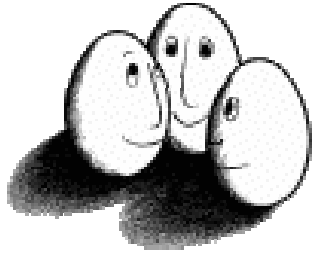


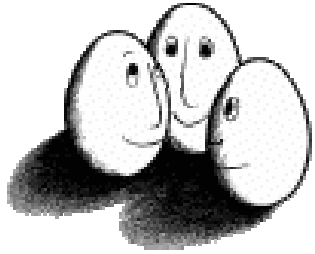
Zeitaspekte

Fabian Bauschulte



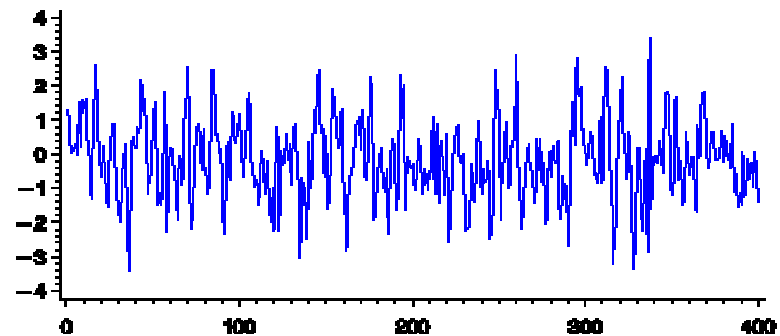
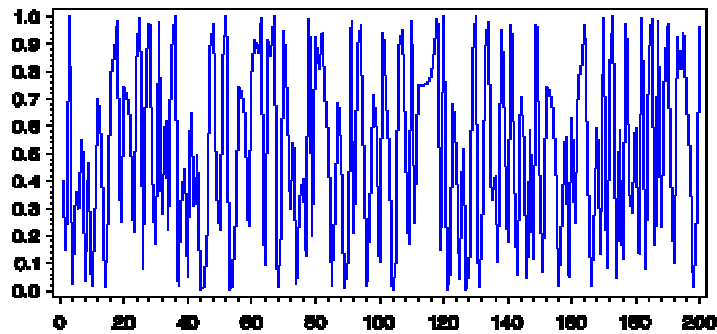
Zeitaspekte – Überblick

- Zeitreihen (engl. time series)
 - Zeitphänomene
 - Lernaufgaben und Repräsentation der Eingabedaten
 - Clustering [Das et al.]
 - Beziehungen zwischen Zeit-Intervallen lernen [Höppner]
- } [Morik/2000a]

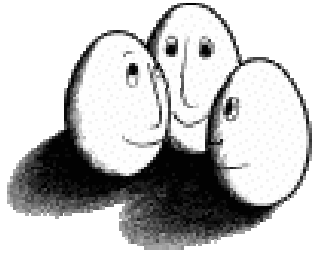


Beispiele für Zeitreihen (1/2)

- Messwerte von einem Prozess*
 - Intensivmedizin
 - Aktienkurse
 - Wetterdaten
 - Roboter [Morik/etal/99b]



Kontinuierliche Messung in z.B. Tagen, Stunden, Minuten, Sekunden

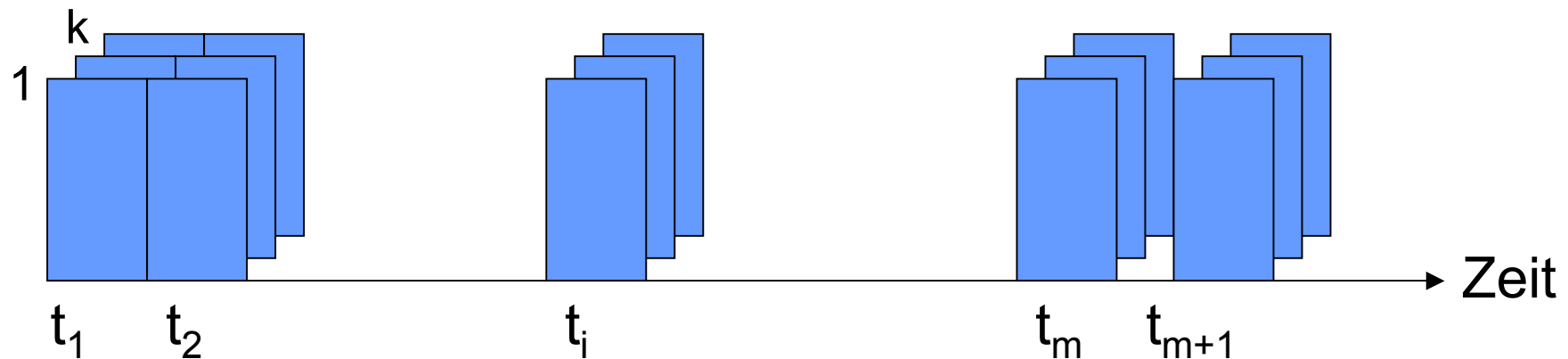


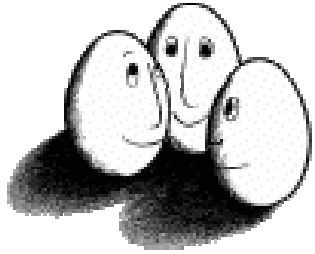
Univariat - Multivariat

Univariat - ein Attribut pro Zeit (Herzfrequenz)



Multivariat - k Attribute (Herzfrequenz, Atemfrequenz, Blutdruck)





Beispiele für Zeitreihen (2/2)

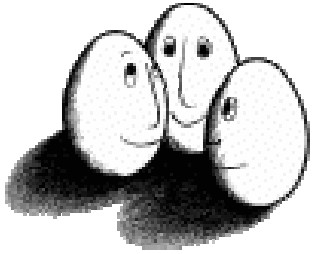
- **Datenbankrelationen**

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

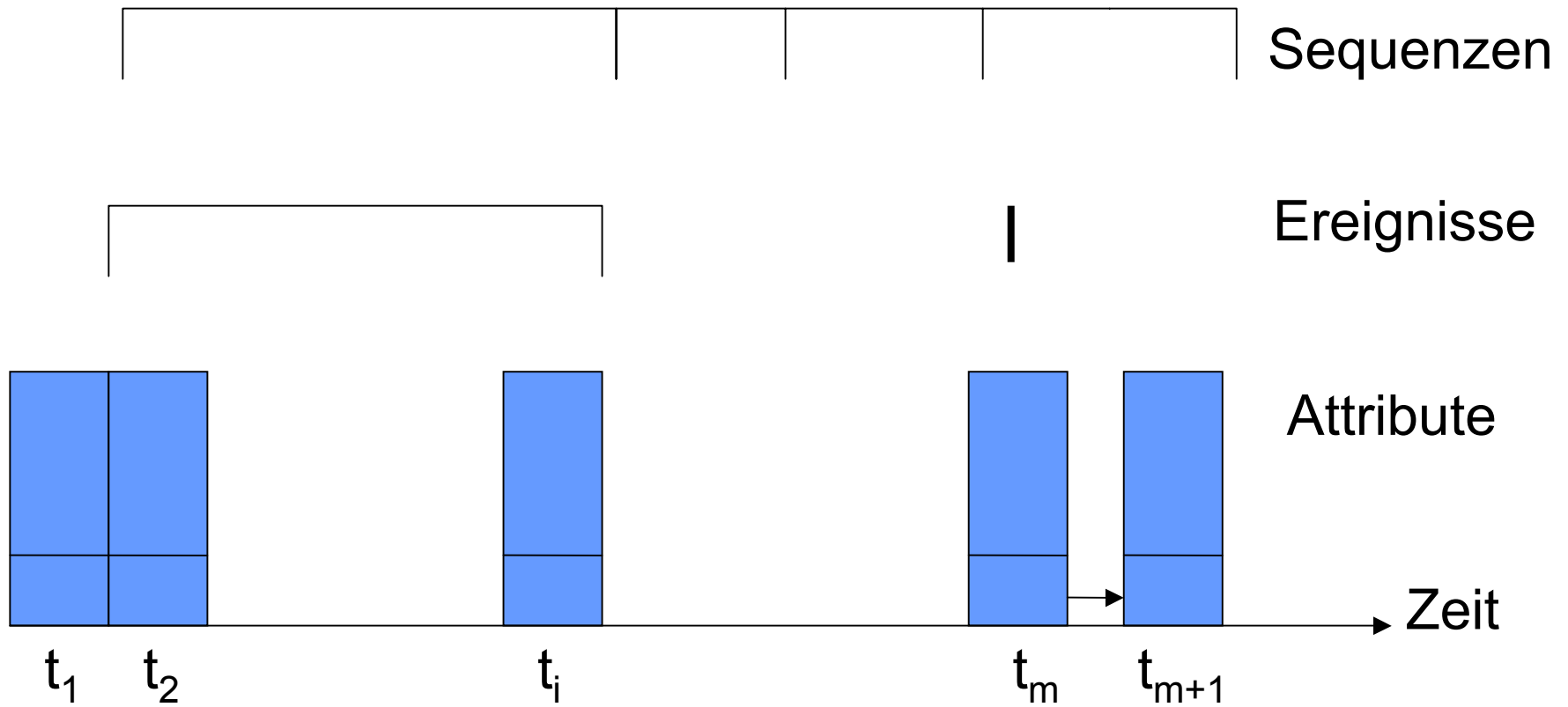
Verkäufe	Monat	Anzahl	Verkäufer	...
	Juni	256	Meier	...

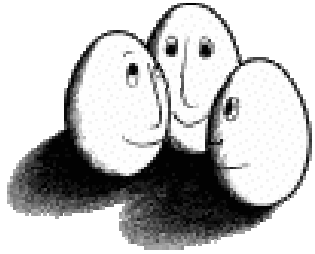


Ereignisse mit Zeitangaben in Jahren, Monaten, Tagen



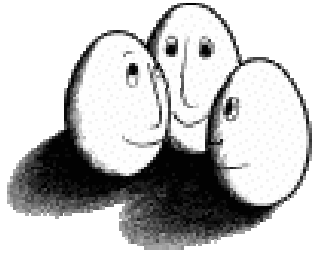
Zeitphänomene





Lernaufgaben (1/2)

- **Univariat**
 - Vorhersagen der $k+n$ -ten Beobachtung
 - einen allgemeinen Trend erkennen (alle Elemente steigen)
 - Lokale Trends finden (Zyklen, lokal steigende Werte)
 - Finde von einem Standard abweichende Werte (Ausreißer)
 - Clustering: Fasse ähnliche Bereiche von aufeinanderfolgenden Werten zu Clustern zusammen
- **Multivariat**
 - Finde zusammen auftretende Werte



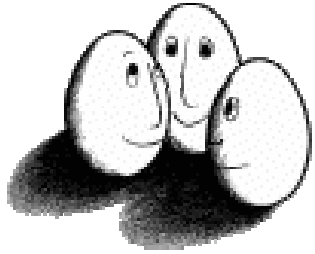
Repräsentation der Eingabedaten (1/2)

Multivariat:

$$i_l : \begin{aligned} &<t_1, a_{11}, \dots, a_{1k}> \\ &<t_2, a_{21}, \dots, a_{2k}> \\ &\dots \\ &<t_i, a_{i1}, \dots, a_{ik}> \end{aligned}$$

Univariat:

$$i_l : \begin{aligned} &<t_1, a_1> \\ &<t_2, a_2> \\ &\dots \\ &<t_i, a_i> \end{aligned}$$

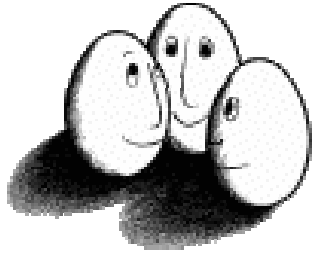


Lernaufgaben (2/2)

Lernaufgaben bei einer **gegebenen Sequenz von Ereignissen**:

(Menge von Ereignissen in partieller Ordnung)

- Finde häufige Episoden in Sequenzen [Mannila et al.] (1)
 - Wenn A auftritt, dann tritt B in der Zeit T auf [Das et al.]
- Beziehungen zwischen Zeit-Intervallen lernen (2)
[Höppner]
 - A startet vor B, B und C sind gleich



Repräsentation der Eingabedaten (2/2)

Ein Ereignis ist ein Tripel (Zustand, Start, Ende).

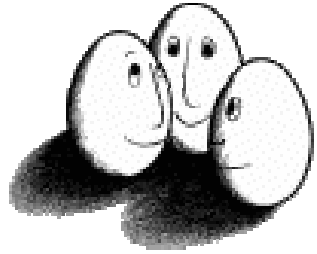
Der Zustand kann ein Wert oder ein Label (Trend bzw. eine Eigenschaft) sein.

Beispiele.: (Steigend, 3, 5); (Fallend, 7, 9); (Stabil, 10, 14)

- Möglichkeiten der Darstellung

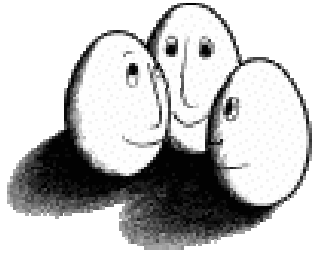
- Sequenz Vektor: $I : T_1 A_1, \dots, T_i A_i$ (1)

- Fakten: $P(I_1, T_b, T_e, A_r, \dots, A_s)$ (2)



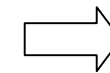
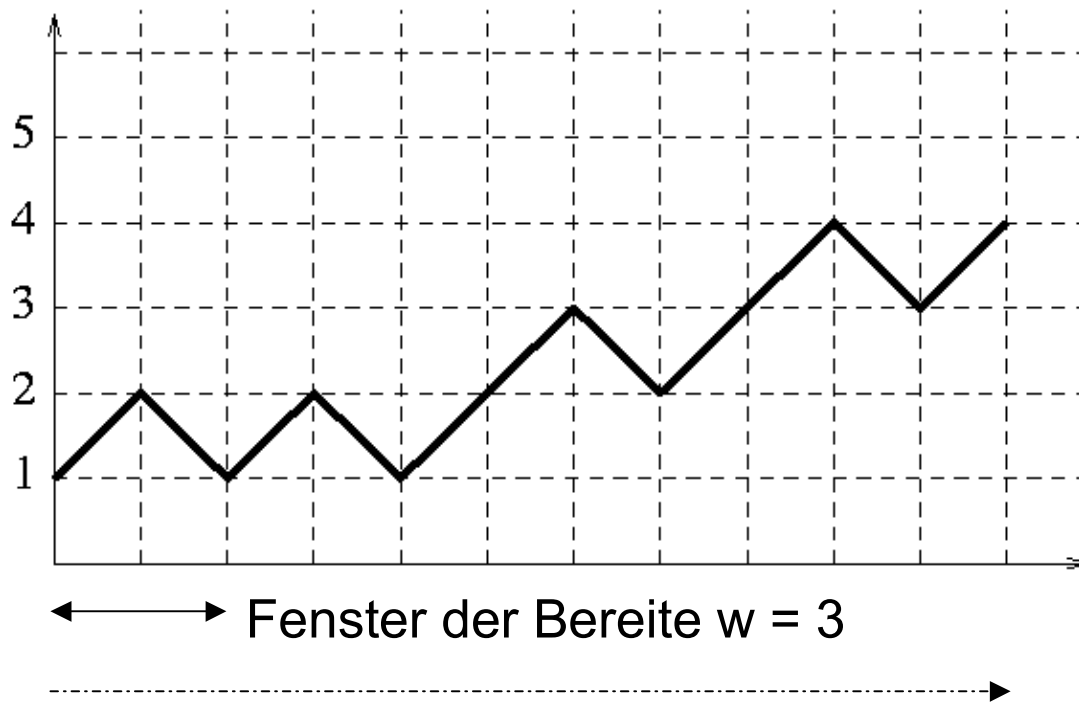
Problem: Wie finde ich die Ereignisse in Zeitreihen?

- Fenster fester Länge w
 - vorgebende oder erlernte Muster
 - Problem: w ist abhängig vom Anwendungsfall
- inkrementelle Analyse der Zeitreihe nach vorgegebenen Mustern [Morik/etal/99b]
 - Bsp.: Roboter
 - Vorteil: Dynamische Länge
- Diskretisierung, z.B. mittels Clustering [Das et al.]

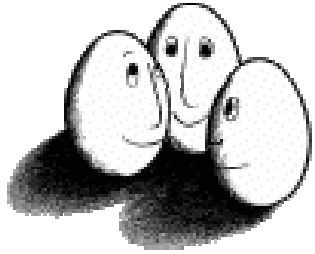


Clustering - (1/3) [Das et al.]

Zeitreihe $s = (x_1, \dots, x_n)$ in Subsequenzen $s_i = (x_i, \dots, x_{i+w-1})$ aufteilen



Schritt 2




Clustering - (2/3)


Distanzmaß $d(s_i, s_j)$: Entfernung zwischen zwei Subsequenzen

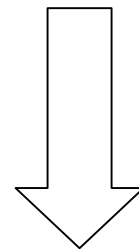
Bsp.: Euklidischer Abstand $(\sum(x_i - y_i)^2)^{0,5}$

Konstante $d > 0$: gibt an, wie groß der Unterschied zwischen den Subsequenzen sein darf

a1= 

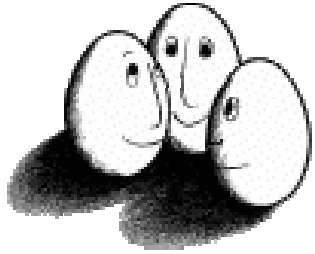
a2= 

a3= 



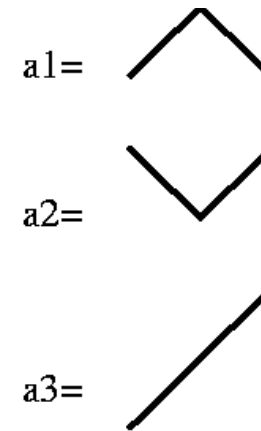
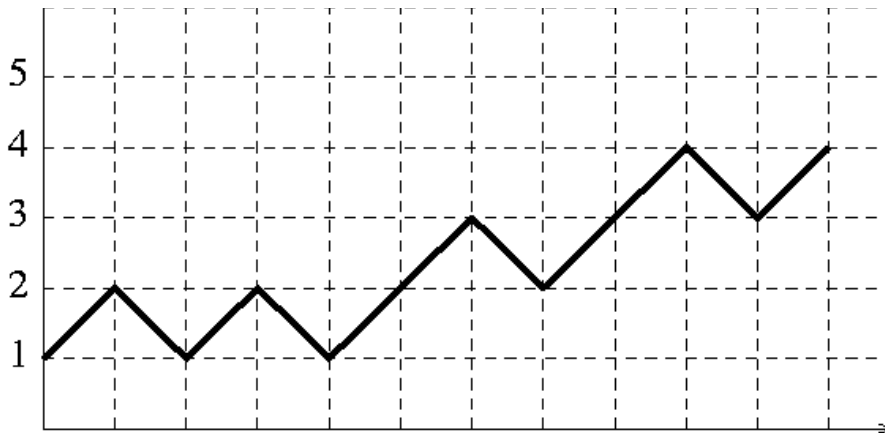
Bilde aus der Menge aller Subsequenzen
Cluster C_1, \dots, C_k

Jeder Cluster erhält ein Symbol a_1, \dots, a_k („Shapes“)



Clustering - (3/3)

Die Serie $s = (x_1, \dots, x_n)$ kann jetzt mit Hilfe der shapes beschrieben werden („diskretisiert“)

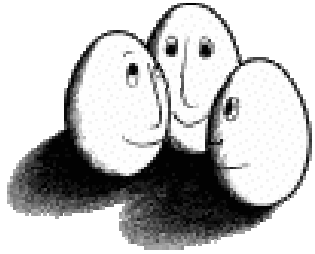


Original time series = (1, 2, 1, 2, 1, 2, 3, 2, 3, 4, 3, 4)

Window width = 3

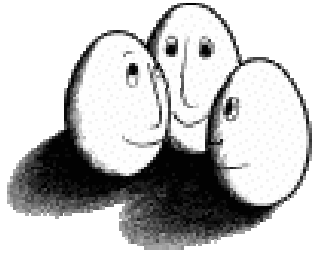
Discretized series = (a1, a2, a1, a2, a3, a1, a2, a3, a1, a2)

Primitive shapes after clustering



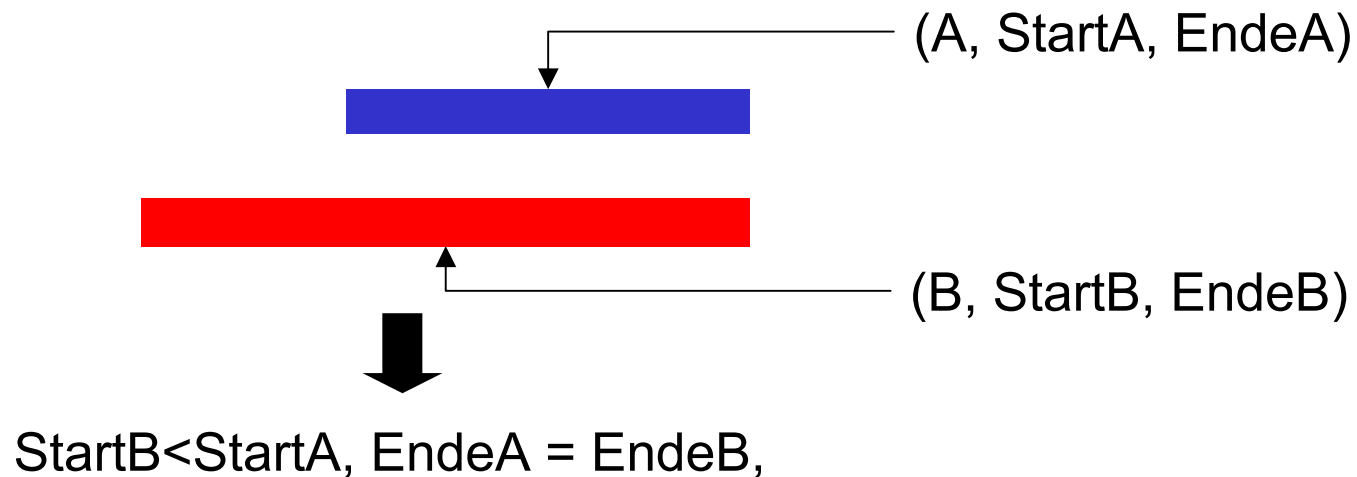
Regeln in diskreten Sequenzen

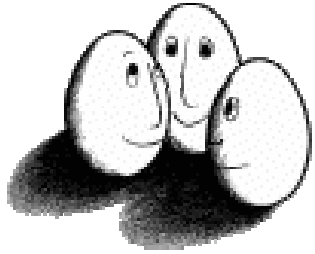
- Regeln der Form
 Wenn A auftritt, dann tritt B in der Zeit T auf einfach ableitbar
- Berechnung in der Zeit $m \cdot k^2$ möglich
 - (k =Anzahl der Symbole, m = #verschiedene Möglichkeiten für T)
- Erweiterung:
 - Wenn A_1 und A_2 und ... und A_n innerhalb der Zeit V auftritt, dann tritt B in der Zeit T auf
 - **Microsoft** ↓ (1), **Microsoft** ↑ (2) + **Intel** → (2) ⇒ **IBM** → (3)
 - Problem: Anzahl der Regeln steigt stark an



Beziehungen zwischen Ereignissen

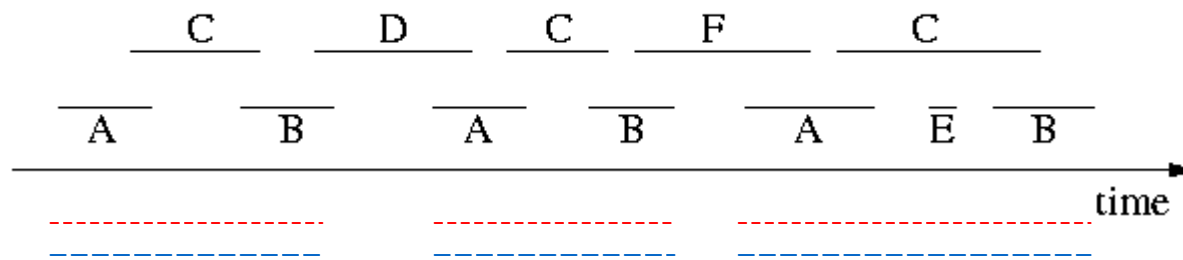
- Von James F. Allen wurden 13 verschiedene Intervallbeziehungen festgelegt:
 - A überlappt B, A beendet B, A vor B, A enthält B, ...
- Bsp.: A beendet B



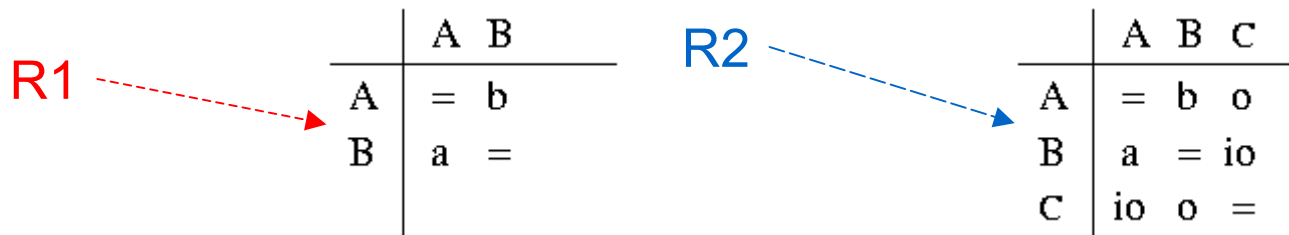


Beziehungen zwischen Zeit-Intervallen lernen [Höppner]

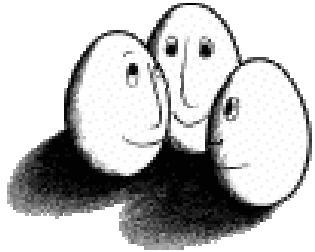
state interval sequence:



Darstellung der Beziehungen als Matrix:

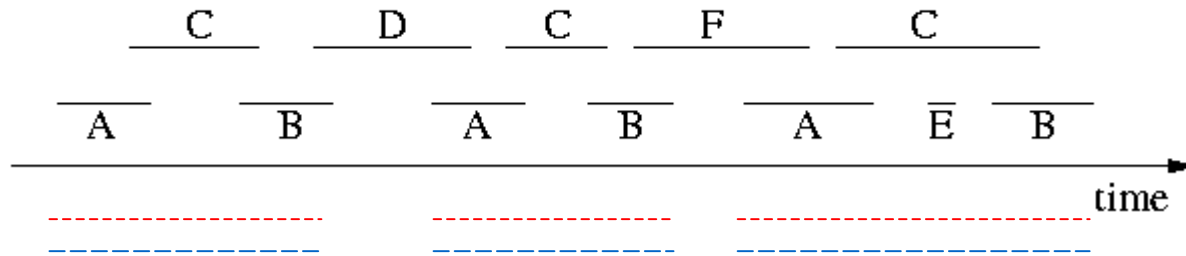


(abbreviations: a=after, b=before, o=overlaps, io=is-overlapped-by)



Wie sehen hier die Regeln aus?

state interval sequence:



Die Regeln sind von der Form $P \rightarrow R$

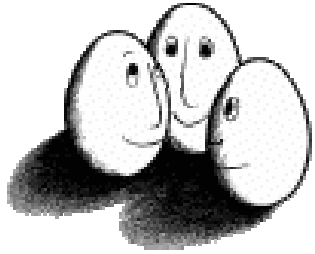
Prämisse P

	A	B
A	=	b
B	a	=

Regel R

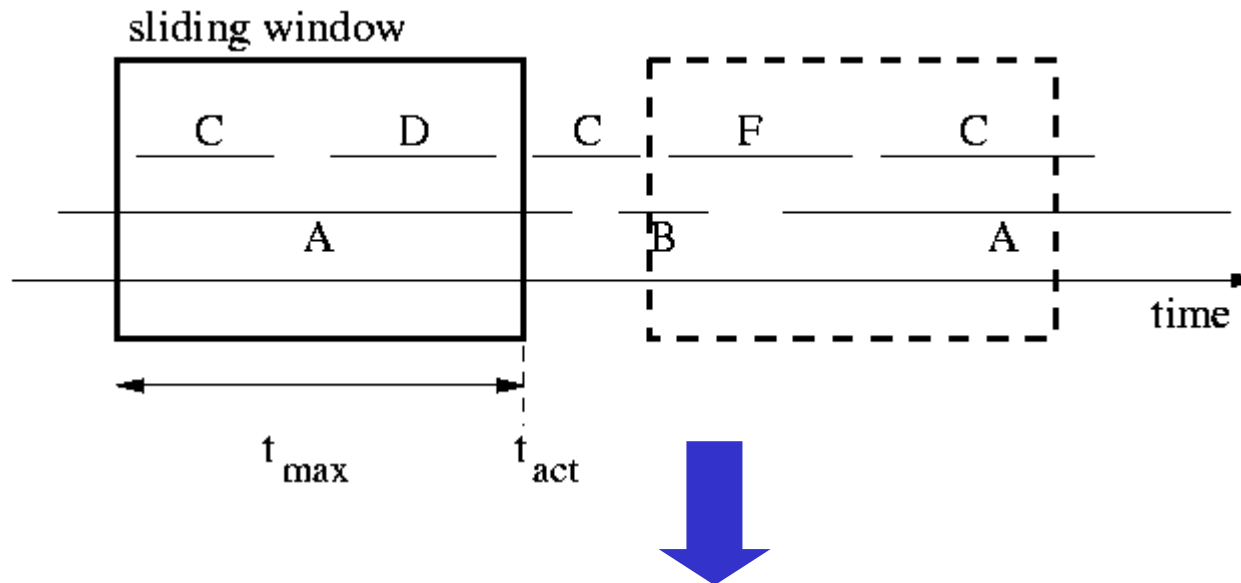
	A	B	C
A	=	b	o
B	a	=	io
C	io	o	=

Beispiel: A, B, C sind Verträge verschiedener Kategorien

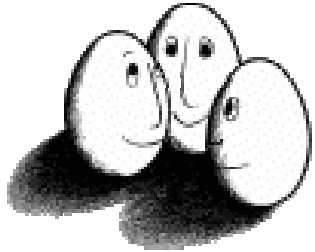


Häufige Muster finden

Muster muss im Fenster der Länge t_{\max} beobachtbar sein

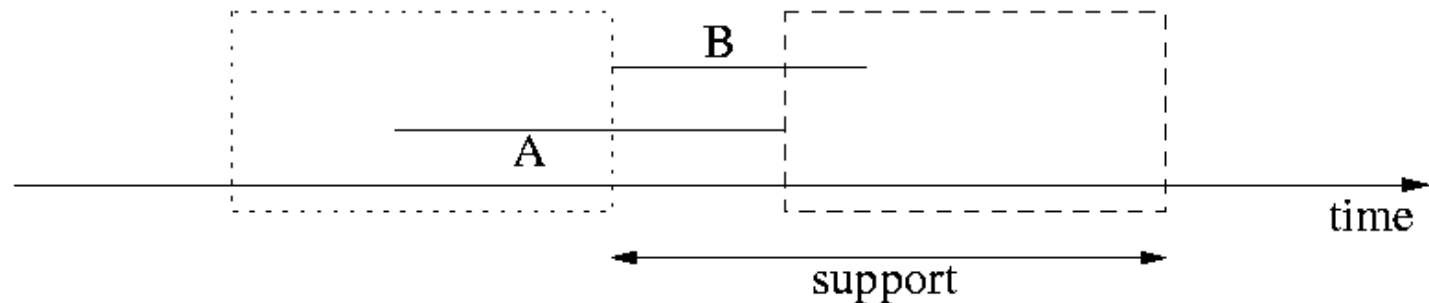


Der maximale Abstand zwischen den Ereignissen eines Muster ist begrenzt



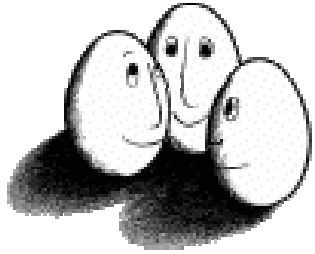
Was bedeutet häufig?

Als Maß für die Häufigkeit von Mustern dient der „Support“



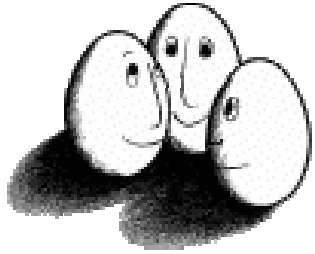
Ein Muster wird als häufig erachtet, wenn es einen Support $> \text{supp}_{\min}$ hat

	A	B
A	=	0
B	1	=



Algorithmus [Agrawal et al.]

- Ermittle den Support aller 1-Muster
- Im k -ten Lauf:
 - entferne alle Muster mit $\text{supp} < \text{supp}_{\min}$
 - generiere aus den verbliebenen k -Mustern eine Menge von Kandidaten für $k+1$ -Muster
 - ermittle den Support der Kandidaten im nächsten Lauf
- Wiederhole diese Schritte, bis keine häufigen Muster mehr gefunden werden können
- Generiere die Regeln aus den häufigen Mustern



Literatur

- [Morik/2000a] Katharina Morik. The representation race - preprocessing for handling time phenomena. In Ramon L´opez de M´antaras 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.
- [Morik/etal/99b] Katharina Morik, Volker Klingspor, and Michael Kaiser. Making Robots Smarter – Combining Sensing and Action through Robot Learning. Kluwer Academic Press, 1999.
- [Mannila et al.] 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.
- [Das et al.] 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.
- [Agrawal et al.] 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.