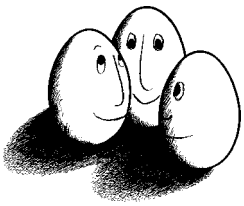
PG-Seminar

Zeitaspekte

Fabian Bauschulte



PG-Seminar
am Fachbereich Informatik
der Universität Dortmund

Dienstag, 13. November 2001 (8:16)

Betreuer:

Prof. Dr. Katharina Morik
Stefan Haustein

Inhaltsverzeichnis

1 Einleitung

Beim *Wissenmanagement*[1] steht man vor dem Problem, in große Datenmengen Wissen zu identifizieren und zu analysieren [4]. Ein interessantes Subproblem bildet in diesem Zusammenhang die *Analyse von Zeitreihen*, welche uns in den unterschiedlichsten Formen begegnen können.

Im Abschnitt 2 werde ich zwei verschiedene Arten von Zeitreihen am Beispiel präsentieren und kurz auf die Extraktion von Zeitreihen aus Dokumenten eingehen. Im Abschnitt 3 liefert eine genauere Betrachtung der Zeitreihen eine Strukturierung der vorkommenden Zeitphänomene. Dies führt uns zu der Fragestellung nach den sich aus den verschiedenen Zeitphänomenen ergebenden Lernaufgaben und der Repräsentation der Eingabedaten, welcher ich im Abschnitt 4 nachgehen werde[2]. Stellvertretend für die verschiedenen Lernverfahren werde ich im Anschluß an diese Einordnung in den Abschnitten 5 und 6 zwei unterschiedliche Lernverfahren im Detail vorstellen.

2 Zeitreihen

Im folgenden will ich zwei Beispiele für unterschiedliche Zeitreihen (engl. *time series*) präsentieren.

2.1 Messwerte von einem Prozess

Diese Form von Zeitreihen kann uns beispielsweise in folgenden Anwendungen begegnen:

- Intensivmedizin (z.B. Herzfrequenz, Atemfrequenz, Blutdruck)
- Aktienkurse
- Wetterdaten (z.B. Luftdruck, Lufttemperatur oder Windgeschwindigkeit)
- Roboter (z.B. Sensoren zur Abstandsmessung [15])

Ein wichtiges Merkmal dieser Art von Zeitreihen ist die *kontinuierliche Messung* in z.B. Tagen, Stunden, Minuten, Sekunden.

[1] Seminarthema: Knowledge Management
[2] nach [14]

Unterscheidung Univariat - Multivariat

Bei dieser Form von Zeitreihen werden muß man die Begriffe *Univariat* und *Multivariat* voneinander abgrenzen.

Univariat Hier wird nur ein Attribut pro Zeit gemessen. Dies könnte in der Intensivmedizin beispielsweise das Attribut *Herzfrequenz* sein.

Multivariat Hier werden k Attribute pro Zeit gemessen. Im Beispiel könnten dies die drei Attribute *Herzfrequenz, Atemfrequenz, Blutdruck* sein.

2.2 Datenbankrelationen

Diese Form von Zeitreihen kann uns beispielsweise in folgenden Anwendungen begegnen:

- Vertragsdaten
- Verkaufsdaten
- Benutzerdaten
- Lebenssituation (Einkommen, Alter)

Im Gegensatz zum vorherigen Beispiel gibt es hier keine kontinuierliche Messung, sondern die Daten liegen als zeitlich gestempelte Datenbanktupel vor.
Das Problem bei dieser Repräsentation ist, dass hier in den meisten Fällen ein Preprocessing notwendig ist. Die Zeitreihen müssen aus den Datenbanktupel 'extrahiert' werden, bevor die verschiedenen Lernverfahren (siehe 4) angewendet werden können. Im Rahmen der *Projektgruppe 402* wird uns diese Form der impliziten Speicherung von Zeitreihen häufiger begegnen als die kontinuierliche Messung.

2.2.1 Aggregation

Eine Form des Preprocessing ist die Aggregation, bei der man bestimmte Daten (deren genauen Detailwerte nicht benötigt werden) zusammenfasst, bevor eine Analyse stattfindet. Die Aggregation führt zu einer Dimensionsreduktion entlang der weggelassenen Dimensionen. Für einfache Aggregationen kann man die SQL-Aggregatfunktionen (*COUNT(), SUM(), MIN(), MAX(), AVG()*) verwenden. Soll über mehrere Attribute aggregiert werden, so kann der *CUBE-Operator*[3] nach [8] verwendet werden. Mit dem *CUBE-Operator* ist es so möglich, schnell verschiedene Zeitreihen unterschiedlichen Detailierungsgrads zu erzeugen.

[3]Seminarthema: Data Cubes

2.2.2 Einschränkung der Datenmenge

In der Praxis kann es aufgrund der Größe einer Datenbanken sinnvoll sein, nicht alle möglichen Attribute bzw. Relationen zu untersuchen, sondern nur eine Teilmenge mit interessanten bzw. relevanten Attribute bzw. Relationen für eine Analyse zu betrachten. Ein Ansatz ist es, interessante Subgruppen mit Hilfe des *MIDOS-Algorithmus*[4] nach [17] aufzudecken. MIDOS ist ein deskriptives Lernverfahren, welches daran interessiert ist, durch Hypothesen beschriebene Teilbereiche des Instanzenraums zu identifizieren, über die lokal interessante Aussagen gemacht werden können. Hierbei wird die Subgruppenerkennung multirelational, d.h. über mehrere Relationen hinweg, durchgeführt. Wurden vom MIDOS-Algorithmus interessante Subgruppen gefunden, so können die Zeitreihen nur aus diesen interessanten Subgruppen extrahiert werden.
Ein anderer Ansatz der Einschränkung der Datenmenge ist die Regelsuche mit Hilfe von *RDT/DB*[5]. RDT/DB nach [5] ist ein ILP[6]-Wissensentdeckungswerkzeug, das direkt mit einem Datenbank-Managementsystem interagiert. Mit Hilfe eines Regelschemas und gegebenen Hintergrundwissen ist es möglich, bestimmte Regeln direkt aus der Datenbank abzuleiten.
Nach einer Regelsuche könnte man dann bestimmte Datenausschnitte auswählen, die für eine Analyse besonders interessant sind.

2.3 Extraktion von Zeitreihen aus Dokumenten

Zeitreihen können natürlich auch in textuellen Dokumenten verborgen sein. Ein Beispiel ist die Domain *Aktien*. Es kann sinnvoll sein, aus unstrukturierten Texten Zeitreihen mittels *Informationsextraktion*[7] nach [6] zu extrahieren (z.B. Aktienkurse oder unternehmensrelvante Daten). Dies könnte in der Praxis so aussehen, dass aus den sich verändernden Dokumenten (z.B. Internetseiten) immer wieder bestimmte Werte extrahiert werden. Bei der *Extraktion* der Daten könnte auch eine domainspezifische *Ontologie* verwendet werden [13] um bessere Ergebnisse zu erzielen[8]. Aus den so erhalten Zeitreihen können mit den in Abschnitt 4, 5 und 6 beschriebenen Verfahren Regeln bzw. interessante Muster gefunden werden.
Gerade bei Internetseiten können relevante Seiten mit Hilfe des *Semantic Web*[9] [3] leichter gefunden werden. Hier werden Ontologien benutzt, um das Wissen über die Internetseiten zu strukturieren und so Agenten zugänglicher zu machen. Eine Ontologie ist eine Menge von Klassen, Relationen und Funktionen, die eine abstrahierte, vereinfachte

[4] Seminarthema: MIDOS/ KEPLER
[5] Seminarthema: RDT/DB
[6] ILP = inductive logik programming
[7] Seminarthema: Informationsextraktion
[8] Seminarthema: Ontologiebasierte Wissensextraktion
[9] Seminarthema: Semantic Web

Sicht auf den relevanten Teil der Welt darstellt [9]. Diese Sicht sollte konsistent sein, muss aber nicht komplett sein. Als mögliche Ontologien sind RDF(S)[10] [16] und die Erweiterung DAML+OIL [10] denkbar.

3 Zeitphänomene

Nach [14] kann man Zeitphänomene grundsätzlich nach zwei Aspekten strukturieren.
Auf einer niedrigen Abstraktionsebene betrachtet man nur die einzelnen Elemente ent-

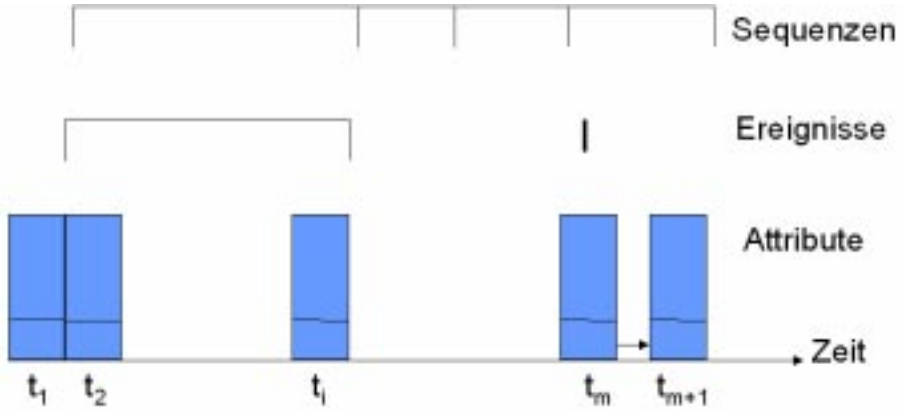


Abbildung 1: Zeitphänomene

lang der Zeitachse. Dies wird auch als **linear precedence** bezeichnet. Die meisten statistischen Ansätze beschränken sich auf diesen Aspekt der Zeit.
Auf einer höheren Abstraktionsebene kann man eine Menge von Elementen als ein einzelnes Ereignis auffassen. Ein Ereignis ist ein Tripel $(\mathsf{Zustand}, \mathsf{Start}, \mathsf{Ende})$. Der Zustand kann ein Wert oder ein Label (Trend bzw. eine Eigenschaft) sein. Beispiele für Ereignisse sind $(\mathsf{Steigend}, 3, 5)$, $(\mathsf{Fallend}, 7, 9)$ oder $(\mathsf{Stabil}, 10, 14)$. Wird eine Zeitreihe vollständig in Ereignisse zerlegt, so erhält man eine Sequenz von Ereignissen. Bei dieser Sicht spricht man von **immediate dominance**.
Abbildung 1 stellt diese zwei unterschiedlichen Sichten auf eine Zeitreihe dar.

4 Lernaufgaben und Repräsentation der Eingabedaten

Dieser Abschnitt basiert auf [14].

[10] Resource Description Framework (Schema)

4.1 Univariat

Lernaufgaben

Hier sind folgende Lernaufgaben denkbar:

Vorhersage Sei eine Zeitreihe mit Elementen bis zu einem Zeitpunkt t_i gegeben, sage das Element voraus, das an der Stelle t_{i+n} auftreten wird.

allgemeinene Trends erkennen Das Erkennen eines allgemeinen Trends, z.B. alle Elemente steigen.

lokale Trends erkennen Das Erkennen eines lokalen Trends, z.B. ein Zyklus[11] oder lokal steigende Werte.

Level change Das Finden von einem vom Standard abweichenden Wert (*Ausreißer*).

Clustering Fasse ähnliche Bereiche von aufeinanderfolgenden Werte zu Clustern zusammen.

Repräsentation der Eingabedaten

Die Messwerte können als Vektor der Form

$$i_l : t_1 a_1, ..., t_i a_i$$

repräsentiert werden. Hier sind nur numerische Werte zugelassen. Natürlich können die Zeitangaben hier auch entfallen.

4.2 Multivariat

Lernaufgaben

Hier sind natürlich auch alle Lernaufgaben aus Abschnitt 4.1 möglich, da man jede multivariate Zeitreihe als eine Menge von univariaten Zeitreihen auffassen kann. Als zusätzliche Lernaufgabe ist z.B. *das Finden von zusammen auftretenden Werten* denkbar.

Repräsentation der Eingabedaten

Die Messwerte können als Vektor der Form

$$i_l : t_1 a_{1_1} ... a_{1_k}, ..., t_i a_{i_1} ... a_{i_k}$$

repräsentiert werden. Auch hier sind nur numerische Werte zugelassen.

[11] z.B. (steigen → stabil → fallen) → (steigen → ...

4.3 Lernaufgaben bei einer gegebenen Sequenz von Ereignissen

Lernaufgaben

Häufige Sequenzen Finde häufige Episoden in Sequenzen. Eine Episode ist eine Menge von Ereignissen mit einer partiellen Ordnung [12].
Eine entdeckte Regel könnte z.B. *Wenn A auftritt, dann tritt B in der Zeit T auf* [7] sein.

Beziehungen zwischen Zeit-Intervallen Beziehungen zwischen Zeit-Intervallen lernen [11].
Eine entdeckte Regel könnte z.B. *A startet vor B* oder *B und C sind gleich* sein.

Repräsentation der Eingabedaten

- Zur Repräsentation der Eingabedaten zum Lernen von **häufigen Sequenzen** bietet sich der Sequenz-Vektor an:

$$I : T_1 A_1, ..., T_i A_i$$

  Hier bezeichnet A_i eine Menge von (nicht notwendigerweise numerischen) Attributen.

- Zur Repräsentation der Eingabedaten zum Lernen von **Beziehungen zwischen Zeit-Intervallen** bietet sich die Darstellung als Fakten der Form

$$P(I_1, T_b, T_e, A_r, ..A_s)$$

  an.

5 Diskretisierung von Zeitreihen durch Clustering

In diesem Abschnitt wird das Verfahren zur Diskretisierung von Zeitreihen nach [7] vorgestellt.

Prinzip

Hier wird eine Zeitreihe $s = (x_1, ..., x_n)$ zunächst in Subsequenzen $s_i = (x_i, ..., x_{i+w-1})$ aufgeteilt. Dies geschieht mit Hilfe eines Fensters der Länge w, welches man entlang der Zeitachse immer jeweils eine Einheit weiterbewegt. In den Subsequenzen bildet man dann Cluster $C_1, ...C_k$ von ähnlichen Subsequenzen. Jedes Cluster erhält dabei ein Symbol $a_1, ..a_k$ (die sog. *shapes*). Die Serie $s = (x_1, ..., x_n)$ kann dann mit Hilfe der *shapes* diskretisiert werden (siehe Abbildung 2).

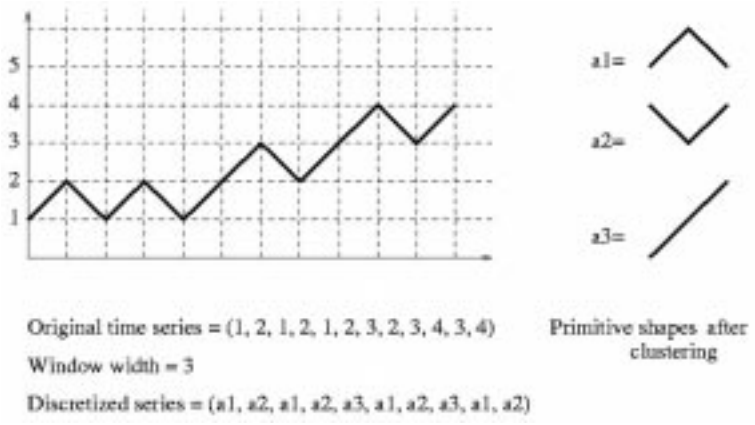


Abbildung 2: Beispiel einer Diskretisierung

Clustering

Man definiert ein Distanzmaß $d(s_i, s_j)$, welches die Entfernung zwischen zwei Subsequenzen s_i und s_j angibt. Ein Beispiel für ein Distanzmaß ist der euklidische Abstand:

$$d(\bar{x}, \bar{y}) = \sqrt[2]{\sum (x_i - y_i)^2}$$

In vielen Anwendungen müssen die Subsequenzen allerdings noch normalisiert werden, da sie evtl. das gleiche Muster besitzen, sich aber in Amplitude und Grundlinie unterscheiden. Eine mögliche Normalisierung $\eta(\bar{x})$ einer Sequenz $\bar{x}$ ist

$$\eta(\bar{x}) = x_i - E\bar{x}$$

wobei $E\bar{x}$ der Mittelwert der Sequenz ist.

Beim Clustering gibt eine Konstante $d > 0$ an, wie groß die Distanz zwischen den Subsequenzen s_i und s_j sein darf. Ist die Distanz $d(s_i, s_j) < d$, so gehören die Subsequenzen zum gleichen Cluster, andernfalls gehören sie verschiedenen Clustern. In Abhängigkeit von diesem Parameter ergeben sich dann die Cluster $C_1, ...C_k$.

Regeln

Aus einer diskretisierten Sequenz sind nun Regeln der Form

> Wenn A auftritt, dann tritt B in der Zeit T auf

einfach ableitbar. Eine Berechnung ist in der Zeit $m * k^2$ möglich, wobei k die Anzahl der Symbole und m die Anzahl der verschiedenen Möglichkeiten für T angibt.

Diese Regeln lassen sich nun zu der Form

Wenn A_1 und A_2 und ... und A_h innerhalb der Zeit V auftritt, dann tritt B in der Zeit T auf

erweitern. Die A_i können hierbei auch aus verschiedenen, univariaten Zeitreihen stammen. Das Problem ist allerdings, dass hier die Anzahl der Regeln stark ansteigt. Dies Problem kann allerdings gelöst werden, wenn man fordert, daß die A_i mit einer bestimmten minimalen Häufigkeit auftreten müssen. Nun kann mit Hilfe des Apriori-Algorithmus[12] [1] der Suchraum sehr effizient eingeschränkt werden.

6 Beziehungen zwischen Ereignissen

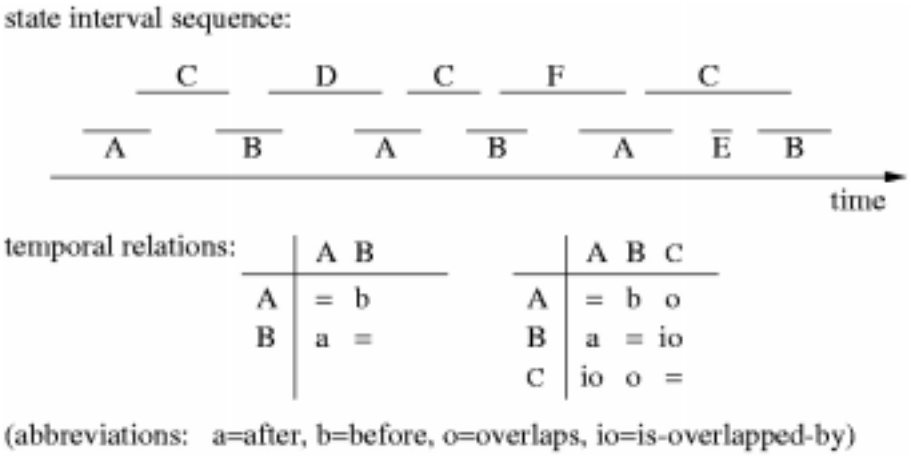


Abbildung 3: Darstellung der Beziehungen als Matrix

Von James F. Allen [2] wurden 13 verschiedene Intervallbeziehungen festgelegt. (z.B. A überlappt B, A beendet B, A vor B, A enthält B)

6.1 Beziehungen zwischen Zeit-Intervallen lernen [11]

Nach [11] kann man zeitliche Muster als Matrix darstellen (siehe Abbildung 3). Damit ein Muster als interessant erachtet wird, muss das Muster in einem Fenster der Länge t_{max} beobachtbar sein (siehe Abbildung 4). Somit ist der maximale Abstand zwischen den Ereignissen eines Muster begrenzt.

[12] Seminarthema: Entdeckung von Assoziationsregeln mit dem Apriori-Algorithmus

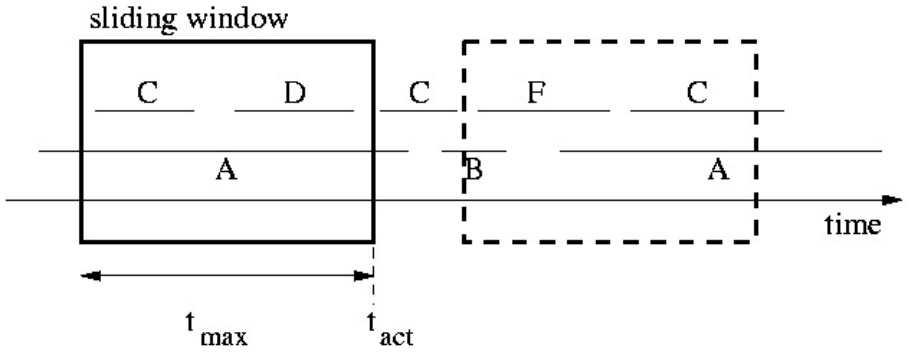


Abbildung 4: Sliding Windows

Häufige Muster finden

Zum Auffinden von häufigen Mustern wird der Apriori-Algorithmus [1] eingesetzt. Dieser Algorithmus hat die Eigenschaft, dass er bei der Kandidatengenerierung die Ordnung der Elemente ausnutzt. Der Support $\mathsf{supp}(\mathsf{P})$ eines Musters P wird dazu als die Zeit definiert, in der das Muster im Fenster beobachtet werden kann (siehe Abbildung 5). Weiterhin wird ein Muster P als häufig erachtet, wenn $\mathsf{supp}(\mathsf{P}) > \mathsf{supp}_{min}$ gilt.

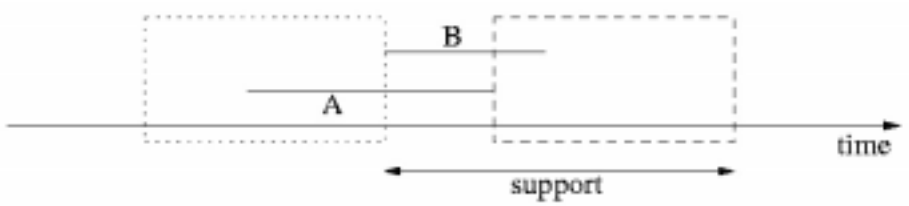


Abbildung 5: Support

Mit diesen Voraussetzungen ist der Apriori-Algorithmus nach [1] anwendbar.

6.2 Algorithmus Apriori [1]

1. Ermittle den Support aller 1-Muster.
2. Im k-ten Lauf: entferne alle Muster P mit $\mathsf{supp}(\mathsf{P}) < \mathsf{supp}_{min}$
3. Generiere aus den verbliebenen k-Mustern eine Menge von Kandidaten für die k+1-Muster
4. Ermittle den Support der Kandidaten im nächsten Lauf
5. Wiederhole diese Schritte, bis keine häufigen Muster mehr gefunden werden können

Im Anschluss können leicht die Regeln aus den häufigen Mustern abgeleitet werden.

7 Fazit

Wir haben gesehen, daß man Zeitreihen in sehr unterschiedlichen Domains bzw. Quellen finden kann - sogar eine Extraktion aus Dokumenten (z.B. Internetseiten) ist denkbar. Gerade in der *Projektgruppe 402* mit der Domain *Versicherungen* sind Zeitreihen und deren Analyse von hohem Interesse. Weiterhin haben wir die sich aus den verschiedenden Zeitphänomen ergebenden Lernaufgaben gesehen, von denen wir zwei in Detail kennengelernt haben.
Meiner Meinung nach ist das Thema *Zeitreihen* ein interessantes Forschungsgebiet, in dem sehr viel Potential steckt. Beispielsweise ist in der KDD ist der Einsatz der Verfahren sehr sinnvoll, da in vielen Datenbanken sehr viel Information in Form von Zeitreihen verborgen ist. Diese Information könnte mit Hilfe von einfachen Regeln dem Benutzer sehr viel übersichtlicher präsentiert werden. Das Problem besteht meiner Meinung nach darin, hier Zeitreihen als solche zu identifizieren und für eine Analyse aufzubereiten, um dann interessante Zeitreihen zu identifizieren.

Literatur

[1] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. AAAI Press/The MIT Press, Cambridge Massachusetts, London England, 1996.

[2] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[3] Tim Berners-Lee, James Hendler, and Ora Lassila. *The Semantic Web*, 2001.

[4] Uwe M. Borghoff and Remo Pareschi. *Information Technology for Knowledge Management.* Springer Verlag, 1998.

[5] Peter Brockhausen and Katharina Morik. Direct access of an ILP algorithm to a database management system. In Bernhard Pfaringer and Johannes Fürnkranz, editors, *Data Mining with Inductive Logic Programming (ILP for KDD)*, MLnet Sponsored Familiarization Workshop, pages 95–110, Bari, Italy, jul 1996.

[6] Hamish Cunningham. *Informations Extraction a User Guide.* Institute for Language, Speech and Hearing (ILASH) and Department of Computer Sciene Universtity of Sheffield, UK, apr 1999. `http://www.dcs.shef.ac.uk/~hamish`.

[7] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule Discovery from Time Series. In Rakesh Agrawal, Paul E. Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 16 – 22, Ney York City, 1998. AAAI Press.

[8] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, and Murali Venkatrao. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29 – 54, 1997.

[9] Tom Gruber. *What is an Ontology*, 2001.

[10] Harmelen, Patel-Schneider, and Horrocks. *Reference Description of the DAML+OIL ontologie markup language*, march 2001.

[11] Frank Höppner. Learning temporal rules from state sequences. In Miroslav Kubat and Katharina Morik, editors, *Workshop notes of the IJCAI-01 Workshop on* Learning from Temporal and Spatial Data, pages 25–31, Menlo Park, CA, USA, 2001. IJCAI, AAAI Press. Held in conjunction with the International Joint Conference on Artificial Intelligence (IJCAI).

[12] Heikki Mannila, Hannu Toivonen, and A.Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–290, November 1997.

[13] A. Mädche, S. Staab, and R. Studer. *Ontology-based Information Extraction and Integration in DGfS/CL'99*, 1999.

[14] Katharina Morik. The representation race - preprocessing for handling time phenomena. In Ramon López de Mántaras and Enric Plaza, editors, *Proceedings of the European Conference on Machine Learning 2000 (ECML 2000)*, volume 1810 of *Lecture Notes in Artificial Intelligence*, Berlin, Heidelberg, New York, 2000. Springer Verlag Berlin.

[15] Katharina Morik, Volker Klingspor, and Michael Kaiser. *Making Robots Smarter – Combining Sensing and Action through Robot Learning.* Kluwer Academic Press, 1999.

[16] W3C. *Resource Description Framework (RDF) Schema Specification 1.0.*

[17] Stefan Wrobel. An algorithm for multi–relational discovery of subgroups. In J. Komorowski and J. Zytkow, editors, *Principles of Data Minig and Knowledge Discovery: First European Symposium (PKDD 97)*, pages 78–87, Berlin, New York, 1997. Springer.