequenzen beschränkt durch

$$\|\vec{x}\| \leq \sqrt{\sum_{r=1}^{d}(\frac{c}{(k+r)^{\phi}})^2} \quad \textit{wobei für } d \textit{ gilt} \quad \sum_{r=1}^{d}\frac{c}{(k+r)^{\phi}} = l \tag{6.6}$$

Die Tatsache, dass die Termfrequenzen dem Zipf'schen Gesetz gehorchen, hat einen starken Einfluss auf die Lernbarkeit von Textklassifikationsaufgaben. Das Zipf'sche Gesetz impliziert, dass die meisten Worte sich nicht oft wiederholen und dass die Anzahl der unterschiedlichen Worte d hoch ist. Wenn das Zipf'sche Gesetz nicht gelten würde, könnte sich ein einzelnes Wort l-mal wiederholen, sodass der Dokumentvektor eine Euklidische Länge von i hätte. Mit dem Zipf'sche Gesetz erhält man dagegen vergleichbar kurze Dokumentvektoren und einen niedrigen Wert für R^2 in der Schranke für die erwartete Generalisierungsperformanz.

Die Kombination von Lemma 1 und Lemma 2 mit Theorem 1 führt zu folgendem Ergebnis

Theorem 2 (Lernbarkeit von TCat-Konzepten)
Für TCat($[p_1 : n_1 : f_1], ..., [p_s : n_s : f_s]$)-Konzepte und Dokumente mit l Worten, deren Verteilung dem generalisierten Zipf Gesetz $tf_r = \frac{c}{(r+k)^{\phi}}$ gehorchen, ist der erwartete

Generalierungsfehler einer (unverzerrten) SVM nach dem Training auf n Beispielen beschränkt durch

$$\varepsilon(Err^n(h_{SVM})) \leq \rho \frac{R^2}{n+1} \frac{a+2b+c}{ac-b^2} \quad mit \quad \begin{aligned} a &= \sum_{i=1}^{s} \frac{p_i^2}{f_i} \\ b &= \sum_{i=1}^{s} \frac{p_i n_i}{f_i} \\ c &= \sum_{i=1}^{s} \frac{n_i^2}{f_i} \\ R^2 &= \sum_{r=1}^{d} \left(\frac{c}{(r+k)^{\phi}} \right)^2 \end{aligned}$$

wenn nicht $\forall_{i=1}^{s} : p_i = n_i$ ist. d wird gewählt, so dass $\sum_{r=1}^{d} \frac{c}{(r+k)^{\phi}} = l$ ist. Für unverzerrte SVMs ist ρ gleich 1 und für verzerrte SVMs ist ρ gleich 2.

6.2. Anwendung von Joachims Modell auf die Versicherungsdaten

Wie bereits beschrieben beruht das Modell von Joachims auf statistischen Eigenschaften. Um es auf die Versicherungsdaten anwenden zu können, müssen ähnliche statistische Eigenschaften auch für diese Daten gelten. Bereits in Kapitel 5 wurde im Abschnitt 5.2.1 festgestellt, dass sich die Versicherungsdaten durch einen hochdimensionalen Merkmalsraum und spärlich besetzte Eigenschaftsvektoren auszeichnen.

Voraussetzung für die Modellierung der Daten als TCat-Konzept ist die Einteilung der Attribute entsprechend ihrer Termfrequenz (*termfrequency*) in hoch- (*high frequency*), mittel- (*medium frequency*) und niedrigfrequente (*low frequency*) Attribute. Die Termfrequenz für ein Attribut a_i ergibt sich als Summe der Termfrequenzen des Attributs in allen Verträge (217586) und berechnet sich folgendermaßen:

$$tf(a_i) = \sum_{j=1}^{217586} tf(a_i, c_j) \tag{6.7}$$

Die Attribute werden nach der Termfrequenz absteigend sortiert und erhalten eine Rangzahl. Die Rangzahl 1 erhält das Attribut mit der höchsten Termfrequenz. Es handelt sich dabei um das Attribut MUT1003 (Fortschreibung Überschuss vorschüssig) mit einer Termfrequenz von 532691. Die Attribute MUT24 (Beginn des Rentenbezugs bei

Ae-Versicherungen), MUT52 (Aufhebung Neueintritt), MUT56 (Invaliderklärung kleine Invalidität) und MUT1047 (Ablauf Risikozuschlag) treten dagegen nur einmal auf und belegen die hintersten Ränge. In Abbildung 6.1 ist der Rang der Attribute gegen die Termfrequenz aufgetragen. Aus der Verteilung der Termfrequenzen ergibt sich eine Einteilung in hoch-, mittel- und niedrigfrequente Attribute. Hochfrequente Attribute haben eine Termfrequenz von mehr 100 000. Bei mittelfrequenten Attributen beträgt die Termfrequenz zwischen 100 und 100 000. Attribute mit einer Termfrequenz von weniger als 100 sind niedrig frequente Attribute. Die Attribute in den drei Kategorien müssen

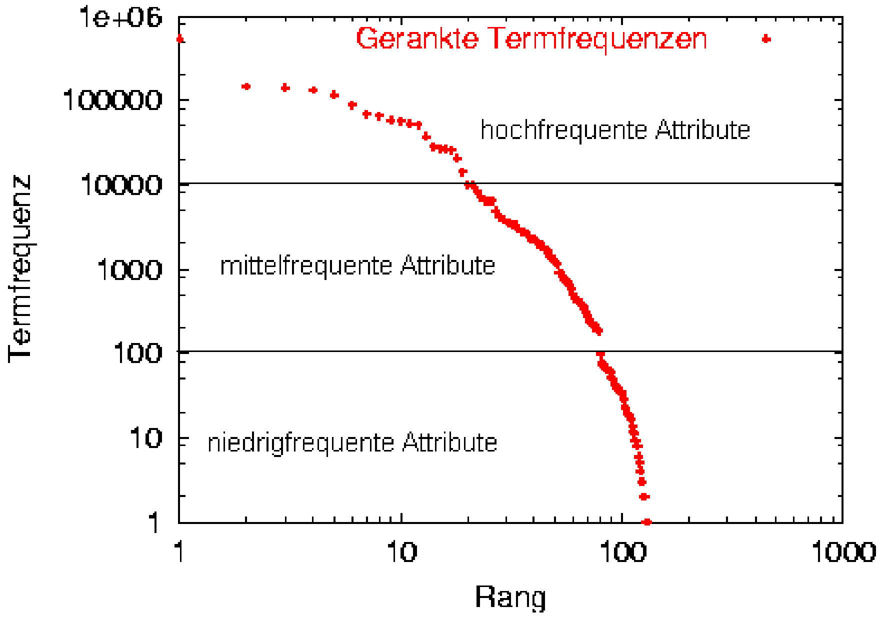


Abbildung 6.1.: Einteilung der Attribute in hoch-, mittel- und niedrigfrequente Attribute anhand ihrer Termfrequenzen

nun jeweils in disjunkte Mengen positiver und negativer Indikatoren sowie irrelevanter Merkmale partitioniert werden. Die Einteilung erfolgt anhand der Odds Ratio-Werte der Attribute. Die Odds Ratio ist wie folgt definiert:

Definition 6 (Odds Ratio)
Für ein Ereignis E mit der Wahrscheinlichkeit $P(E)$ ist die Odds definiert als

$$\frac{Pr(E)}{1 - Pr(E)}. \tag{6.8}$$

Für zwei Ereignisse E_1 und E_2 mit den Wahrscheinlichkeiten $P(E_1)$ und $P(E_2)$ ist die Odds Ratio das Verhältnis der Odds von E_1 zur Odds von E_2. Die Odds Ratio ist somit *definiert als:*

$$OR = \frac{\frac{Pr(E_1)}{1-Pr(E_1)}}{\frac{Pr(E_2)}{1-Pr(E_2)}} \tag{6.9}$$

Für die Versicherungsdaten bezeichnet E_1 das Ereignis, dass in einem zurückgekauften Vertrag das Attribut a_i vorkommt. E_2 ist das Ereignis, dass in einem fortgesetzten Vertrag das Attribut a_i vorkommt. Die Odds von E_1 berechnet sich folgerndermaßen:

$$\frac{Pr(E_1)}{1-Pr(E_1)} = \frac{\|v \in V \mid Rueckkauf = 1\, und\, tf(a_i) > 0\|}{\|v \in V \mid Rueckkauf = 1\, und\, tf(a_i) = 0\|} \tag{6.10}$$

Die Odds von E_2 ist in diesem Fall

$$\frac{Pr(E_2)}{1-Pr(E_2)} = \frac{\|v \in V \mid Rueckkauf = 0\, und\, tf(a_i) > 0\|}{\|v \in V \mid Rueckkauf = 0\, und\, tf(a_i) = 0\|} \tag{6.11}$$

Attribute mit einem Odds Ratio Wert größer als 1 sind positive Indikatoren, Attribute mit einem Odds Ratio Wert kleiner als 1 sind negative Indikatoren und Attribute mit einer Odds Ratio von 1 sind irrelevante Merkmale.

Die Berechnung der Odds Ratio Werte ergibt, dass für die Gruppe der hochfrequenten Attribute keine positiven jedoch fünf negative Indikatoren existieren. In der Gruppe der mittelfrequenten Attribute gibt es 29 positive und 45 negative Indikatoren. 13 positive und 39 negative Indikatoren treten bei den niedrigfrequenten Attributen auf. Es gibt keine irrelevanten Merkmale.

Eine Übersicht über die Indikatoren gibt Abbildung 6.2.

Um von den unterschiedlichen Eigenschaften der einzelnen Verträge zu abstrahieren, wird ein typischer Vertrag betrachtet. Ein durchschnittlicher Vertrag ist durch 8 Merkmale gekennzeicht. 25% dieser 8 Merkmale kommen bei positiven Beispielen aus der Menge der hochfrequenten negativen Indikatoren. Jedoch tritt keines dieser Merkmale in einem durchschnittlichen negativen Vertrag auf.

Tabelle 6.1 gibt einen Überblick über die relativen Häufigkeiten der anderen Merkmale. Bei Anwendung der Prozentzahlen auf die durchschnittliche Anzahl Merkmale kann diese Tabelle direkt in das folgende TCat-Konzept übersetzt werden:

```
TCat ( [2 : 5 : 5],                      # hochfrequent
       [6 : 2 : 29], [0 : 1 : 45],       # mittelfrequent
       [0 : 0 : 13], [0 : 0 : 39],       # niederigfrequent
     )
```

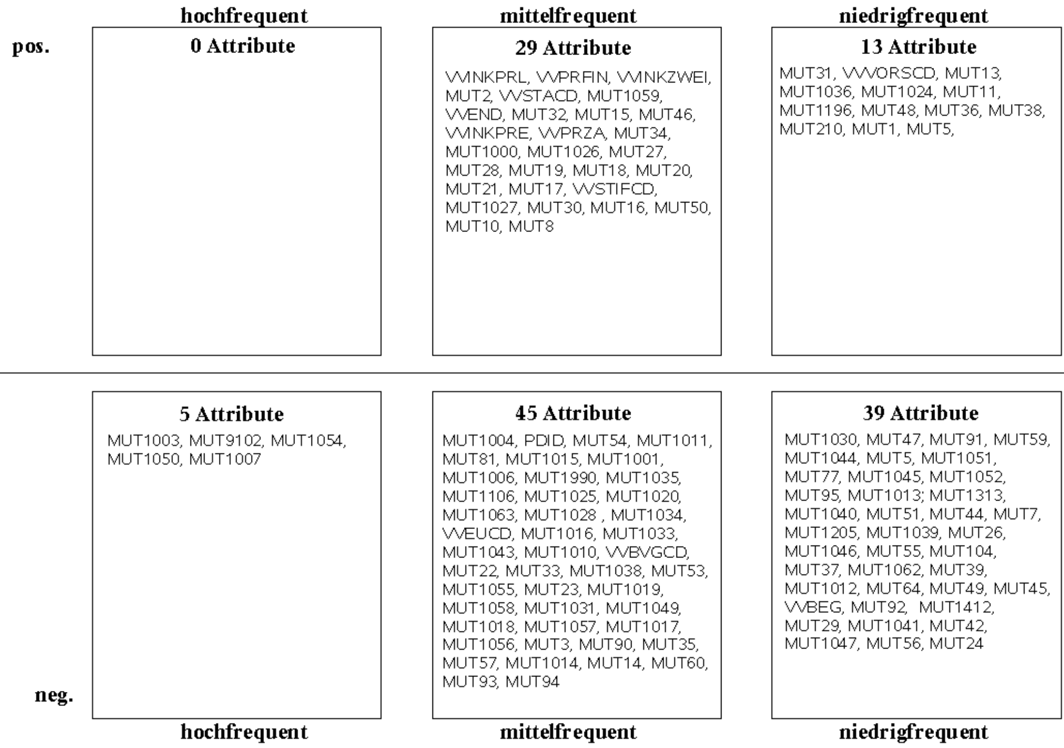


Abbildung 6.2.: Indikatoren

Das Lernbarkeitstheorem der TCat-Konzepte beweist, dass der erwartete Generalisierungsfehler einer SVM nach dem Training auf n Beispielen beschränkt ist durch:

$$\frac{R^2}{n+1}\frac{a+2b+c}{ac-b^2} \tag{6.12}$$

Dabei ist R^2 die maximale Euklidische Länge eines Merkmalsvektors in den Trainingsdaten. Die Faktoren a, b, c können aus dem TCat-Konzept berechnet werden:

$$a = \sum_{i=1}^{5} \frac{p_i^2}{f_i} = 2.041 \quad b = \sum_{i=1}^{5} \frac{p_i n_i}{f_i} = 2.207 \quad c = \sum_{i=1}^{5} \frac{n_i^2}{f_i} = 5.16$$

Gehorchen die Termfrequenzen dem generalisierten Zipf'schen Gesetz kann die maximale Euklidische Länge R^2 abgeschätzt werden durch

$$R^2 = \sum_{r=1}^{d} \left(\frac{c}{(r+k)^2} \right)^2 \tag{6.13}$$

In Abbildung 6.4 ist die Termfrequenz gegen den Rang aufgetragen. Die gestrichelte

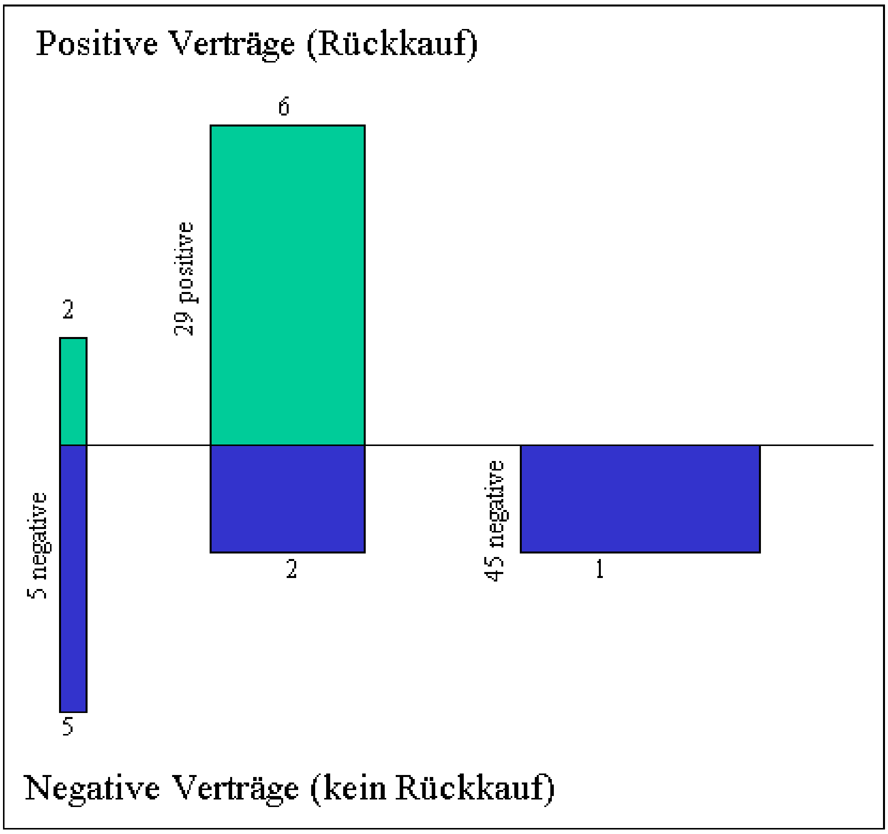


Abbildung 6.3.: TCat-Konzept für die Vertragsdaten

	hochfrequent	mittelfrequent		niedrigfrequent	
	5 neg.	29 pos.	45 neg.	13 pos.	39 neg.
pos. Vertrag	25%	75%	0%	0%	0%
neg. Vertrag	62.5%	25%	12.5%	0%	0%

Tabelle 6.1.: Zusammensetzung eines durchschnittlichen positiven und negativen Vertrages

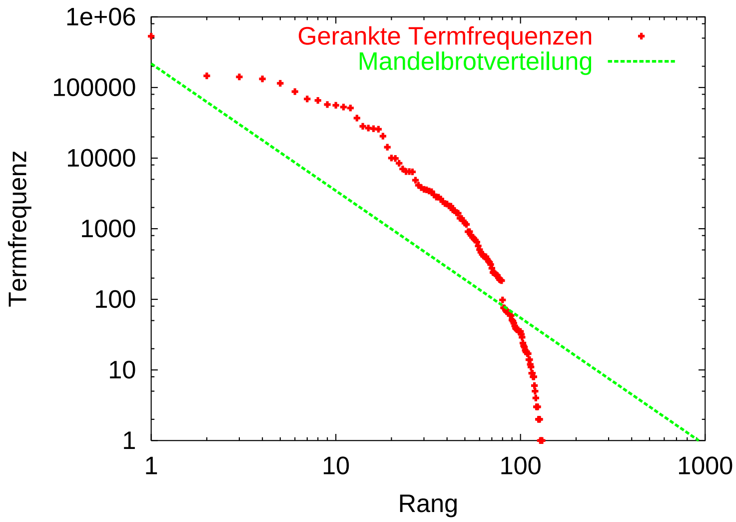


Abbildung 6.4.: Mandelbrotverteilung

Line ist die Anpassung der Mandelbrotverteilung für die Parameter $k = 0$ und $\phi = 1.7$. Unter der Annahme, dass die Verträge hinreichend homogen sind, trifft das generalisierte Zipf'sche Gesetz auch auf individuelle Verträge zu. Die resultierende Schranke für R^2 ist:

$$\sum_{i=1}^{4} TF_i^2 = 36 \geq R^2 \tag{6.14}$$

Setzt man diesen Wert in die Gleichung ein, erhält man folgende Schranke für den erwarteten Fehler:

$$\varepsilon(Err^n(h_{SVM})) \leq \frac{7,3665 * 36}{n+1} \leq \frac{266}{n+1} \tag{6.15}$$

Nach Joachims Modell beträgt der erwarteten Generalisierungsfehler nach einem Training auf 1000 Beispielen weniger als 26,6%. Damit ist gezeigt, dass der Datensatz in der TF/IDF Repräsentation von einer SVM getrennt werden kann. Das Modell liefert eine Erklärung für das erzielte Lernergebnis.

6.3. Abschätzung der Eignung eines Datensatzes für eine TF/IDF Transformation

Mit dem Modell von Joachims lässt sich die Lernbarkeit der Versicherungsdaten in der TF/IDF Repräsentation durch die SVM beweisen. Zur Abschätzung der Lernbarkeit anderer Datensätze in einer TF/IDF Repräsentation wird in dieser Arbeit die Euklidische Länge der Merkmalsvektoren verwendet. Ihr Einfluss auf die Höhe des erwarteten Generalisierungsfehlers hilft geeignete Datensätze für die TF/IDF Repräsentationen auszuwählen.

Für einen Datensatz mit n Attributen $a_1, \ldots, a_n$ sei m die maximale Termfrequenz eines Attributes a_i in der TF/IDF Repräsentation. Die Euklidische Länge beträgt im schlechtesten Fall (worst case) $\sqrt{n} \cdot m$, wenn alle n Attribute eine Termfrequenz von m haben. Ein Datensatz eignet sich zur Transformation in TF/IDF Merkmale, wenn die tatsächliche Euklidische Länge niedriger ist als dieser schlechteste Fall.

Zur Realisierung dieser Abschätzung auf einem Datensatzes wird jede Zeitreihe z_j des Datensatzes als Vektor mit n Attributen $a_1, \ldots, a_n$ aufgefasst. Für jede Zeitreihe z_j kann parallel „on the fly" die Termfrequenz für alle n Attribute berechnet werden. Es kann in einem Datenbankdurchlauf die Zeitreihe z_j mit der maximalen Euklidischen Länge bestimmt werden.

$$\hat{R} = \max_{z_j} \left(\sqrt{\sum_{i=1}^{n} tf(a_i, c_j)^2} \right) \tag{6.16}$$

Für den Fall, dass $\hat{R} \leq \sqrt{n} \cdot m$ ist, ist der Datensatz für die Transformation in TF/IDF Merkmale geeignet.

Diese Vorgehensweise zur Abschätzung wird anhand der Versicherungsdaten geprüft. Für die Versicherungsdaten ist $n = 13$. Die maximale Termfrequenz eines Attributes beträgt 15, folglich ist $m = 15$. Für die Euklidische Länge ergibt sich im schlechtesten Fall ein Wert von $R = \sqrt{13} \cdot 15 = 54{,}08$. Die tatsächliche maximale Euklidische Länge $\hat{R}$ beträgt $22{,}91$. Dieser Wert ist niedriger als der zu erwartende schlechteste Fall. Damit kann die Eignung des Datensatzes zur TF/IDF Transformation angenommen werden.

6.4. Verwandte Arbeiten

[Domeniconi et al. 2002] setzen die SVM zur Vorhersage von kritischen Ereignissen in einem Computernetzwerk ein. Die Autoren begründen die Wahl der SVM als Lernverfahren mit Hilfe von Joachims Theorie. Mit der Übereinstimmung der statistischen Eigenschaften ihrer Daten zu den von Joachims geforderten Eigenschaften begründen sie die Wahl der SVM.

Die vorliegende Diplomarbeit geht nicht nur auf die statistichen Eigenschaften der untersuchten Daten ein, sondern berechnet auch das von Joachims entwickelte TCat-Konzept. Ein Schwerpunkt der Arbeit liegt auf der Vorverarbeitung, mit besonderer Berücksichtigung der Merkmalskonstruktion.

Bei [Domeniconi et al. 2002] werden Merkmale folgendermaßen erzeugt: Die Folge der Ereignisse in dem Computernetzwerk wird als Sequenz betrachtet. Die Sequenz wird in n Fenster partitioniert. Für jeden der m Ereignistypen wird ihre Häufigkeit innerhalb der Fenster bestimmt. Auf diese Weise wird eine $m \times n$-Matrix erzeugt. Die Bestimmung der Häufigkeit eines Ereignistyps in einem Fenster kann als Häufigkeitskodierung angesehen werden. Die vorliegende Diplomarbeit berücksichtigt zusätzlich die Analogie zur Dokumentfrequenz.

[Domeniconi et al. 2002] betonen den Vorteil der SVM für ihre Lernaufgabe ausschließlich über die Erfüllung der statistischen Eigenschaften. Im Gegensatz dazu zeigt die vorliegende Arbeit, dass die TF/IDF Transformation der Merkmale vor Anwendung der SVM zu einer Verbesserung der Lernergebnisse führt.

7. Zusammenfassung und Ausblick

Die Qualität des Lernergebnisses bei der Wissensentdeckung ist abhängig von einer geeigneten Repräsentation der zu analysierenden Daten. Die Transformation der Rohdaten in eine geeignete Repräsentation ist Aufgabe der Vorverarbeitung. Ein zunächst ungeeignet erscheinender Merkmalsraum kann durch Merkmalsauswahl und Merkmalsgenerierung so bearbeitet werden, dass er für die Wissensentdeckung verwendet werden kann. Bei der Merkmalsauswahl werden aus dem gesamten Merkmalsraum die besten Merkmale ausgewählt. Dadurch reduziert sich ein hochdimensionaler in einen niegrigdimensionalen Merkmalsraum. Dieser ist für die Wissensentdeckung leichter zugänglich. Bei der Merkmalsgenerierung werden aus den gegebenen Merkmalen für das Lernen besser geeignete Merkmale erzeugt.

In dieser Diplomarbeit wurden zunächst Ansätze zur Merkmalsauswahl, zur Merkmalsgenerierung und zur Kombination beider Methoden vorgestellt. Danach wurde TF/IDF zur häufigkeitsbasierten Merkmalsgenerierung vorgeschlagen. TF/IDF ist ein aus dem Information Retrieval stammendes Maß zur Gewichtung von Worten. Im Maschinellen Lernen kommt es vor allem in der Textklassifikation zur Anwendung. Es wurden zwei Möglichkeiten vorgestellt, dieses Maß zur Erzeugung von neuen Merkmalen aus temporalen Daten wie Zeit- oder Ereignissequenzen einzusetzen. Eine Möglichkeit besteht darin, die Änderung von Merkmalswerten zu betrachten, eine andere, das Vorkommen von Merkmalswerten zu untersuchen. Beide Möglichkeiten wurden als Operatoren im Mining Mart System realisiert. Die Anwendbarkeit von TF/IDF zur Merkmalsgenerierung wurde an einem Datensatz der Versicherungsanstalt Swiss Life getestet. Dieser Datensatz umfasst Kunden- und Versicherungsvertragsdaten in anonymisierter Form. Die Aufgabe war, den Rückkauf eines Vertrages vorhersagen zu können. Von einem Rückkauf spricht man, wenn der Versicherungsnehmer die Versicherung durch Rücktritt oder Kündigung vorzeitig beendet. In einem solchen Fall ist die Versicherungsgesellschaft verpflichtet, dem Versicherungsnehmer den aktuellen Zeitwert der Versicherung auszuzahlen. In der Annahme, dass häufige Änderungen eines Vertrages Ausdruck von Unzufriedenheit des Kunden sind, wurde TF/IDF zur Erzeugung von Merkmalen verwendet, die Änderungen von Merkmalswerten und Vorkommen von Änderungsgründen in den Versicherungsverträgen erfassen. Zur Beurteilung der TF/IDF Repräsentation wurden zwei Vergleichsrepräsentationen erzeugt. Die eine Vergleichsrepräsentation wurde ohne Berücksichtigung der zeitlichen Dimension der Daten erstellt, die andere erfasst ausschließlich binäre Merkmale für das Vorhandsein von Änderungen bzw. Änderungsgründen. Auf die drei Repräsentationen wurden vier unterschiedliche Lernverfahren angewandt.

Alle Lernverfahren erzielten auf der TF/IDF Repräsentation ein deutlich besseres Lernergebnis als auf der Repräsentation ohne Berücksichtigung der zeitlichen Dimension. Die binäre und die TF/IDF Repräsentation erzielten bei J4.8 und der SVM herausragend gute Ergebnisse. Thorsten Joachims statistisches Modell für die Textklassifikation wurde zur Erklärung des guten Ergebnisses der SVM auf der TF/IDF Repräsentation verwendet. Die Versicherungsdaten wurden als TCat-Konzept dargestellt. Für TCat-Konzepte hat Joachims die Lernbarkeit durch die SVM bewiesen und eine Schranke für den erwarteten Generalisierungsfehler der SVM angegeben. Durch Anwendung seiner Theorie konnte das gute Lernergebnis der SVM auf der TF/IDF Repräsentation bestätigt werden. Das Modell von Thorsten Joachims kann ausschließlich auf bereits transformierte Merkmale angewendet werden. Zur Abschätzung der Eignung anderer Datensätze für eine TF/IDF Repräsentation wurde eine Heuristik auf Basis der Euklidischen Länge der Merkmalsvektoren angegeben.

In der vorliegenden Diplomarbeit wurden Experimente mit einem einzelnen Datensatz durchgeführt. Zur weiteren Überprüfung des Vorteils einer TF/IDF Repräsentation sind zusätzliche Experimente mit anderen Datensätzen notwendig.

Für den TF/IDF Ansatz sind Erweiterungen denkbar, die in dieser Diplomarbeit nicht untersucht werden konnten:

- Die Anwendung von TF/IDF zur Lösung von anderen Lernaufgaben,
- die Verwendung von Zeitfenstern bei Zeitreihen oder Ereignissequenzen und
- die Berücksichtigung der Änderungsrichtung bei numerischen Attributen

Hier ergeben sich weitere interessante Forschungsmöglichkeiten.

Literaturverzeichnis

[AGRAWAL et al. 1993] AGRAWAL, R., T. IMIELINSKI und A. SWAMI (1993). *Mining Association Rules between Sets of Items in Large Databases*. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, S. 207–216, Washington, D. C.

[AGRAWAL und SRIKANT 1994a] AGRAWAL, R. und R. SRIKANT (1994a). *Fast Algorithms for Mining Association Rules in Large Data Bases*. In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*, S. 478–499, Santiago, Chile.

[AGRAWAL et al. 1996] AGRAWAL, RAKESH, H. MANNILA, R. SRIKANT, H. TOIVONEN und A. I. VERKAMO (1996). *Fast Discovery of Association Rules*. In: FAYYAD, USAMA M., G. PIATETSKY-SHAPIRO, P. SMYTH und R. UTHURUSAMY, Hrsg.: *Advances in Knowledge Discovery and Data Mining*, Kap. 12, S. 307–328. AAAI Press/The MIT Press, Cambridge Massachusetts, London England.

[AGRAWAL und SRIKANT 1994b] AGRAWAL, RAKESH und R. SRIKANT (1994b). *Fast Algorithms for Mining Association Rules*. In: *Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile*. IBM Almaden Research Center.

[AHA und L. 1996] AHA, DAVID W. und B. R. L. (1996). *A Comparative Evaluation of Sequential Feature Selection Algorithms*. In: FISHER, DOUG und H.-J. LENZ, Hrsg.: *Learning from Data*, Kap. 4, S. 199–206. Springer, New York.

[ALLEN 1984] ALLEN, J. F. (1984). *Towards a General Theory of Action and Time*. Artificial Intelligence, 23:123–154.

[BALA et al. 1995] BALA, JERZY, J. HUANG, H. VAFAIE, K. DEJONG und H. WECHSLER (1995). *Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification*. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, S. 719 – 724, San Francisco, CA, USA. Morgan Kaufmann.

[BAUSCHULTE et al. 2002] BAUSCHULTE, FABIAN, I. BECKMANN, S. HAUSTEIN, C. HUEPPE, Z. EL JERROUDI, H. KOEPCKE, P. LOOK, K. MORIK, B. SHULIMOVICH, K. UNTERSTEIN und D. WIESE (2002). *PG-402 Endbericht Wissensmanagement*. Technischer Bericht, Fachbereich Informatik, Universität Dortmund.

[Bensusan und Kuscu 1996] Bensusan, Hilan und I. Kuscu (1996). *Constructive Induction using Genetic Programming*. In: Fogarty, T. und G. Venturini, Hrsg.: *ICML'96, Evolutionary computing and Machine Learning Workshop*. Morgen Kaufmann.

[Bi et al. 2001] Bi, Zhiqiang, C. Faloutsos und F. Korn (2001). *The DGX Distribution for Mining Massive, Skewed Data*. In: *7th International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

[Bloedorn und Michalski 1998] Bloedorn, Eric und R. Michalski (1998). *Data-driven Constructive Induction: Methodology and Applications*. In: Liu, Huan und H. Motoda, Hrsg.: *Feature Extraction, Construction, and Selection – A Data Mining Perpective*, Kap. 4, S. 51 – 68. Kluwer.

[Blum und Langley 1997] Blum, Avrim L. und P. Langley (1997). *Selection of Relevant Features and Examples in Machine Learning*. Artificial Intelligence, S. 245–271.

[Burges 1998] Burges, C. (1998). *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 2(2):121–167.

[Chang und Lippmann 1991] Chang, Eric I. und R. D. Lippmann (1991). *Using Genetic Algorithms to Improve Pattern Classification Performance*. In: Lippmann, R. P., J. Moody und D. S. Touretzky, Hrsg.: *Advances in Neural Information Processing Systems*, Bd. 3, S. 797–803. Morgan Kaufmann.

[Christianini und Shawe-Taylor 2000] Christianini, N. und J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.

[Domeniconi et al. 2002] Domeniconi, Carlotta, C. shing Perng, R. Vilalta und S. Ma (2002). *A Classification Approach for Prediction of Target Events in Temporal Sequences*. In: Elomaa, Tapio, H. Mannoila und H. Toivonen, Hrsg.: *Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Artificial Intelligence. Springer.

[Fayyad et al. 1996] Fayyad, U. M., G. Piatetsky-Shapiro und P. Smyth (1996). *From Data Mining to Knowledge Discovery: An Overview*. In: Fayyad, U. M., G. Piatetsky-Shapiro, P. Smyth und R. Uthurusamy, Hrsg.: *Advances in Knowledge Discovery and Data Mining*, Kap. 1, S. 1–34. AAAI/MIT Press.

[Fischer et al. 2002] Fischer, Simon, R. Klinkenberg, I. Mierswa und O. Ritthoff (2002). Yale*: Yet Another Learning Environment* – Tutorial. Technischer Bericht CI-136/02, Collaborative Research Center 531, University of Dortmund, Dortmund, Germany. ISSN 1433-3325.

[Fisseler 2003] Fisseler, Jens (2003). *Anwendung eines Data Mining-Verfahrens auf Versicherungsdaten*. Diplomarbeit, Fachbereich Informatik, Universität Dortmund.

[Freitas 2001] Freitas, A. (2001). *Understanding the crucial role of attribute interaction in data mining*. Artificial Intelligence Review, 16(3):177–199.

[G.K.Zipf 1949] G.K.Zipf (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley.

[Höppner 2002] Höppner, Frank (2002). *Discovery of Core Episodes from Sequences*. In: Hand, David J., N. M. Adams und R. J. Bolton, Hrsg.: *Pattern Detection and Discovery*, Bd. 2447 d. Reihe *Lecture notes in computer science*, S. 1–12, London, UK. ESF Exploratory Workshop, Springer.

[James 1985] James, M. (1985). *Classication Algorithms*. Wiley.

[Joachims 2001] Joachims, Thorsten (2001). *The Maximum-Margin Approach to Learning Text Classifiers: Methods, Theory, and Algorithms*. Doktorarbeit, Fachbereich Informatik, Universität Dortmund.

[Joachims 2002] Joachims, Thorsten (2002). *Learning to Classify Text using Support Vector Machines*, Bd. 668 d. Reihe *Kluwer International Series in Engineering and Computer Science*. Kluwer.

[Kietz et al. 2000] Kietz, Jörg-Uwe, A. Vaduva und R. Zücker (2000). *Mining Mart: Combining Case-Based-Reasoning and Multi-Strategy Learning into a Framework to reuse KDD-Application*. In: Michalki, R.S. und P. Brazdil, Hrsg.: *Proceedings of the fifth International Workshop on Multistrategy Learning (MSL2000)*, Guimares, Portugal.

[Kietz et al. 2001] Kietz, Jörg-Uwe, A. Vaduva und R. Zücker (2001). *Mining-Mart: Metadata-Driven Preprocessing*. In: *Proceedings of the ECML/PKDD Workshop on Database Support for KDD*.

[Lavrac et al. 1998] Lavrac, Nada, D. Gamberger und P. Turney (1998). *A relevancy filter for constructive induction*. IEEE Intelligent Systems, 13(2):50–56.

[Lewis 1995] Lewis, D. (1995). *Evaluating and Optimizing Autonomous Text Classification Systems*. In: *Proceedings of SIGIR 95*, S. 246–254.

[Liu et al. 1998] Liu, Bing, W. Hsu und Y. Ma (1998). *Integrating Classification and Association Rule Mining*. In: *Knowledge Discovery and Data Mining*, S. 80–86.

[Liu und Motoda 1998] Liu, H. und H. Motoda (1998). *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Kluwer.

[Mandelbrot 1959] Mandelbrot, Benoit (1959). *A Note on a Class of Skew Distribution Functions: Analysis and Critique of a Paper by H.A.Simon*. Informationi and Control, 2:90 – 99.

[Markovitch und Rosenstein 2002] Markovitch, Shaul und D. Rosenstein (2002). *Feature generation using general constructor functions*. Machine Learning, 49(1):59–98.

[Michalski 1983] Michalski, Ryszard S. (1983). *A Theory and Methodology of Inductive Learning*. In: Michalski, R. S., J. G. Carbonell und T. M. Mitchell, Hrsg.: *Machine Learning — An Artificial Intelligence Approach*, Bd. 1, Kap. 4, S. 83–135. Morgan Kaufmann, Palo Alto, CA.

[Moore et al. 2001] Moore, J. H., J. S. Parker und L. W. Hahn (2001). *Symbolic discriminant analysis for mining gene expression patterns*. In: De Raedt, Luc und P. A. Flach, Hrsg.: *Machine Learning: EMCL 2001: 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings*, Bd. Volume 2167 d. Reihe *Lecture Notes in Computer Science*, S. 372–381. Springer-Verlag Heidelberg.

[Morik und Scholz 2002] Morik, Katharina und M. Scholz (2002). *The Mining-Mart Approach*. In: *Workshop Management des Wandels der 32. GI Jahrestagung*.

[Morik und Scholz 2003] Morik, Katharina und M. Scholz (2003). *The MiningMart Approach to Knowledge Discovery in Databases*. In: Zhong, Ning und J. Liu, Hrsg.: *Intelligent Technologies for Information Analysis*. Springer-Verlag. to appear.

[Pazzani 1998] Pazzani, Michael J. (1998). *Constructive Induction Of Cartesian Product Attributes*. In: Liu, Huan und H. Motoda, Hrsg.: *Feature Extraction, Construction and Selection: a Data Mining Perspective*, S. 341–354. Kluwer.

[Punch et al. 1993] Punch, W. F., E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland und R. Enbody (1993). *Further Research on Feature Selection and Classification Using Genetic Algorithms*. In: Forrest, Stephanie, Hrsg.: *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, S. 557–564, San Mateo, CA. Morgan Kaufmann.

[Pyle 1999] Pyle, Dorian (1999). *Data Preparation for Data Mining*. Morgan Kaufmann Publishers.

[Quinlan 1993] Quinlan, John Ross (1993). *C4.5: Programs for Machine Learning*. Machine Learning. Morgan Kaufmann, San Mateo, CA.

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

[Ritthoff et al. 2002] Ritthoff, Oliver, R. Klinkenberg, S. Fischer und I. Mierswa (2002). *A Hybrid Approach to Feature Selection and Generation Using an Evolutionary Algorithm*. In: Bullinaria, John A., Hrsg.: *Proceedings of the 2002 U.K. Workshop on Computational Intelligence (UKCI-02)*, S. 147–154, Birmingham, UK. University of Birmingham.

[Ritthoff et al. 2001] Ritthoff, Oliver, R. Klinkenberg, S. Fischer, I. Mierswa und S. Felske (2001). Yale*: Yet Another Machine Learning Environment*. In: Klinkenberg, Ralf, S. Rüping, A. Fick, N. Henze, C. Herzog, R. Molitor und O. Schröder, Hrsg.: *LLWA 01 – Tagungsband der GI-Workshop-Woche* Lernen – Lehren – Wissen – Adaptivität, Nr. Nr. 763 in *Forschungsberichte des Fachbereichs Informatik, Universität Dortmund*, S. 84–92, Dortmund, Germany. ISSN 0933-6192.

[Salton und Buckley 1988] Salton, G. und C. Buckley (1988). *Term Weighting Approaches in Automatic Text Retrieval*. Information Processing and Management, 24(5):513–523.

[Staudt et al. 1998] Staudt, Martin, J.-U. Kietz und U. Reimer (1998). *A Data Mining Support Environment and its Application on Insurance Data*. In: *Knowledge Discovery and Data Mining*, S. 105–111.

[T. Baeck und Michalewicz 2000] T. Baeck, D. B. Fogel und T. Michalewicz (2000). *Evolutionary Computation 1, Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK.

[Vafaie und Jong 1993] Vafaie, H. und K. D. Jong (1993). *Robust Feature Selection Algorithms*. In: *Proceedings of the Fifth Conference on Tools for Artificial Intelligence*, S. 356–363. IEEE Computer Society Press.

[Vafaie und Jong 1998] Vafaie, Haleh und K. D. Jong (1998). *Evolutionary Feature Space Transforamtion*. In: Liu, Huan und H. Motoda, Hrsg.: *Feature Extraction, Construction, and Selection – A Data Mining Perpective*, Kap. 20, S. 307 – 323. Kluwer.

[Vapnik 1982] Vapnik, V. (1982). *Estimation of Dependencies Based on Empirical Data*. Springer.

[Vapnik 1998] Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, Chichester, GB.

[Witten und Frank 2000] Witten, Ian und E. Frank (2000). *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

[Wrobel et al. 2000] Wrobel, S., K. Morik und T. Joachims (2000). *Maschinelles Lernen und Data Mining*. In: G?rz, G., C.-R. Rollinger und J. Schneeberger, Hrsg.: *Einf?hrung in die K?nstliche Intelligenz*, S. 3–. Oldenburg.

[YANG und HONAVAR 1998] YANG, JIHOON und V. HONAVAR (1998). *Feature Subset Selection Using a Genetic Algorithm*. IEEE Intelligent Systems, 13:44–49.

[ZHENG 1998] ZHENG, ZIJIAN (1998). *A Comparison of Constructing Different Types of New Feature for Decision Tree Learning*. In: LIU, HUAN und H. MOTODA, Hrsg.: *Feature Extraction, Construction and Selection: a Data Mining Perspective*, S. 239 – 255. Kluwer.

[ZÜCKER und KIETZ 2000] ZÜCKER, REGINA und J.-U. KIETZ (2000). *How to pre-process large databases*. In: *Data Mining, Decision Support, Meta-learning and ILP: Forum forPractical Problem Presentation and Prospective Solutions*, Lyon, France.

Erklärung

Hiermit erkläre ich, Hanna Köpcke, die vorliegende Diplomarbeit mit dem Titel

Häufigkeitsbasierte Merkmalsgenerierung für die Wissensentdeckung in Datenbanken

selbständig verfaßt und keine anderen als die hier angegebenen Hilfsmittel verwendet zu haben.

Dortmund, den 9. Dezember 2003