



# Vorlesung Wissensentdeckung

## Closed Sets, Web Mining

Katharina Morik, Claus Weihs

LS 8 Informatik  
Computergestützte Statistik  
Technische Universität Dortmund

28.4.2015



# Gliederung

## 1 Closed Sets

## 2 Web Mining

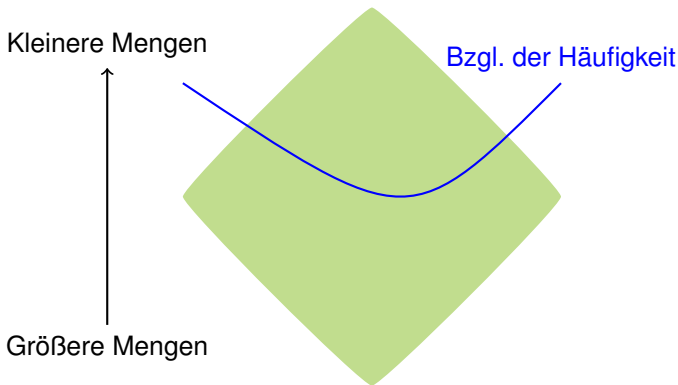
- Finden von häufigen Subgraphen
- Ranking von Web-Seiten nach Autorität



## Zu viele Muster!

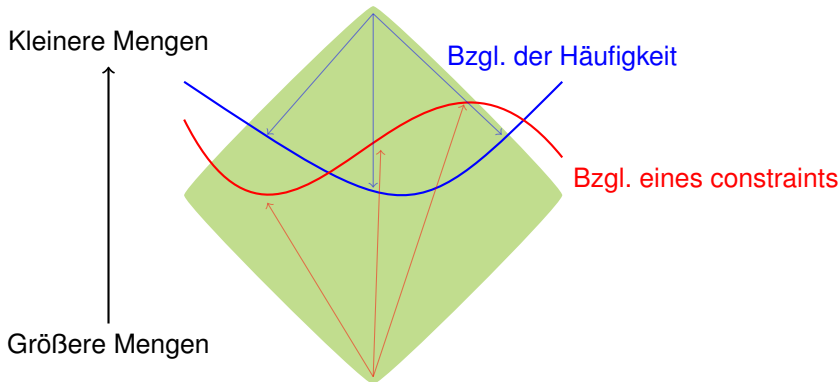
- Es werden sehr viele häufige Mengen gefunden, die redundant sind.
- Wir müssen also aus den Beispielen
  - eine untere Grenze und
  - eine obere Grenze konstruieren.
- Eine Halbordnung bzgl. Teilmengenbeziehung haben wir schon.
- Die Grenzen haben wir auch.
- Gemerkt?

# Untere Grenze



- Wenn eine Menge häufig ist, so auch all ihre Teilmengen. (Anti-Monotonie)
- Beschneiden der Ausgangsmengen für die Kandidatengenerierung gemäß dieser Grenze!

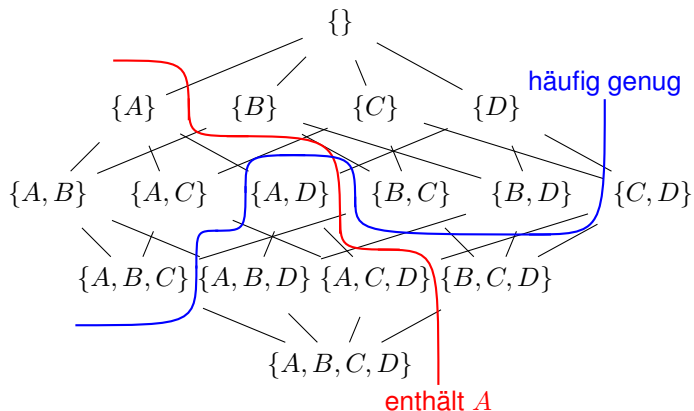
# Obere Grenze



- Monotonie der Seltenheit: Wenn eine Teilmenge selten ist, so auch jede Menge, die sie enthält. Seltenheit ist ein constraint.
- Beschneidung der Kandidatengenerierung nach der Monotonie.

# Beispiel mit Frequency threshold 0.3

A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Dank an Jean-Francois Boulicaut!



## Kondensierte Repräsentationen

- Statt Suche nach allen häufigen Mengen: Suche nach einer kondensierten Repräsentation,
  - die kleiner ist als die ursprüngliche Repräsentation und
  - aus der wir alle häufigen Mengen und ihre Häufigkeit ableiten können, ohne noch mal die Daten selbst anzusehen.
- Kondensierte Repräsentationen für Assoziationsregeln:
  - Closed item sets
  - Free sets
- Operator, der die Menge aller Assoziationsregeln ableitet:
  - Cover operator



## Closed Item Sets

A	B	C	D
1	1	1	1
0	1	1	0
1	0	1	0
1	0	1	0
1	1	1	1
1	1	1	0

- $closure(S)$  ist die maximale Obermenge (gemäß der Teilmengenbeziehung) von  $S$ , die noch genauso häufig wie  $S$  vorkommt.
- $S$  ist ein closed item set, wenn  $closure(S) = S$
- $support(S) = support(closure(S))$  (für alle  $S$ )
- Bei einem Schwellwert von 0.1 sind alle Transaktionen häufig genug.
- Closed sind:  $C, AC, BC, ABC, ABCD$ 
  - keine Obermenge von  $C$  kommt auch 6 mal vor
  - $A$  kommt 5 mal vor, aber auch die Obermenge  $AC$  und keine Obermenge von  $AC$





# Kondensierte Repräsentation und Ableitung

- Closed item sets sind eine kondensierte Repräsentation:
  - Sie sind kompakt.
  - Wenn man die häufigen closed item sets  $C$  berechnet hat, braucht man nicht mehr auf die Daten zuzugreifen und kann doch alle häufigen Mengen berechnen.
- Ableitung:
  - Für jede Menge  $S$  prüfen wir anhand von  $C$ : Ist  $S$  in einem Element  $X$  von  $C$  enthalten?
    - Nein , dann ist  $S$  nicht häufig.
    - Ja , dann ist die Häufigkeit von  $S$  genau die der kleinsten solchen Obermenge  $X$ .
      - Wenn es in mehreren Elementen von  $C$  vorkommt, nimm die maximale Häufigkeit!

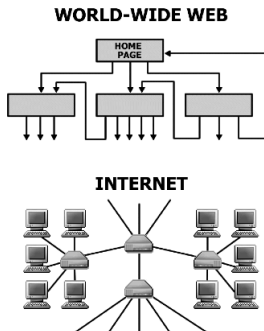


## Was wissen Sie jetzt?

- Sie kennen eine Repräsentation, die weniger Elemente als häufig ausgibt, aus der aber alle häufigen Mengen hergeleitet werden können.
- Es gibt noch viele andere Methoden, um nur interessante Muster auszugeben, aber hier lernen Sie nur eine kennen.

# Das Web und das Internet als Graph

Webseiten sind Knoten, verbunden durch Verweise. Router und andere Rechner sind Knoten, physikalisch verbunden.



Reka Albert, Albert-Laszlo Barabasi: Statistical Mechanics of Complex Networks, arXiv, 2006



## Eigenschaften des World Wide Web (WWW)

Die Struktur des Webs wurde schon früh untersucht  
(<http://arxiv.org/pdf/cond-mat/0106096v1.pdf>)

- *Small Worlds*: Der Pfad zwischen zwei Knoten hat nur wenige Knoten. In einem Ausschnitt des WWW mit 200 Mio. Seiten fanden Broder et al (2000) durchschnittliche Pfadlängen von 16 Knoten.

## Eigenschaften des World Wide Web (WWW)

- Clustering Coefficient  $C$ :** Bei den meisten Knoten  $i$  sind nicht alle ihre direkten Nachfolger  $k_i$  miteinander verbunden, sondern es gibt nur  $E_i$  Kanten zwischen ihnen. Wenn die Nachfolger streng zusammenhängend wären, gäbe es  $k_i(k_i - 1)/2$  Kanten zwischen ihnen.

$$C_i = \frac{2E_i}{k_i(k_i - 1)}$$

Auf das WWW angewandt ignoriert man die Richtung der Kanten. Adamic (1999) fand bei 153.127 Web-Sites  $C=0,1078$ , bei einem Zufallsgraphen der selben Größe nur  $C'=0,00023$ .

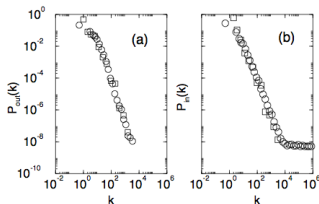
# Eigenschaften des World Wide Web (WWW)

- **Exponentialverteilung:** Die Wahrscheinlichkeit, dass ein Knoten  $k$  Kanten hat, folgt einer Exponentialverteilung:

$$P(X = k) \sim k^{-\gamma}$$

Die bedingte Wahrscheinlichkeit hängt dann nicht von der Größe ab (*scale-free*):  $P(X \geq k | X \geq m) = P(X \geq k)$

Mit hoher Wahrscheinlichkeit gehen nur wenige Kanten ab, kommen nur wenige Kanten an.



Verteilungen der Anzahl von Kanten:

(a) ausgehende (b) eingehende.



# Web Mining

Das WWW hat zu einer Menge interessanter Forschungsaufgaben geführt. Unter anderem gibt es:

- Indexieren von Web-Seiten für die Suche – *machen wir hier nicht*
- Analysieren von Klick-Strömen – web usage mining *kommt später*
- Co-Citation networks – *machen wir hier nicht*
- Finden häufiger Muster in vernetzten Informationsquellen
- Ranking von Web-Seiten nach Autorität



## Finden häufiger Muster in Graphen

- Wir betrachten Web-Sites als Graphen.
  - Jede Web-Seite ist ein Knoten und die Knoten sind verbunden: das Klicken auf einer Seite führt zu einer anderen Seite.
  - Alternativ: die HTML-Struktur wird als Graph aufgefasst.
- Die Beobachtungsmenge beinhaltet viele Graphen und Muster sollen in vielen davon vorkommen.
- Alternativ: Ein Muster soll häufig in einem Graphen vorkommen.





## Apriori-artiges Finden von Mustern

- Wir betrachten Web-Sites als Graphen. Jede Web-Seite ist ein Knoten und die Knoten sind verbunden: das Klicken auf einer Seite führt zu einer anderen Seite.
- Die Beobachtungsmenge beinhaltet viele Graphen und die Muster sollen in vielen davon vorkommen.
- Analog zu Apriori werden häufige Subgraphen erweitert zur Kandidatengenerierung.
- Die Erweiterung ist jetzt etwas komplizierter.
- AGM-Algorithmus: A. Inokuchi, T. Washio, H. Motoda: An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. PKDD Conference 2000.

## Beispiel für Subgraphen



Die beiden linken 5-großen Graphen haben einen 4-großen Graphen gemeinsam.

Die beiden rechten 6-großen Graphen sind die Kandidaten.

Hong Cheng, Xifeng Yan, Jiawei Han: Mining Graph Patterns. In: Charu Agharwal, Haixun Wang (eds): Managing and Mining Graph Data 2010



## Apriori-artiges Finden von Mustern: Notation

- $V(g)$  sind die Knoten eines Graphen  $g$ ;
- $E(g)$  sind die Kanten eines Graphen  $g$ ;
- Annotation  $l$  bildet ein Label auf einen Knoten oder eine Kante ab.

### Subgraph Isomorphie

Für zwei annotierte Graphen  $g, g'$  ist die *Subgraph Isomorphie* die Einbettung  $f : V(g) \rightarrow V(g')$ , so dass

- $\forall v \in V(g) : l(v) = l'(f(v))$
- $\forall (u, v) \in E(g) : (f(u), f(v)) \in E(g')$  und  
 $l(u, v) = l'(f(u), f(v))$

wobei  $l, l'$  die Annotationen von  $g, g'$  sind.



## Häufige Graphen und Anti-Monotonie

Wir können nun die Lernaufgabe definieren als das Finden aller häufiger Graphen.

### Häufiger Graph

- Gegeben eine Menge annotierter Graphen  $D = G_1, \dots, G_n$  und ein Subgraph  $g$ , dann ist  $support(g) = \frac{|D_g|}{|D|}$  und die Menge  $D_g = \{G_i | g \subseteq G_i, G_i \in D\}$ .
- Finde alle Graphen  $g$ , deren  $support(g)$  nicht kleiner ist als  $minsup$ .

### Anti-Monotonie

Ein Subgraph mit  $k$  Knoten bzw. Kanten ist nur dann häufig, wenn alle seine Subgraphen häufig sind.

 $Apriori(D, minsup, S_k)$ 

**Input:** Graphen  $D$ , Schwellwert  $minsup$ ,  $k$  große Subgraphen  $S_k$

**Output:** Die Menge aller  $k + 1$ -großen häufigen Subgraphen  $S_{k+1}$

```

 $S_{k+1} := \{\}$ 
for  $g_i \in S_k$  do
  for  $g_j \in S_k$  do
    for  $g = g_i \odot g_j$  do
      if  $support(g) \geq minsup, g \notin S_{k+1}$ 
      then  $S_{k+1} := S_{k+1} \cup g$ 
    if  $S_{k+1} \neq \{\}$  then
      call  $Apriori(D, minsup, S_{k+1})$ 
return
```



## Closed Subgraphs

Analog zu den Closed Sets gibt es auch bei den Graphen eine Closed Subgraph Darstellung.

### Closed subgraph

A subgraph  $g$  is a *closed subgraph* in a graph set  $D$ , if

- $g$  is frequent in  $D$  and
- there exists no proper supergraph  $g'$  such that  $g \subset g'$  and  $g'$  is frequent in  $D$ .



## Was wissen Sie jetzt?

- Man kann eine Web-site als Graph auffassen, bei dem die Seiten (Knoten) miteinander verbunden sind.
- Auch bei einer Menge von Graphen kann man häufige Muster (Teilgraphen) finden. Sie kennen den Apriori-Algorithmus für Graphen, der ein Muster durch Hinzunahme eines Knotens erweitert.
- Auch bei häufigen Mustern in Graphen gibt es eine aggregierte Darstellung und Sie kennen die Definition.



## Ranking von Web-Seiten

Was sind besonders *wichtige* Seiten?

- Eine Seite, von der besonders viele Links ausgehen, heißt *expansiv*.
- Eine Seite, auf die besonders viele links zeigen, heißt *beliebt*.
- Wie oft würde ein zufälliger Besucher auf eine Seite  $i$  kommen? Zufällige Besuche von einer beliebigen Startseite aus:
  - Mit der Wahrscheinlichkeit  $\alpha$  folgt man einer Kante der aktuellen Seite (Übergangswahrscheinlichkeit).
  - Mit der Wahrscheinlichkeit  $1 - \alpha$  springt man auf eine zufällige Seite, unter der Annahme, dass die Seiten gleich verteilt sind (Sprungwahrscheinlichkeit).

Der Rang einer Seite  $PageRank(i)$  ist der Anteil von  $i$  an den besuchten Knoten.





## Zufalls-Surfermodell: PageRank

Matrix  $M_{ij}$  für Kanten von Knoten  $j$  zu Knoten  $i$ ;  $n(j)$  ist die Anzahl der von  $j$  ausgehenden Kanten;  $N$  Knoten insgesamt.

$$\begin{pmatrix} 1 & \dots & & N \\ 1 & 0 & \dots & M_{1N} \\ \vdots & \dots & M_{ij} = 1/n(j) & \dots \\ N & \dots & \dots & 0 \end{pmatrix}$$

Matrix  $N \times N$  mit den Einträgen  $1/N$  gibt die Gleichverteilung der Knoten an (Sprungwahrscheinlichkeit).

Die Wahrscheinlichkeit, die Seite zu besuchen, ist die Summe von Sprung- und Übergangswahrscheinlichkeit, angegeben in  $N \times N$  Matrix  $M'$ :

$$M' = (1 - \alpha) \left[ \frac{1}{N} \right] + \alpha M \quad (1)$$



# PageRank

Eigenvektoren von  $M'$  geben den Rang der Knoten an.  
Man kann das Gleichungssystem für  $\alpha < 1$  lösen:

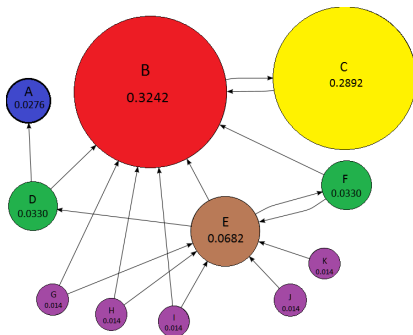
$$\text{Rang}_i = (1 - \alpha) \left[ \frac{1}{N} \right] + \alpha \sum_j M^{-1}ij$$

PageRank ist der rekursive Algorithmus:

$$\text{Rang}_i = \frac{1 - \alpha}{N} + \alpha \sum_{\forall j \in \{(i,j)\}} \frac{\text{Rang}_j}{n(j)} \quad (2)$$

# PageRank Beispiel

Mit  $\alpha = 0,85$  hier ein kleines Beispiel (wikipedia). Die Größe der Kreise entspricht der Wahrscheinlichkeit, mit der ein Surfer auf die Seite kommt. Seite C wird nur von einer einzigen, aber gewichtigeren Seite verlinkt und hat also einen höheren PageRank als Seite E, obwohl E von sechs Seiten verlinkt wird.





## Alternativ: HIT – Hyperlinked-Induced Topic search

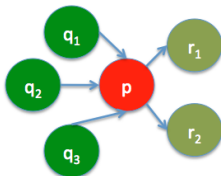
Ähnlich ist eine Anfrage-orientierte Bewertung von Web-Seiten. Statt des Zufalls-Surfers wird eine Suchanfrage gestellt und nur die Menge  $G$  der gelieferten Seiten mit allen Seiten, die auf diese verbunden sind, bewertet. Idee:

- Eine expansive Seite ist ein *hub*  $h$ . Sie soll auf beliebige Seiten zeigen. Die Expansionsbewertung aller Seiten ist ein Vektor  $\vec{h}$ .
- Eine beliebte Seite ist eine *authority*  $a$ . Auf sie soll von beliebigen Seiten verwiesen werden. Die Beliebtheitsbewertung aller Seiten ist ein Vektor  $\vec{a}$ .

Die Matrix der Kanten  $M$  hat  $M_{ij} = 1$ , falls es eine Kante zwischen  $i$  und  $j$  gibt, 0 sonst.

$$\vec{h} = M\vec{a} \quad \vec{a} = M^T\vec{h} \quad (3)$$

# Berechnung von $\vec{h}, \vec{a}$



$k$  Iterationen

- Berechne  $\vec{a}$  für alle Knoten  $p \in G$ :

- $a_p := \sum h_q$ ;
- $norm := \sqrt{\sum_{p \in G} a_p^2}$ ;
- $a_p := a_p / norm$ ;

- Berechne  $\vec{h}$  für alle Knoten  $p \in G$ :

- $h_p := \sum a_r$ ;
- $norm := \sqrt{\sum_{p \in G} h_p^2}$ ;
- $h_p := h_p / norm$ ;

# HIT

- HIT bewertet die Seiten, die eine Suchmaschine geliefert hat, nach Beliebtheit  $h$  und Autorität  $a$ .
- Alle Suchergebnisse werden nach der Bewertung geordnet.
- HIT liefert Seiten mit großem  $h$  und Seiten mit großem  $a$ .
- Probleme:
  - Wenn eine Seite verschiedene Inhalte hat, kann sie auch ausgegeben werden, wenn sie kein guter *hub* für die Anfrage ist: viele  $r$  zu unterschiedlichen Themen!
  - Wenn viele Seiten einer Web-Site auf Seite zeigen, bekommt diese hohe Autorität (topic hijacking), obwohl die  $q_i$  nicht unabhängig voneinander waren.
- Die Summen  $a_p, h_p$  sollten gewichtet werden. PageRank tut das.



## Was wissen Sie jetzt?

- Sie kennen jetzt die Grundlage des Ranking von Web-Seiten und einige Probleme. Schreiben Sie den Kern von PageRank und HIT in Matrix-Notation (Gleichungen 1, 2, 3).
- PageRank schätzt die Wahrscheinlichkeit ab, auf die Seite zu kommen, indem es Kanten folgt und zufällig auf Knoten springt. Dabei verwendet es die Wahrscheinlichkeit  $\alpha$  als Gewicht der Übergangswahrscheinlichkeiten und  $1 - \alpha$  als Gewicht der Sprungwahrscheinlichkeit.
- HIT summiert die Beliebtheit der abgehenden Seiten als Wert der Expansion (*hub*).  
HIT summiert die Expansion der eingehenden Seiten als Wert der Beliebtheit (*authority*).