Applied Analytical Data Science
Teil 7: Regression (forts.), Assoziationsregelverfahren

Dr. Jörg-Uwe Kietz,
Vorlesung an der Univ. Zürich,

Mittwoch, 14:00-15:45 Uhr Vorlesung,
16:00-17:30 Uhr Übung

http://www.kietz.ch/AADS/

Heutiges Programm

- Regression
  - lineare Regression
  - Regressions- & Modellbäume
  - andere regressionstaugliche Verfahren

- Assoziationsregeln
  - Definition, Anwendungen, Beispiel
  - Apriori-Verfahren
  - Assoziationsregeln & Hierarchien
  - Assoziationsregeln & Sequenzen

Definition Klassifikation und Regression

Gegeben:

Eine Menge von Beispielen, beschrieben durch

- Eingabeattribute, und
- ein Zielattribut: nominal (Klassifikation)
  numerisch (Regression)

Gesucht:

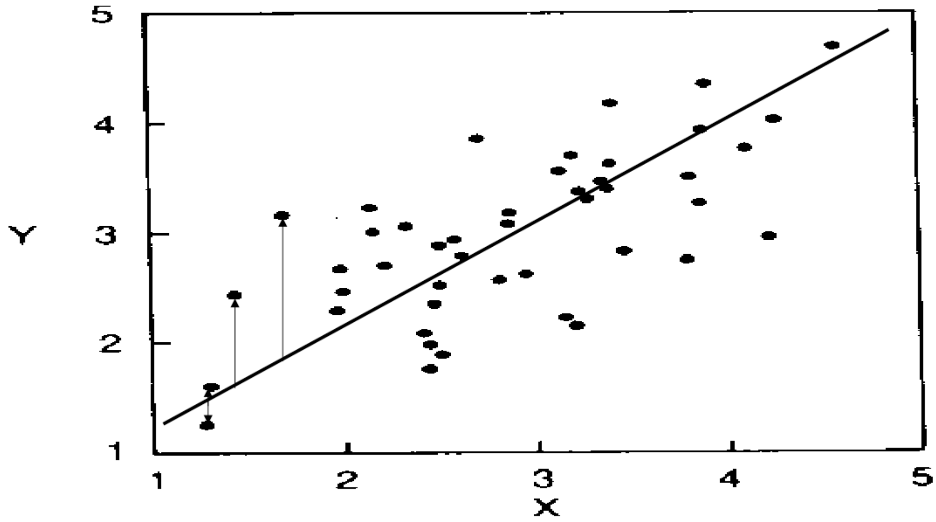
Eine Funktion (Baum, Regeln, Funktion, Neuronales Netz ...):

- zur Berechnung/Vorhersage des Zielattributes, für jedes neue, nur durch die Eingabeattribute beschriebene Beispiel.

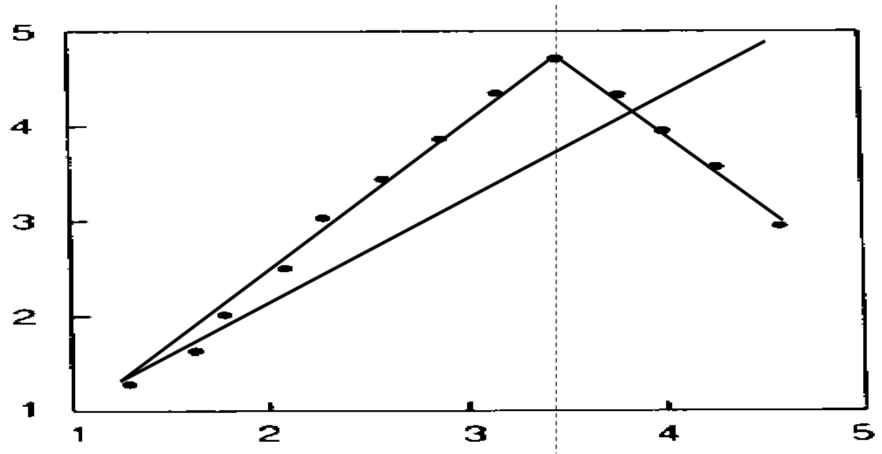
Lineare Regression

- Bestimmung der besten Hyperebene, d.h. Bestimmung der Gewichte $w_0, \ldots, w_k$, so dass der
  Zielwert $x^{(i)}$ eines Beispiels i aus den
  Attributen $a_1, \ldots, a_k$ berechnet werden kann
  $x^{(i)} = w_0{}^{(i)} + w_1 a_1{}^{(i)} + \ldots + w_k a_k{}^{(i)} = \Sigma_{j=0}{}^k \; w_j a_j{}^{(i)}$
- Die Gewichte ergeben sich als Lösung des Optimierungsproblems zur Minimierung des quadratischen Fehlers über allen Trainingsbeispielen
  Minimierung von: $\Sigma_{i=1}{}^n \left(x^{(i)} - \Sigma_{j=0}{}^k \; w_j a_j{}^{(i)} \right)^2$

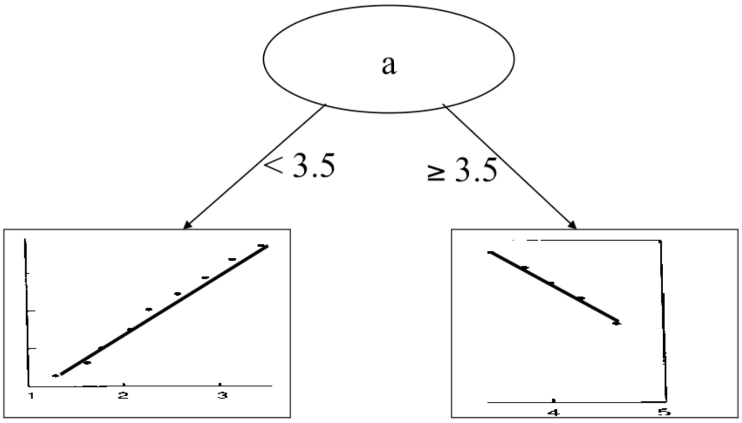
Lineare Regression



Lineare Modelle können zu einfach sein



Lösung: Modellbäume



Modell und Regressionsbäume

- Ein **Regressionsbaum** ist ein Entscheidungsbaum an dessen Blättern ein **konstanter Wert** für ein **skalares Zielattribut** vorhergesagt wird.

- Ein **Modellbaum** ist ein Entscheidungsbaum an dessen Blättern eine **lineare Regression** zur Berechnung des **skalaren Zielattributs** benutzt wird.

- Die Attributauswahl beim Bilden der Bäume basiert auf der **Fehlerminimierung** (oder vereinfacht Minimierung der Standard Abweichung) statt auf Informationgain durch Wahl des Attributes.

Andere regressionstaugliche Verfahren

- Instance based learning (IBL):
  Vorhersage des Durchschnitts der k nächsten Nachbarn.

- Support vector machine (SVM):
  Minimierung des Fehlers statt Maximierung des Abstands, ergibt lineare Regression mit Kernfunktionen für nicht-lineare Aufgaben.

- Neuronale Netze:
  direkte Interpretation der Ausgabeeinheit ergibt eine nicht-lineare Regression.

Heutiges Programm

- Regression
  - lineare Regression
  - Regressions- & Modellbäume
  - andere regressionstaugliche Verfahren

- Assoziationsregeln
  - Definition, Anwendungen, Beispiel
  - Apriori-Verfahren
  - Assoziationsregeln & Hierarchien
  - Assoziationsregeln & Sequenzen

Assoziationsregeln Definition

Gegeben:

- Eine Menge von Transaktionen
- Eine minimale Sicherheit (confidence)
- Eine minimale Unterstützung (support)

Gesucht:

- Alle maximalen Assoziationsregeln, mit
- höherer Sicherheit, und
- höherer Unterstützung

Assoziationsregeln Aufgaben

Aufgaben:

- Warenkorbanalyse, oder
  - Was wird zusammen gekauft?
  - Cross-selling, Sonderangebotsplannung, ...

Im Allgemeinen:

- Finde häufige Muster, z.B.:
  - Produkt / Kunden Muster
  - Sequenzmuster in Zeitreihen, Texten, ...

Beispiel Assoziationsregeln

Einige Transaktionen

1: cola, beer, juice

2: beer, wine

3: juice, beer

4: cola, juice, wine

5: juice, beer, wine

Was wurde zusammen gekauft?

cola → juice	C:100%	S:40%
juice → beer	C:75%	S:60%
beer → juice	C:75%	S:60%
wine → beer	C:66%	S:40%
wine → juice	C:66%	S:40%
cola, beer → juice	C:100%	S:20%
cola, wine → juice	C:100%	S:20%

...

Assoziationsregeln

- Gegeben ist eine Menge **D** von Transaktionen **t**. Jedes **t** besteht aus einer Menge von Items **I** (Atome: Binäres Attribute) $\mathbf{t} \subseteq \mathbf{I}$
- Assoziationregel: $\mathbf{X} \rightarrow \mathbf{Y}$, mit $X \subseteq \mathbf{I}$ und $Y \subseteq \mathbf{I}$
- Eine Transaktion **t** erfüllt eine Regel $\mathbf{X} \rightarrow \mathbf{Y}$, falls $(\mathbf{X} \cup \mathbf{Y}) \subseteq \mathbf{t}$
- **Support** ist der Anteil der Transaktionen, der die Regel erfüllt:

$$\text{support}(X \rightarrow Y) = \frac{|\{t \in D \mid (X \cup Y) \subseteq t\}|}{|D|}$$

- **Confidence** misst die Korrektheit der Regel:

$$\text{confidence}(X \rightarrow Y) = \frac{|\{t \in D \mid (X \cup Y) \subseteq t\}|}{|\{t \in D \mid X \subseteq t\}|} = \frac{\text{support}(X \rightarrow Y)}{\text{support}(X)}$$

Def: Assoziationsproblem

- Definition (**Assoziationsproblem**):
  - gegeben: Menge D von Transaktionen
    Wert für minimalen Support s_{min}
    Wert für minimale Konfidenz c_{min}
  - gesucht: Finde alle Assoziationsregeln $X \rightarrow Y$, so daß
    $\text{support}(X \rightarrow Y) \geq s_{min}$
    $\text{confidence}(X \rightarrow Y) \geq c_{min}$

Heutiges Programm

- Regression
  - lineare Regression
  - Regressions- & Modellbäume
  - andere regressionstaugliche Verfahren

- Assoziationsregeln
  - Definition, Anwendungen, Beispiel
  - Apriori-Verfahren
  - Assoziationsregeln & Hierarchien
  - Assoziationsregeln & Sequenzen

Apriori Algorithmus

Idee:

Schritt 1: Bestimme Itemmengen mit vorgegebenem Mindestsupport

=> **häufige Itemmengen**

Schritt 2: Sei X häufige Itemmenge, dann gilt für $X' \subseteq X$:

$$\text{confidence}((X - X') \rightarrow X') = \frac{\text{support}(X)}{\text{support}(X - X')}$$

d.h. die Konfidenz läßt sich aus dem Support der häufigen Itemmengen berechnen.

Bilde alle **Assoziationsregeln** aus häufigen Itemmengen mit vorgegebener Mindestkonfidenz.

Berechnung der häufigen Itemmengen

- für $X' \subseteq X$ gilt: $\text{support}(X') \geq \text{support}(X)$
  => wenn X häufige Itemmenge ist, dann auch X'.
- berechne sukzessive die häufigen Itemmengen mit 1,2,3,... Elementen.
- fasse die häufigen Itemmengen mit i Elementen zur Menge I_i zusammen: $I_i := \{X \mid X \text{ ist häufige Itemmenge}, |X| = i\}$
- für $X \in I_{n+1}$ gilt: alle n-elementigen Teilmengen von X sind häufig.
- Berechnung von I_{n+1} aus I_n:
  - n-elementige **häufige** Mengen werden um jeweils 1 Element erweitert.
  - prüfe, ob (n+1)-elementige Menge häufig ist: wenn ja, nehme (n+1)-elementige Menge in I_{n+1} auf.

Beispiel Apriori

Einkaufs-Transaktionen	gekaufte Artikel (Items)
t_1	Saft, Cola, Bier
t_2	Saft, Cola, Wein
t_3	Saft, Wasser
t_4	Cola, Bier, Saft
t_5	Saft, Cola, Bier, Wein
t_6	Wasser

$\Downarrow$

Artikel (Item)	Transaktionen, in denen der Artikel vorkommt	Support des Items
Saft	t_1, t_2, t_3, t_4, t_5	5/6 = 83.3%
Cola	t_1, t_2, t_4, t_5	4/6 = 66.6%
Bier	t_1, t_4, t_5	3/6 = 50 %
Wein	t_2, t_5	2/6 = 33.3%
Wasser	t_3, t_6	2/6 = 33.3%

Beispiel Aprioi: häufige Itemmengen

Artikel (Item)	Transaktionen, in denen der Artikel vorkommt	Support des Items
Saft	t_1, t_2, t_3, t_4, t_5	5/6 = 83.3%
Cola	t_1, t_2, t_4, t_5	4/6 = 66.6%
Bier	t_1, t_4, t_5	3/6 = 50 %
Wein	t_2, t_5	2/6 = 33.3%
Wasser	t_3, t_6	2/6 = 33.3%

$\Downarrow$

n	H_n	I_n
1	{{Saft},{Cola},{Bier},{Wein},{Wasser}}	{{Saft},{Cola},{Bier}}
2	{{Saft, Cola},{Saft, Bier},{Cola, Bier}}	{{Saft, Cola},{Cola, Bier}, {Saft, Bier}}
3	{{Saft, Cola, Bier}	{{Saft, Cola, Bier}}
4	{}	{}

Beispiel Aprioi:Von Itemmengen zu Regeln

zu Schritt 2:

- gegeben: c_{min} = 75%
- wähle z.B. {Saft, Cola} $\in I_2$

$$\text{confidence}(\text{Saft} \rightarrow \text{Cola}) = \frac{\text{support}(\{\text{Saft}, \text{Cola}\})}{\text{support}(\{\text{Saft}\})} = \frac{2/3}{5/6} = 80\%$$

$$\text{confidence}(\text{Cola} \rightarrow \text{Saft}) = \frac{\text{support}(\{\text{Saft}, \text{Cola}\})}{\text{support}(\{\text{Cola}\})} = \frac{2/3}{2/3} = 100\%$$

Algorithmus für häufige Itemmengen

1. Initialisierung:
   s_{min} = Wert für minimalen Support;
   $n := 1$;
   $I := \emptyset$;
   $H_n := \{\{i\} \mid i \text{ ist ein Item}\}$;
2. gehe über die Datenbasis D und bestimme für alle $H \in H_n$ den Support;
3. $I_n := \{H \in H_n \mid \text{support}(H) \geq s_{min}\}$;
   $I := I \cup I_n$;
4. Falls $I_n = \emptyset$, gebe I als Ergebnis aus;
5. $H_{n+1} := \{\{i_1,i_2,\ldots,i_{n+1}\} \mid \forall j: 1 \leq j \leq n+1: (\{i_1,i_2,\ldots,i_{n+1}\} - \{i_j\}) \in I_n\}$;
   $n := n+1$;
6. gehe nach 2.

Gefundene Assoziationsregeln

Fortführung des Beispiels: abgeleitete Assoziationsregeln zu den häufigen Itemmengen

Regeln mit Support ≥ 50%	**erfüllende Transaktionen**	**Support**	**Konfidenz**
Saft → Cola	t_1, t_2, t_4, t_5	66%	80%
Cola → Saft	t_1, t_2, t_4, t_5	66%	100%
Cola → Bier	t_1, t_4, t_5	50%	75%
Bier → Cola	t_1, t_4, t_5	50%	100%
Saft → Bier	t_1, t_4, t_5	50%	60%
Bier → Saft	t_1, t_4, t_5	50%	100%
Saft, Bier → Cola	t_1, t_4, t_5	50%	100%
Cola, Saft → Bier	t_1, t_4, t_5	50%	75%
Bier, Cola → Saft	t_1, t_4, t_5	50%	100%
Cola → Saft, Bier	t_1, t_4, t_5	50%	75%
Bier → Cola, Saft	t_1, t_4, t_5	50%	100%
Saft → Cola, Bier	t_1, t_4, t_5	50%	60%

Zusammenfassung Apriori

- sofern häufige Itemmengen mit n Elementen existieren, benötigt der Algorithmus n Durchläufe

  => 1 Megabyte Warenkorb-Daten im Sekundenbereich analysiert

- Algorithmus entdeckt Tausende von Regeln:
  - **Visualisierung, Browsing** notwendig
  - Regeln mit sehr hohen Support-/Konfidenzwerten schon bekannt
  - Regeln im "**Mittelfeld**" interessant
  - verwende zusätzlich statistische Maße für Bewertung von Regeln
- Regeln berücksichtigen viele **externe Einflußfaktoren** nicht, da diese in den Transaktionen nicht repräsentiert sind:
  - Käufergruppen (Alter, sozialer Status, ...)
  - Tageszeit, Wochentag
  - Werbekampagnen

Heutiges Programm

- Regression
  - lineare Regression
  - Regressions- & Modellbäume
  - andere regressionstaugliche Verfahren

- Assoziationsregeln
  - Definition, Anwendungen, Beispiel
  - Apriori-Verfahren
  - Assoziationsregeln & Hierarchien
  - Assoziationsregeln & Sequenzen

Assoziationsregeln & Hierarchien

- Regel können durch Hintergrundwissen verallgemeinert werden
  - verwende **Taxonomien** über Items zur Bildung von Warengruppen,

    z.B. Bier *is-a* alkoholisches Getränk

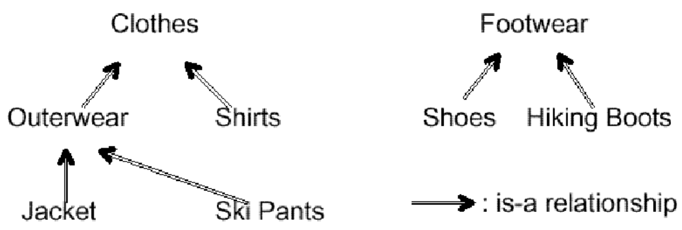
    Chips *is-a* Salzgebäck
  - damit können abstraktere Regeln gebildet werden,

    z.B. (Salzgebäck → Bier)

    (Chips → alhoholisches Getränk)
- benötigtes Wissen kann durch **Ontologien** und **Domänenmodelle** bereitgestellt werden

Beispiele für Hierachien

Example: Taxonomy for Clothes and Footwear (Srikant/Agrawal 1995)



Apriori-Erweiterung für Hierarchien

Basis-Idee zum Lernen von Assoziationen mit Hierarchien:

(1) Erweiterung (saturierung) der Transaktionen: Transaktion mit **Jacket** => Transaktion mit **Outerwear** und **Clothes**

(2) Benutze Frequent-Itemsets (Apriori) auf den erweiterten Transaktionen.

(3) Benutze Generate-Rules (Apriori) um alle Regeln mit minimum Support und Confidence zu generieren.

=> nicht alle Regeln sind interesant!!!

(4) Lösche alle uninteresanten (insb. die redundanten) Regeln

Saturierung der Itemsets

- transaction t **supports** an itemset X, if for each item $x \in X$ t contains x or some sub-concepts of x:
  - check becomes simpler if we first add to t all (direct/indirect) super-concepts of x; we call the resulting transaction t‘ an **extended** transaction.
  - **t supports X** iff[1]
    t‘ is a superset of X.

Itemsets und Hierarchien

Example: (Srikant/Agrawal 1995)

- given database D:

Database $\mathcal{D}$

Transaction	Items Bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

- assume: minimum support = 30%, minimum confidence = 60%

Frequent Itemsets

Itemset	Support
{ Jacket }	2
{ Outerwear }	3
{ Clothes }	4
{ Shoes }	2
{ Hiking Boots }	2
{ Footwear }	4
{ Outerwear, Hiking Boots }	2
{ Clothes, Hiking Boots }	2
{ Outerwear, Footwear }	2
{ Clothes, Footwear }	2

Rules

Rule	Support	Conf.
Outerwear ⇒ Hiking Boots	33%	66.6%
Outerwear ⇒ Footwear	33%	66.6%
Hiking Boots ⇒ Outerwear	33%	100%
Hiking Boots ⇒ Clothes	33%	100%

Beobachtungen zu hierarchischen Regeln (1)

- rules 'Ski Pants => Hiking Boots',
  'Jackets => Hiking Boots'

  do **not** have minimum support.

  - generalized rule 'Outerwear => Hiking Boots'
    may have minimum support.

- let $\hat{x}$ denote the (direct/indirect) super-concept of x:

  if 'x => y' has **minimum support**, so do
  'x => $\hat{y}$', '$\hat{x}$ => y' and '$\hat{x}$ => $\hat{y}$'.

Beobachtungen zu hierarchischen Regeln (2)

- rule ‘Outerwear => Hiking Boots‘ has minimum confidence
    - generalized rule may **not** have minimum confidence:
      ‘Clothes => Hiking Boots‘ has only confidence of 50%

- if ‘x => y‘ has **minimum confidence**, so does ‘x => $\hat{y}$’.

  rules '$\hat{x}$ => y' and '$\hat{x}$ => $\hat{y}$' may not have minimum confidence.

Beobachtungen zu hierarchischen Regeln (3)

- support for an item x is **not** equal to the sum of the supports of its sub-concepts:
  - several sub-concepts may be present in a **single** transaction

Pruning von redundanten Regeln

- assume ‘Outerwear => Hiking Boots‘ with support 8%, confidence 70%
- assume that 25% of ‘Outerwear‘-transactions are ‘Ski Pants‘-transactions
- we therefore **expect** a rule

  ‘Ski Pants => Hiking Boots‘

  with support 2% and confidence 70%.
- if such a rule is generated, the more special rule is **redundant** when compared to the more general rule.

Interessante Regeln

- let R be a user-specified minimum interest
- for a rule 'X => Y', we call the rules
  $'X \Rightarrow \hat{Y}', '\hat{X} \Rightarrow Y', '\hat{X} \Rightarrow \hat{Y}'$ its ancestors
- for a rule 'X => Y', we call the rule $'\hat{X} \Rightarrow \hat{Y}'$ its
  close ancestor, if there is no rule 'X' => Y'', such that

  - 'X' => Y'' is an ancestor of 'X => Y' and
  - $'\hat{X} \Rightarrow \hat{Y}'$ is an ancestor of 'X' => Y''.

- A rule 'X => Y' is **interesting** if
  - it has no ancestors or
  - it is R-interesting with respect to its close ancestors

Beispiele für interessante Regeln

Example: (Srikant/Agrawal 1995)

Rule #	Rule	Support
1	"Clothes ⇒ Footwear"	10
2	"Outerwear ⇒ Footwear"	8
3	"Jackets ⇒ Footwear"	4

Item	Support
Clothes	5
Outerwear	2
Jackets	1

- assume R=2
- rule 1 is **interesting**, since it has **no ancestors**.
- rule 2 is **interesting**, since its support is twice (**R times**) the expected support based on rule 1:

(i) $\frac{\text{support(Outerwear)}}{\text{support(Clothes)}} = \frac{2}{5}$

(ii) expected support('Outerwear => Footwear') =

$$\text{support('Clothes => Footwear')} \cdot \frac{2}{5} = 10 \cdot \frac{2}{5} = 4$$

however: support('Outerwear => Footwear') = 8 = R· 4

Beispiele für uninteressante Regeln

- rule 3 is **not interesting**, since its support is **equal** to the expected support based on rule 2:

(i) $\frac{\text{support(Jackets)}}{\text{support(Outerwear)}} = \frac{1}{2}$

(ii) expected support('Jackets => Footwear') =
support('Outerwear => Footwear') $\cdot \frac{1}{2} = 8 \cdot \frac{1}{2} = 4$

Remark:
measure of interest is able to prune between 40% and 60% of all generated rules.

Fazit zu Hierarchien und Assoziationen

- basic algorithm can be made **more efficient** by applying several optimization steps,
    - e.g. pre-compute all super-concepts of all available items

- taxonomy provides means for
    - generating **more useful** rules
    - **pruning uninteresting** rules

Heutiges Programm

- Regression
  - lineare Regression
  - Regressions- & Modellbäume
  - andere regressionstaugliche Verfahren

- Assoziationsregeln
  - Definition, Anwendungen, Beispiel
  - Apriori-Verfahren
  - Assoziationsregeln & Hierarchien
  - Assoziationsregeln & Sequenzen

Assoziationsregeln & Sequenzen

- Definition: Eine Sequenz-abhängige Assoziationsregel $A \Rightarrow^T B$ ist eine Assoziation über einer diskreten Sequenz, mit der Bedeutung: „Wenn A vorkommt, kommt B innerhalb von T Elementen vor.
- Der **Supportset(A,T)** einer Sequenz A ist die Menge aller Fenster der Grösse T, in denen die Elemente der Sequenz $A = A_1,\ldots,A_n$ in dieser Reihenfolge, aber nicht notwendigerweise aufeinanderfolgend, vorkommen.
- **Support($A \Rightarrow^T B$) = | Supportset((A,B),T) |**
- Die **Konfidenz($A \Rightarrow^T B$) = | Supportset((A,B),T) | / | Supportset(A,T) |**

Mining von Assoziationen über Sequenzen

- Das Apriori-Verfahren (=> Teil 4) kann fast unverändert zum Finden von sequenz abhängigen Assoziationsregeln benutzt werden, da Item-Mengen als sortierte Item-Listen implementiert werden.

=> Die Berücksichtigung der Sequenz (vs. Menge, insb. das Weglassen der Sortierung) macht Apriori zu Apriori-Sequenz.

Aber: Apriori ist exponentiell in der Anzahl der verschiedenen Items, d.h. der Aufwand von Apriori ist $O(2^{|Items|})$

- Apriori-Sequenz ist exponentiell in der Anzahl der möglichen Item-Sequenzen der Länge T, davon gibt es: $|Items|^T$

=> Der Aufwand von Apriori-Sequenz ist $O(2^{|Items|^T})$

Fazit

- Regression ist die Data Mining Methode zur Vorhersage von skalaren (kontinuierlichen) Werten.
  - Basis ist lineare Regression => Erweiterung zu Modellbäumen.
  - Auch IBL, SVM und Neuronale Netze können benutzt werden.
- Assoziations-Regel entdecken häufige Muster von zusmmen auftretenden Elementen in Transaktionen
  - Apriori ist der Basis Algorithmus dieses Effizient zu tun
  - Erweiterung zu Hierarchien über Elementen (Produktgrupppen)
  - Erweiterung zu Sequenzen von Elementen (Zeit, Text, ...)

Literatur

Data Mining Methoden für Regression und Assoziationsregeln:

- Witten, I.; Frank, E.: Practical Machine Learning Tools and Techniques with Java implementations, Morgan Kaufmann, 2000.

- K. Morik; S. Wrobel; T. Joachims: "Maschinelles Lernen und Data Mining" Beitrag zum »Handbuch KI«, G. Görz, J. Schneeberger und C.-R. Rollinger (Hrsg.), Oldenbourg Verlag, im erscheinen.
  PDF download: http://www-ai.informatik.uni-dortmund.de/LEHRE/VORLESUNGEN/MLRN/SKRIPT/handbuch_ki-ml.pdf
- R. Studer, S. Staab, A. Hotho, A. Mädche: Folien zur Vorlesung „Knowledge Discovery, WS 1999/2000“, AIFB, Univ. Karlsruhe, http://www.aifb.uni-karlsruhe.de/WBS/Lehrveranstaltungen/kdd99_00/