Reihe: Telekommunikation @ Mediendienste · Band 16

Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof. Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Rainer Kuhlen, Konstanz, Dr. Rudolf Pospischil, Brüssel, Prof. Dr. Claudia Löbbecke, Köln, und Prof. Dr. Christoph Zacharias, Köln

PD Dr.-Ing. habil. Martin Engelien
Prof. Dr.-Ing. habil. Klaus Meißner (Hrsg.)

Virtuelle Organisation und Neue Medien 2004

Workshop GeNeMe2004
Gemeinschaften in Neuen Medien

TU Dresden, 7. und 8. Oktober 2004

EUL VERLAG

Bibliographische Information der Deutschen Bibliothek

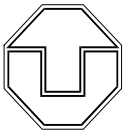
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

ISBN 3-89936-272-1
1. Auflage September 2004

Printed in Germany
Druck: RSP Köln

JOSEF EUL VERLAG GmbH
Brandsberg 6
53797 Lohmar
Tel.: 0 22 05 / 90 10 6-6
Fax: 0 22 05 / 90 10 6-88
E-Mail: info@eul-verlag.de
http://www.eul-verlag.de

Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.



Technische Universität Dresden - Fakultät Informatik
Privat-Dozentur Angewandte Informatik, Professur Multimediatechnik

PD Dr.-Ing. habil. Martin Engelien
Prof. Dr.-Ing. habil. Klaus Meißner
(Hrsg.)

http://www.pdai.de/geneme/

GeNeMe 2004

Gemeinschaften in Neuen Medien

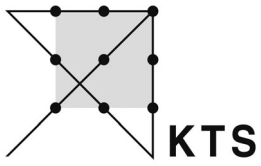
Workshop zu Organisation, Kooperation und Kommunikation auf der Basis innovativer Technologien

Forum für den Dialog zwischen Wissenschaft und Praxis

an der
Fakultät Informatik der Technischen Universität Dresden

in Zusammenarbeit mit der
Gesellschaft für Informatik e.V.
GI-Regionalgruppe Dresden

gefördert von der Klaus Tschira Stiftung



KLAUS TSCHIRA STIFTUNG
GEMEINNÜTZIGE GMBH

am 07. und 08. Oktober 2004 in Dresden
www.geneme.pdai.de
geneme@pdai.de

A.8 Data Mining in Peer-to-Peer-Systemen

Magnus Kolweyh, Ulrike Lechner
Universität Bremen

1. Einleitung/Motivation

Neben „klassischen“ virtuellen Gemeinschaften, die einen gemeinsamen Netzort für die Interaktion benutzen und die auf technischer Ebene vornehmlich durch den Einsatz von Client-Server-Architekturen gekennzeichnet sind, lassen sich Gemeinschaftsformen identifizieren, deren Ursprung in vollkommen dezentralen Peer-to-Peer (P2P)-Netzen liegt. Diese P2P-Gemeinschaften haben in einigen Anwendungsgebieten wie dem Filesharing großen Erfolg [SF02].

Die Erfahrungen mit P2P-Netzwerken zeigen, dass P2P-Gemeinschaften, d.h. Gemeinschaften, die P2P-Netzwerke für die Interaktion nutzen, sich stark von anderen Gemeinschaften unterscheiden. Die neuartigen P2P-Netzwerke beeinflussen die Gestaltung der Dienste, die sozialen Netzwerke, genauso wie die Inhalte, die in den Gemeinschaften ausgetauscht werden. Formal ist eine Gemeinschaft definiert als ein Medium und eine Menge von Agenten. Ein Medium ist beschrieben durch ein System von Kommunikationskanälen, einen logischen Raum und eine Organisation [LS00].

Der gemeinsame Netzort virtueller Gemeinschaften wird vielfach als entscheidend für die Interaktion und die Entwicklung eines Gemeinschaftsgefühls und eines sozialen Netzwerkes angesehen (vgl. [Hu02]). Dieser eine, für die Gemeinschaft zentrale Netzort ist Anlaufpunkt für die Gemeinschaft und bietet vor allem Dienste für die Organisation und Strukturierung der Interaktion und der in einer Gemeinschaft verfügbaren Inhalte [Pre00,SS00]. Allerdings können sich die Gestaltung des Netzortes genauso wie die Ausprägung des sozialen Netzwerkes je nach Geschäftszweck unterscheiden [Hu02].

Die P2P-Gemeinschaften, die statt eines gemeinsamen Ortes ein P2P-Netzwerk verwenden, unterscheiden sich von anderen virtuellen Gemeinschaften maßgeblich: Sie sind offen, Inhalte, Ressourcen und Strukturen sind typischerweise nicht persistent und werden dezentral bereitgestellt. Es gibt keine definierte Sprache für die Repräsentation von Inhalten und kaum vordefinierte Dienste für die Organisation und Strukturierung. Es existiert des Weiteren kein zentraler Punkt mit Information, welche Inhalte im Netzwerk verfügbar sind. Dies alles trägt dazu bei, dass P2P heute im wesentlichen auf Filesharing oder sehr einfache Applikationen beschränkt ist. Allerdings können P2P-Netzwerke in Bezug auf Skalierbarkeit und Performanz Vorteile gegenüber den typischen Client-Server-Architekturen haben.

Es fehlen bisher Algorithmen, um die großen, meist unstrukturierten Datenmengen von P2P-Applikationen bewältigen zu können. Komplexe Data Mining-Systeme lassen sich

in diesem Kontext nur bedingt einsetzen. Sie sind eng auf ein einzelnes Data Warehouse zugeschnitten und arbeiten nur mit diesem gut zusammen. Eine Adaption auf allgemeine Data Warehouses oder analoge Datenquellen, wie dezentrale Systeme, gestaltet sich als primäre Aufgabe. P2P stellt auch eine Reihe von neuen Anforderungen in Bereichen wie Skalierbarkeit, Nicht-Persistenz von Daten und Strukturen, Sicherheit und Anonymität an die Data Mining-Technologien auf.
In diesem Papier wird ein Konzept und eine erste simulative Implementierung eines Data Mining-Algorithmus als Teil eines Knowledge Discovery-Prozesses im P2P-Kontext präsentiert. Wir zeigen die Adaption des Knowledge Discovery in Databases Prozesses und von Association Rule Mining. Exemplarisch werden an Hand eines empirischen Modells, das Assoziationsregeln innerhalb eines typischen Filesharing-Systems extrahiert, Modell und Adaption validiert.

2. Knowledge Discovery in P2P-Systemen

In der Literatur [FPSS96b] wird Data Mining oft in zweierlei, unterschiedlicher Hinsicht verstanden. Zum Einen als Synonym für Knowledge Discovery, zum Anderen als essentieller Teil eines gesamten Knowledge Discovery-Prozesses. Knowledge Discovery in Databases (KDD) hat als Zielsetzung, Methoden und Werkzeuge zur automatischen Datenanalyse zur Verfügung zu stellen, mit denen Information und Wissen aus großen Datenbeständen extrahiert werden können. Diese Zielsetzung kann für bestimmte Datenmengen konkretisiert und als Mustersuche beschrieben werden. Neben diesem reinen Pattern Searching müssen die Daten zunächst so aufbereitet werden, dass eine formalisierte Suche überhaupt in einem geeignetem Rahmen stattfinden kann. Gleichzeitig liefern selbst statistisch validierbare Muster in den Datenmengen noch keine Information und vor allem kein Wissen für den untersuchten Kontext. Um die Daten ausreichend beschreiben zu können, sind daher eine Reihe von weiteren Schritten notwendig, die sowohl vor, als auch nach der eigentlichen Datensuche, dem Data Mining, auszuführen sind.
Wir verstehen Data Mining also als Teil des gesamten KDD-Prozesses, genauer als essentiellen und vor allem nicht isolierbaren Teil. Unser Ansatz benutzt die allgemeine Beschreibung eins KDD-Prozesses von Fayyad, Piatetsky-Shapiro et al. [FPS96]. Wir verknüpfen dieses ganzheitliche Knowledge Discovery-Konzept mit den speziellen Beschaffenheiten von P2P-Systemen [MKL+02] und identifizieren für einen KDD-Prozess im P2P-Kontext folgende Schritte:

1. Problembeschreibung:
   (a) Benennung der wesentlichen Ziele des KDD-Prozesses

(b) Beschreibung der Art des zu extrahierenden Wissens und dessen Repräsentation

2. Datenvorverarbeitung
   (a) Datengewinnung: Methodik und Form der Datengewinnung, Integration und Migration verschiedener Datenquellen mit Hilfe von Kodierungs- und Integrationsverfahren
   (b) Datenbereinigung: Prüfen der gewonnenen Daten, Umgang mit verrauschten, fehlenden oder widersprüchlichen Daten, Auflösung von Konflikten, Bereinigung von inkonsistenten Daten
   (c) Sichtung relevanter Daten, Auswahl von Lern- und Testsets
   (d) Transformations-, Aggregations-, und Konsolidierungsprozesse
3. Dezentrales Data Mining
   (a) Synchronisation: Dezentrales Protokoll, Voting-Strategien
   (b) Dezentrale Mustersuche: Identifikation von signifikanten Mustern, Klassifikation mit Hilfe von Regeln, Bäumen, Regression, Clustering
4. Post Data Mining
   (a) Evaluation der generierten Muster: Tests auf Unabhängigkeit, Konsistenz und Widersprüche
   (b) Entwurf eines Knowledge Model: Vorhersagen treffen, notwendige Veränderungen aufzeigen
   (c) Wissensrepräsentation: Visualisierung der Daten, Regeln und Abhängigkeiten

Für P2P-Systeme lassen sich die Datenvorverarbeitung und der dezentrale Prozess der Datengewinnung als essentielle Anforderungen identifizieren. Die Daten sind chaotisch im Netz verteilt und lassen sich in kein semantisches Schema integrieren. Gleichzeitig können Algorithmen nie zentral angewandt werden, um den einzelnen Peers Wissen über ihre Umwelt zukommen zu lassen. Während sich vorherige Arbeiten bereits mit den dezentralen Konzepten wie Synchronisation und Datenerhalt auf Netzebene auseinandersetzen [WS03], verfolgen wir einen empirischen Ansatz für ein real existierendes System. Wir benutzen hierfür ein bekanntes Data Mining Konzept, Association Rule Mining, und stellen im Folgenden die speziellen Anforderungen eines solchen Konzeptes im P2P-Kontext dar.

3. P2P Data Mining

Wir haben im vorherigen Abschnitt den Knowledge Discovery-Prozess dargestellt und hervorgehoben, dass Data Mining einen signifikanten Teil dieses Prozesses beinhaltet. Gleichzeitig ist die Datenvorverarbeitung immer entscheidend für den Erfolg eines solchen Ansatz und in chaotischen Datensystemen wie P2P Systemen nicht trivial. Wir

stellen hier zunächst Association Rule Mining vor und beschreiben danach die speziellen Anforderungen an P2P Systeme.

3.1 Association Rule Mining

Association Rule Mining (ARM) stellt ein verbreitetes Data Mining-Konzept zur Verfügung. Assoziationen können als spezielle Typen von Wissen aufgefasst werden. Andere Arten von Wissen sind in diesem Zusammenhang etwa Klassenbeschreibungen, quantitative Vorhersagen oder Clustering-Methoden. ARM stellt Informationen durch Assoziationsregeln dar, die Aussagen in Form von „ein Kunde der Milch kauft, kauft auch Brot“ implizieren können.

ARM lässt sich wie folgt definieren : $I=(i_1,...,i_n)$ ist eine Menge von Literalen, auch *Items* genannt. In einem P2P-Netz sind dies Inhalte wie Dokumente, Musik oder Videodateien sein. $A_i=v$ ist ein Item, wobei v ein Wert des Attributes A_i einer Relation $R(A_1,...,A_n)$ ist. X ist eine Teilmenge von Items in I, *Itemset* genannt. D ist eine Menge von Transaktionen, wobei jede Transaktion aus den Werten *tid* (einem eindeutigen Identifier für die Transaktion und einem t-Itemset besteht: $t=(tid, itemset\_t)$). So ist ein Tuple $(v_1,...,v_n)$ Teil einer Relation $R(A_1,...,A_n)$. In einem P2P-Netz kann das Auftreten von Musiksongs bei bestimmten Benutzern als Transaktion formuliert werden. Eine Transaktion t beinhaltet einen Itemsatz für alle Items mit $i \in X$ und i als einem t-Itemsatz. Im P2P-Kontext muss also für alle benutzten Instanzen der betrachteten Musiksongs festgehalten werden, ob diese von einem bestimmten Benutzer produziert werden. Jeder Itemsatz X kann nun mit dem relativen Häufigkeitsmaß (support) $supp(X)= |X(t)| / |D|$, $X(t)=\{t \text{ in } D \mid t \text{ enthält } X\}$ bezüglich seines Auftretens in I gekennzeichnet werden,

Eine Assoziationsregel ist dann die Implikation $X \Rightarrow Y$, mit X und Y als Itemsätze. Für alle Assoziationsregeln lässt sich nun neben dem Häufigkeitsmaß (support) noch ein weiteres Maß, die *confidence* definieren:

1. Der support $supp(X \Rightarrow Y)$ einer Assoziationsregel $X \Rightarrow Y$ ist $X \cup Y$
2. Die confidence $conf(X \Rightarrow Y)$ einer Regel ist das Verhältnis $|(X \cup Y)(t)| / |X(t)|$ mit Transaktionen t oder eben $supp(X \cup Y) / supp(X)$.

Man kann also mit $supp(X \Rightarrow Y)$ die relative Häufigkeit eines auftretenden Musters und mit $conf(X \Rightarrow Y)$ die Stärke dieser Implikation bezeichnen. Agrawal u.a. benutzen ein Support-Confidence-Framework [AIS93] um mit Hilfe dieser beiden Faktoren Assoziationsregeln in einer Datenbank aufzustellen.

Üblicherweise werden von Experten oder Nutzern der untersuchten Domäne die Bezugsparameter *minsupport* und *minconfidence* bereitgestellt, um die Stärke der extrahierten Regeln als exakt, stark, wahrscheinlich oder nicht interessant zu bewerten.

Wir können hiermit eine minimale relative Häufigkeit eines Musters (bei P2P-Systemen sind dies Dateien) bzw. eine minimale Stärke der Implikation einer Regel festlegen und so nur Muster berücksichtigen, die häufig genug bezüglich der Gesamtmenge sind bzw. nur Regeln mit starker Aussagekraft als relevant in die Ergebnismenge integrieren.
Für die Suche nach interessanten Regeln hat Piatetsky-Shapiro [PS91] bereits argumentiert, dass eine Regel $X \Rightarrow Y$ nicht interessant ist, falls gilt: $supp(X \Rightarrow Y) \approx supp(X) * supp(Y)$. Dies lässt sich auch aus der Wahrscheinlichkeitstheorie interpretieren, denn mit $p(X \cup Y) \approx p(x) * p(Y)$ wird impliziert, dass X unabhängig von Y existiert [MS98].
Wir wollen im Folgenden Assoziation Rule Mining für ein P2P-Filesharing-System adaptieren und zeigen hier die speziellen Anforderungen auf. Diese Beschreibung liefert die Basis für unseren empirischen Ansatz.

3.2 Rule Mining in Peer-to-Peer-Systemen

Wir können zwei besondere Herausforderungen bezüglich der untersuchten Datenbank für Data Mining und hier im speziellem für Association Rule Mining identifizieren, (1) Größe und (2) Verteiltheit.
Datenbanken mit Terrabytes von Daten sind oft zu groß, um sie in einem einzelnen Schritt nach Mustern zu durchsuchen. Solche Problemstellungen werden in der Literatur als Large-Scale-Probleme gekennzeichnet. Wir sprechen im Data Mining-Kontext daher auch von Large-Scale Associaton Rule Mining. Eine ideale Lösung für dieses Problem würden echte parallele Instanzen auf Hardwareebene darstellen, welche teuer oder schlicht nicht verfügbar sind. In der Praxis werden daher in solchem Fall oft probabilistische Methoden wie Instanzauswahl mittels Heuristiken angewandt [Toi96]. Verteilte Datenbanken, die nicht sequentiell abrufbare Daten enthalten, bilden für Association Rule Mining die zweite Herausforderung. Wir sehen heute einen Übergang von Datenbanken, die alle Instanzen auf einem einzelnen Server halten, hin zu verteilten, miteinander mitverknüpften Datenbanken in verschiedenen Systemen. P2P stellt dies exemplarisch dar [MKL+02].
P2P-Systeme sind ein sehr interessantes Beispiel, da sie sowohl sehr große als auch verteilte Systeme darstellen, die auch in der Praxis relevant sind. Für solche Systeme lässt sich das Large-Scale Distributed Data Mining Problem (LSD-ARM) identifizieren [WS03].
Hierfür muss zunächst eine geeignete Synchronisation entwickelt werden. Die Knoten müssen im Netz unabhängig voneinander agieren und die Daten immer aktuell halten. Gewonnene Regeln sind immer nur zu bestimmten Zeitpunkten aktuell, sobald neue Daten eintreffen, muss eine neue Iteration des benutzen Algorithmus gestartet werden.

Gleichzeitig gibt es keinen bestimmten Zeitpunkt, an dem der Mining-Prozess als abgeschlossen gekennzeichnet werden kann. Wenn zu bestimmten Zeitpunkten also jeweils neu produzierte Daten neue Analyseprozesse starten, muss zusätzlich berücksichtigt werden, wie aufwändig ein solches Modell ist. Ein Knoten kann keine beliebige Menge von anderen Knoten als Datenmenge speichern, sondern ist auf die lokalen Informationen seiner direkten Nachbarknoten angewiesen. Ein weiteres Problem bei der Untersuchung von Filesharing-Daten bildet die Ad-Hoc-Natur des benutzen P2P-Netzes. Knoten betreten und verlassen das Netz fast willkürlich mit hoher Frequenz [RFI02].
Die hier beschriebenen Anforderungen können in einem kompletten KDD-Prozess sichtbar gemacht werden. Wir entwickeln hierfür ein Modell, das diesen Prozess exemplarisch wiedergeben kann.

4. Systembeschreibung

Um die im vorherigen Abschnitt beschriebenen Anforderungen in der Praxis umzusetzen, entwickeln wir ein System *PeerMiner*, das zunächst die Daten aus einem P2P-Netz ausliest und filtert. Auf der gewonnenen Datenmenge benutzen wir einen dezentralen Association Rule Mining-Algorithmus (*Majority-Rule*) und generieren in einem simulierten Filesharing-Prozess dynamisch Assoziationsregeln an einzelnen Peers.

4.1 Datenvorverarbeitung

Im weiteren Verlauf wollen wir die von einer bestehenden Filesharing-Community produzierten Daten analysieren und Assoziationsregeln generieren. Wir benutzen als Basis ein typisches, bekanntes Filesharing-Network, Direct Connect [Neo]. Direct Connect bietet als Struktur sogenannte zentrale Hubs, auf denen dann aber die Clients vollkommen dezentral miteinander kommunizieren. Die Hubs dienen zur Authentifizierung, Sicherung gegenüber Fakern, Clustering nach gewünschten Inhalten oder auch bereitgestellter Bandbreite und vor allem zur Filterung bezüglich der Menge der produzierten Inhalte. Für die meisten Hubs existiert ein Sharelimit, welches als Zutrittstoken dient. Das Hub leitet einen Teilnehmer bei nicht ausreichendem Sharesize an niedriger priorisierte Hubs weiter, welche geringere Anforderungen an die Shares stellen. Das Freerider-Problem [AH00] wird damit sinnvoll bekämpft, einem Freerider wird der Zugang zum System gestattet, er findet sich dann meist auf Hubs wieder, die selbst nur wenig Inhalte zur Verfügung stellen. Analog dazu bleiben Poweruser auf den Hubs zusammen. Wir können in unserem Ansatz einen Poweruser generieren und haben auf große Datenmengen Zugang ohne Freerider und Faker berücksichtigen zu müssen.

Die bereitgestellten Daten können beliebigen Formats und Inhalts sein, wir sind hiermit nicht an bestimmte Dateitypen gebunden.
Gleichzeitig bietet das Direct Connect-Protokoll die Möglichkeit, komplette Sharelisten von einzelnen Nutzern relativ einfach zu transferieren. Zu diesem Zwecke implementieren wir *PeerMiner*, der automatisch Datenlisten von verbundenen Nutzern eines Hubs aus dem Netz liest und in ein lokales Datenframework schreibt. Die Daten werden gefiltert, um eindeutige Items zu erhalten. Hierzu muss ein Modell entwickelt werden, welches möglichst effizient und schnell arbeitet. Zur Erzeugung dieses Modells dient eine Voruntersuchung, die alle ungefilterten Daten bezüglich ihrer Häufigkeit darstellt.

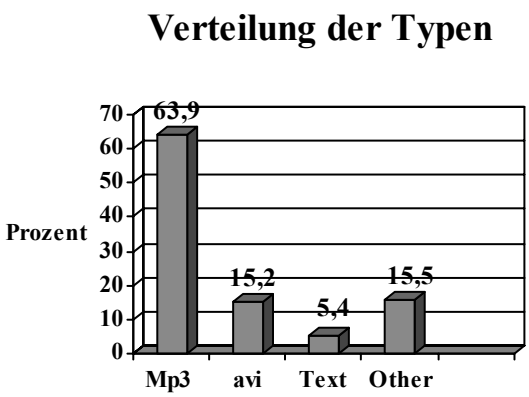


Abbildung 1

Abbildung 1 zeigt die Verteilung einzelner Datentypen im Netz bezüglich ihrer Häufigkeit, hierbei spielt es keine Rolle, wie viel Datenmenge insgesamt durch einzelne Datentypen entsteht, für den Data Mining-Prozess ist allein das Auftreten an sich interessant. Als häufig anzutreffende Typen lassen sich hier mp3-komprimierte Audiodaten sowie Videodaten im avi-Format erkennen.

Micheal Jackson - Beat it.mp3	9.94 MB
05.Micheal Jackson - Beat it(1) [SeriouS].mp3	5.92 MB
04 Beat it.mp3	3.95 MB
michael jackson - beat it (moby remix).mp3	3.39 MB
08 Michael Jackson - Beat It.mp3	1.70 MB
10 - Beat It.mp3	4.98 MB

spartan.avi	699.26 MB
spartan.avi	699.26 MB
spartan.avi	699.26 MB
spartan.avi	699.26 MB
Spartan.DVDSCR .XViD-DVL.avi	699.26 MB
spartan.avi	699.26 MB
spartan.avi	699.26 MB

Tabelle 1

Musikdateien haben jedoch für unseren Ansatz einen entscheidenden Nachteil: Ein Musiksong kann durch eine Vielzahl von Ausdrücken beschrieben werden, beispielsweise existiert für Michael Jacksons Song „Beat it“ eine Vielzahl von Beschreibungen (siehe Tabelle 1). Zwar bietet das Protokoll auch die Möglichkeit neben den Bezeichnern der Daten auch deren Größe zu transferieren, jedoch sind die meisten vorhandenen mp3-Dateien in unterschiedlichen Bitraten kodiert, sodass sich auch hier keine eindeutigen Items als Eingabe für unseren Data Mining-Ansatz identifzieren lassen. Um das Modell einfach, nachvollziehbar und effizient zu halten, berücksichtigen wir nur avi-Daten. Wir untersuchen eine Stichprobe von vorhandenen avi-Dateien und stellen fest, dass es sich größtenteils um komprimierte (divx, xvid) Filme handelt, wobei identische Filme sehr häufig eine identische Dateigröße um die 700 Megabyte (eine CD) erreichen. Die Bezeichner selbst sind größtenteils identisch und beinhalten lediglich den Filmnamen selbst (Tabelle 1).

Wir können damit eine für den Association Rule Mining-Prozess günstige Eingabe erhalten, indem wir zunächst nur avi-Dateien filtern, deren Größe nahe bei 700 MB liegt und implizieren damit kodierte Filme als untersuchten Datenbestand. *PeerMiner* transferiert alle verfügbaren Dateiinformationen in die lokale Datenbank und filtert nach Dateiendung und Größe. Diese einzelnen Schritte spiegeln die im KDD-Prozess beschriebenen Anforderungen zur Datengewinnung, Datenbereinigung und Auswahl der untersuchten Instanzen wieder.

Im Anschluss an diesen Prozess ist eine geeignete lokale Simulation des dezentralen Data Mining-Prozesses gefordert, hierfür benutzen wir ein lokales Majority-Voting-Protokoll lokal an den einzelnen Peers.

.

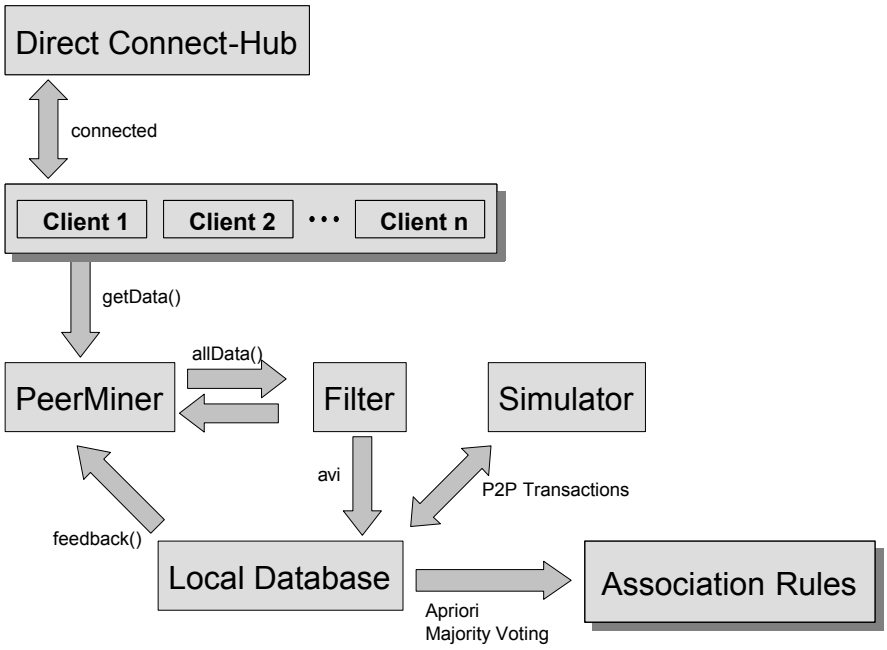


Abbildung 2

4.2 Majority-Vote Mining

Wolff/Schuster [WS03] beschreiben einen Algorithmus, der das LSD-ARM-Problem effizient löst, indem es auf das gut untersuchte Problem von Majority Votes reduziert wird. Der Ansatz kombiniert sequentielles Association Rule Mining lokal an den einzelnen Clients mit einen Majority-Voting-Protokoll. An jedem Knoten werden Assoziationsregeln lokal generiert. Der Algorithmus terminiert nicht und liefert immer eine Ad-Hoc-Beschreibung zu einem bestimmten Zeitpunkt an jedem Knoten. Obwohl wir die Daten mit *PeerMiner* auf eine lokale, zentrale Datenbasis extrahiert haben, muss berücksichtigt werden, dass diese Vorgehensweise nur zur Simulation dient und real nur in kleinen Netzen mittels Synchronisation und Broadcasting eingesetzt werden kann, nicht aber in den hier untersuchten P2P-Systemen. Wir benutzen daher auch in der Simulation das LSD-Majority-Protokoll, um an den einzelnen Clients Assoziationsregeln zu generieren und diese mittels Voting zu aktualisieren.

4.3 Simulation

Um Assoziationsregeln aus einem P2P-Netz zu gewinnen, benutzen wir das in Abbildung 2 dargestellte Modell. *PeerMiner* liefert uns die Itemsets eines festzulegenden Datenuniversums an beteiligten Nutzern.

1) Love_actually=no 8 ==> Mystic_River=yes Kill_Bill=no 8 conf:(1)
2) Mystic_River=yes Love_actually=no 8 ==> Kill_Bill=no 8 conf:(1)
3) Love_actually=no Kill_Bill=no 8 ==> Mystic_River=yes 8 conf:(1)
4) Love_actually=no 8 ==> Kill_Bill=no 8 conf:(1)
5) Love_actually=no 8 ==> Mystic_River=yes 8 conf:(1)
6) Lord_of_the_Rings_3=no Master_and_Commander=no 7 ==> Kill_Bill=no 7 conf:(1)
7) Spartan=yes 7 ==> Kill_Bill=no 7 conf:(1)
8) Cold_Mountain=no 7 ==> Along_came_Polly=no 7 conf:(1)
9) Mystic_River=yes Lord_of_the_Rings_3=no Master_and_Commander=no 6 ==> Kill_Bill=no 6 conf:(1)
10) Mystic_River=yes Lord_of_the_Rings_3=no Kill_Bill=no 6 ==> Master_and_Commander=no 6 conf:(1)

1) Mystic_River = yes and Kill_Bill = no ==> Love_actually = no
2) Love_actually=no Kill_Bill=no ==> Mystic_River=yes
3) Mystic_River = yes ==> Along_came_Polly = yes Love_actually = no
4) Love_actually=no ==> Kill_Bill=no
5) Mystic_River = yes ==> Along_came_Polly = yes Love_actually = no
6) Cold_Mountain = yes and Mystic_River = yes ==> Along_came_Polly = yes Last_Samurei = no
7) Mystic_River = yes and Kill_Bill = no ==> Love_actually = no Last_Samurei = no
8) Cold_Mountain = yes and Kill_Bill = no ==> Spartan = yes Love_actually = no
9) Mystic_River = yes and Kill_Bill = no ==> Spartan = yes Love_actually = no
10) Mystic_River = yes ==> Love_actually = no Last_Samurei = no

Abbildung 3

PeerMiner kann des Weiteren nach bestimmten Größen und Typen einer Datei filtern und die Ergebnisse in eine lokale Datenbank schreiben. Wir simulieren dann ein dynamisches P2P-Netzwerk, mit probabilistischem Modell für Transaktionen zwischen den einzelnen Nutzern und dem in Abschnitt 3.4 vorgestellten Modell zur Synchronisation durch Majority Voting.

4.4 Ergebnisse

Als Ergebnis der Datenvorverarbeitung haben wir eine Eingabe für einen Association Rule Mining-Algorithmus erhalten. Wir können nun den beschriebenen Majority—Voting-Algorithmus benutzen um simulativ dezentral Assoziationsregeln zu erzeugen. Vergleichsweise benutzen wir den bekannten Apriori-Algorithmus [AS94] dezentral. Abbildung 3 zeigt hier exemplarisch zwei extrahierte Regelmengen, mit den vorgestellten Bezugsparametern *minsupport=0,35* und *minconfidence=0,9.* In Regel 1) der Abbildung 3 wird durch das Item *Love_actually* mit Attribut *yes* (=Datei bei Benutzer vorhanden) mit der absoluten Häufigkeit 8 die Assoziationsregel *Mystic_River=yes* und *Kill_Bill=no* mit *conf=1* impliziert, also der stärkstmöglichen Implikation bei *supp(X) > minsupport*. Die linke Tabelle der Abbildung 3 zeigt hierbei eine Regelmenge, die zentral mit Apriori durchsucht wurde, die rechte Tabelle stellt eine lokal extrahierte Regelmenge mittels Majority Voting auf der selben Eingabemenge dar. Wir sehen, dass die Algorithmen zu unterschiedlichen Regelmengen führen, aber auch gemeinsame Regeln beinhalten.

5. Zusammenfassung

Wir haben die Schnittstelle zwischen Data Mining und P2P-Systemen untersucht und Associaton Rule Mining als wertvolles Konzept identifiziert. Hierfür haben wir zunächst einen kompletten Knowledge Discovery-Prozess mit den zugrundeliegenden

besonderen Anforderungen für P2P beschrieben. Exemplarisch wurde von uns Association Rule Mining auf ein bestehendes, typisches Filesharing-Netz, DirectConnect, angewandt. Wir haben hierzu *PeerMiner* implementiert, welcher die im Netz befindlichen Informationen auslesen, nach Filetypen filtern und in eine lokale Datenbank speichern kann. Die extrahierten Daten wurden in einem dezentralen Ansatz nach Assoziationsregeln durchsucht. Wir haben für P2P die speziellen Notwendigkeiten wie Synchronisation skizziert, einen verteilten Algorithmus Majority Voting implementiert, dessen Einsatz in einer Simulation mit realen Filesharing-Daten gezeigt und einige extrahierte Regeln dargestellt.

P2P-Systeme stellen für die Forschung eine Reihe interessanter Thematiken zur Verfügung. Bei der Betrachtung von Beiträgen im P2P-Kontext lässt sich insbesondere ein Schwerpunkt im Bereich Suchen und Lokalisation von Ressourcen identifizieren [MKL+02,RFI02]. Für den Erfolg zukünftiger P2P-Systeme auch außerhalb der Filesharing-Domäne, wird neben dem effizienten Suchen auf Netzwerkschicht vor allem eine integrierte Semantik entscheidend sein. Aktuelle Applikationen wie Oceanstore oder Free Haven [DFM01] deuten in den Bereichen Datenmanagement und Anonymität die erweiterten Möglichkeiten von P2P außerhalb der Filesharing-Domäne bereits an. Data Mining kann als Konzept zur Datenanalyse dienen und somit eine Grundlage für neuartige Formen von P2P-Systemen liefern. Methoden wie Associaton Rule Mining sind dabei essentielle Bestandteile bei der Implementierung innovativer Services auf Anwendungsebene.

6. Literatur

[AH00] E. Adar and B. Huberman. Freeriding on Gnutella. Firstmonday 5 (10), 2000.

[AIS93] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207–216, Washington, D.C., 26–28 1993.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994. http://citeseer.ist.psu.edu/agrawal94fast.html

[CNR02] M. Clement, G. Nerjes and M. Runte (2002) Bedeutung von Peer-to-Peer Technologien für die Distribution von Medienprodukten im Internet. In D. Schoder, K. Fischbach and R. Teichmann (eds): *Peer-to-Peer. Ökonomische, technische und juristische Perspektiven*, pp 71-80, Springer Verlag, 2002.

[DFM01] R. Dingledine, M. J. Freedman, and D. Molnar. The free haven project: Distributed anonymous storage service. Lecture Notes in Computer Science, 2009:67–95, 2001.

[FPSS96a] U. Fayyad, G.P. Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. AI Magazine, 17:37–54, 1996.

[Hu02] J. Hummel. *Virtuelle Gemeinschaften*, University of St. Gallen, 2002.

[LS00] U. Lechner and B.F. Schmid Communities and media – towards a reconstruction of communities on media. In E. Sprague, (ed) *Proc. of the 33th Int. Hawaii Conference on System Sciences* (HICSS 2000). IEEE Press, 2000.

[FPS96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining toward a unifying framework. In Proceeding of The Second Int. Conference on Knowledge Discovery and Data Mining, pages 82--88, 1996.

[MKL+02] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer to peer computing. Technical Report 57, HP Labs, 2002.

[MS98] N. Megiddo and R. Srikant. Discovering predictive association rules. In Knowledge Discovery and Data Mining, pages 274–278, 1998.

[Neo] NeoModus. Direct connect. http://www.neomodus.com.

[PS91] Gregory Piatetsky-Shapiro: Discovery, Analysis, and Presentation of Strong Rules. Knowledge Discovery in Databases 1991: 229-248

[Pr00] J. Preece, *Online Communities*, Addison Wesley, 2000.

[RFI02] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of largescale peertopeer systems and implications for system design. IEEE Internet Computing Journal, 6(1), 2002.

[SET] SETI@home. Seti@home: Search for extraterrestrial intelligence. http://www.setiathome.ssl.berkeley.edu.

[SF02] D. Schoder and K. Fischbach. Peer-to-Peer Anwendungsbereiche und Herausforderungen. In D. Schoder, K. Fischbach, and R. Teichmann (eds*): Peer-to-Peer. Ökonomische, technische und juristische Perspektiven*, pp 3-24, Springer Verlag, 2002.

[SS00] K. Stanoevska and B.F. Schmid. Requirements Analysis for Community Supporting Platforms Based on the Media Reference Model. In: EM-Electronic Markets – Communities & Platforms,10(4), 2000.

[Ti98] P. Timmers. Business Models for Electronic Markets. In: *EM - Electronic Markets. The International Journal of Electronic Markets and Business Media* 8(2), 1998.

[Toi96] H. Toivonen. Sampling large databases for association rules. In: T. M. Vijayaraman, A.P. Buchmann, C. Mohan, and N.L. Sarda, editors, In Proc. 1996 Int. Conf. Very Large Data Bases, pages 134–145. Morgan Kaufman, 1996.

[UUT02] H. Unger, H. Unger and N. Titova. Structuring of decentralized computer communities. In *High Performance Computing 2002 (HPC 2002)*.pp 245-250, 2002.

[Wed] S. Wedeniwski. Zetagrid. http://www.zetagrid.net/.

[WS03] R. Wolff and A. Schuster. Association rule mining in peer to peer systems. In Proceedings of The Third IEEE International Conference on Data Mining (ICDM), page 363. IEEE computer society, 2000.