Complete Mining of Frequent Patterns from Graphs: Mining Graph Data

Seminar aus Maschinellem Lernen

Hongtao Yan

Übersicht

- Motivation
- Wiederholung Warenkorbanalyse (Apriori Algorithmus)
- Darstellung der graphstrukturierten Daten
  - Adjacency matrix
  - Induced Subgraph
  - Vorverarbeitung von labeled Links und self-Looped Knoten im Graph
  - Matrixcode
- Extraktion von häufigen Graphs
  - Join Operation
  - Kanonische Form
- Leistungsfähigkeit beurteilen
- Anwendung
  - Web Browsing Pattern Analyse
  - Chemische Karzinogenese Analyse

Motivation

- Erweitung des Aprioro-Algorithmus zum Itemset Mining auf Graphen
- Darstellung der Graphen als "Matrix"
- Mining häufige "induced" Graphs

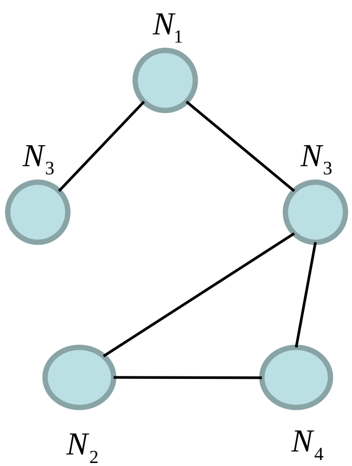
Anwendung

- Web browsing analyse
- Chemische Analyse

Warenkorbanalyse (Apriori Algorithmus)

- Es sei $I = \{i_1, i_2, ... i_n\}$ eine Menge von Objekten (Items)
- D sei eine Menge von Transaktionen, wobei jede Transaktion eine Menge von Objekten ist.
- Eine Assoziationsregel ist eine Implikation der Form X=>Y, wobei X und Y Untermengen von I sind. (Und X und Y keine gemeinsamen Elemente haben)
- Eine Regel X=>Y hat den Konfidenzwert c**,** falls c% der Transaktionen aus D, die X enthalten auch Y enthalten.
- Eine Regel X=>Y hat den Support s, wenn s% der Transaktionen aus D, X vereinigt Y enthalten.

Adjacency matrix



$$\Rightarrow \quad \begin{array}{c c} & \begin{array}{ccccc} N_1 & N_2 & N_3 & N_3 & N_4 \end{array} \\ \begin{array}{c} N_1 \\ N_2 \\ N_3 \\ N_3 \\ N_4 \end{array} & \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$\sup(G) = \frac{nummer\ of\ transaction\ including\ an\ induced\ subgraph\ G}{total\ nummerof\ transactions}$$

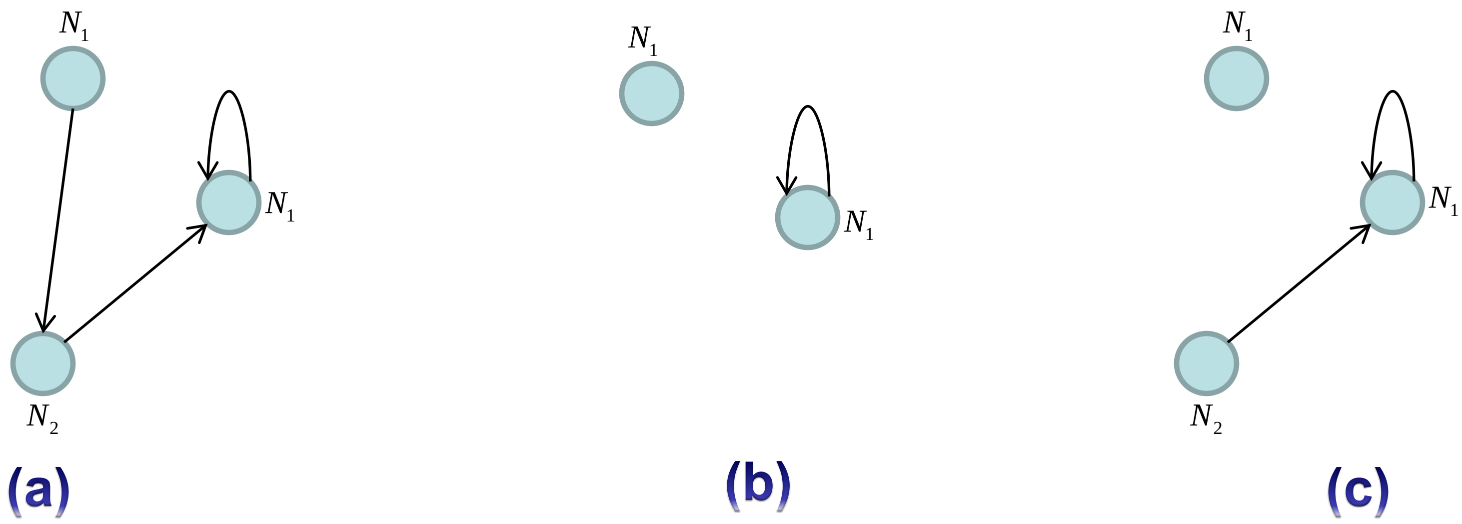
$$\sup(B => H) = \sup(B \cup H) \qquad conf\,(B => H) = \frac{\sup(B \cup H)}{\sup(B)}$$

Induced Subgraph

- Formale Definition von „induced Graph“

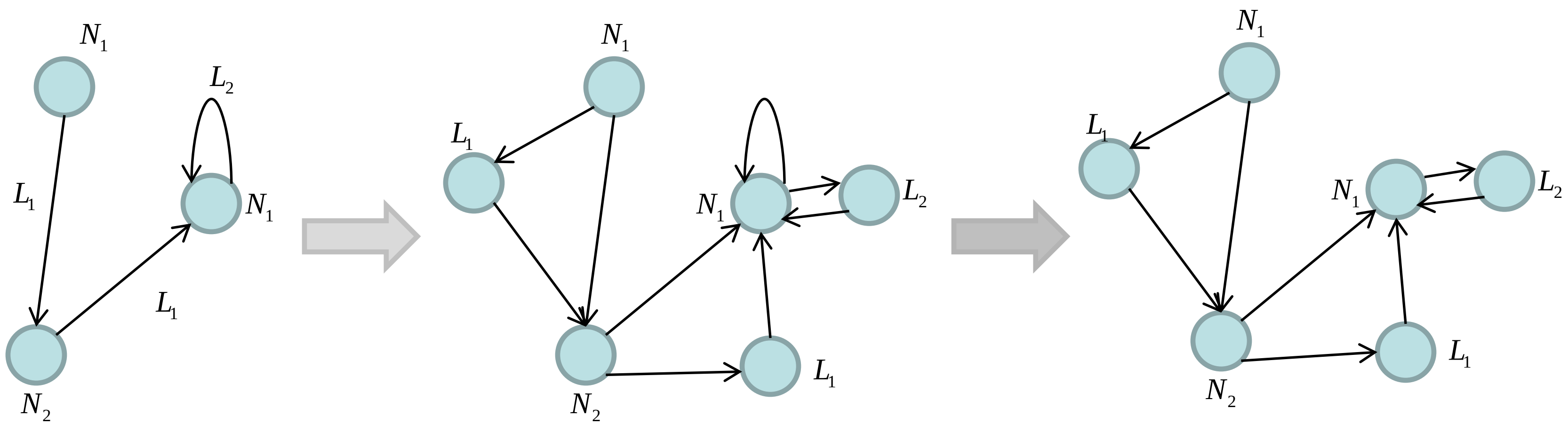
$$V(G') \subseteq V(G), \quad E(G') \subseteq E(G)$$

$$\forall u, v \in V(G')\, \{u,v\} \in E(G) \Leftrightarrow \{u,v\} \in E(G')$$



Labeled Links und self-Loop

- Umwandlung der labeled Links und self-looped Knoten



$$
\begin{array}{c|ccc}
 & N_1 & N_1 & N_2 \\
\hline
N_1 & L_2 & 0 & 0 \\
N_1 & 0 & 0 & L_1 \\
N_2 & L_1 & 0 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccccc}
 & N_1 & N_1 & N_2 & L_1 & L_1 & L_2 \\
\hline
N_1 & 1 & 0 & 0 & 0 & 0 & 1 \\
N_1 & 0 & 0 & 1 & 0 & 1 & 0 \\
N_2 & 1 & 0 & 0 & 1 & 0 & 0 \\
L_1 & 1 & 0 & 0 & 0 & 0 & 0 \\
L_1 & 0 & 0 & 1 & 0 & 0 & 0 \\
L_2 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccccc}
 & N_1' & N_1 & N_2 & L_1 & L_1 & L_2 \\
\hline
N_1' & 0 & 0 & 0 & 0 & 0 & 1 \\
N_1 & 0 & 0 & 1 & 0 & 1 & 0 \\
N_2 & 1 & 0 & 0 & 1 & 0 & 0 \\
L_1 & 1 & 0 & 0 & 0 & 0 & 0 \\
L_1 & 0 & 0 & 1 & 0 & 0 & 0 \\
L_2 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Matrixcode

$$X_k = \begin{pmatrix} 0 & x_{1,2} & x_{1,3} & \cdots & x_{1,k} \\ x_{2,1} & 0 & x_{2,3} & \cdots & x_{2,k} \\ x_{3,1} & x_{3,2} & 0 & \cdots & x_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{k,1} & x_{k,2} & x_{k,3} & \cdots & 0 \end{pmatrix}$$

Ungerichtet Graph:

$$code(X_k) = x_{1,2}x_{1,3}x_{2,3}x_{1,4}\cdots x_{k-2,k}x_{k-1,k}$$

Bsp.:

$$\begin{array}{c|c} & \begin{matrix} N_1 & N_2 & N_3 & N_3 & N_4 \end{matrix} \\ \begin{matrix} N_1 \\ N_2 \\ N_3 \\ N_3 \\ N_4 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$code(X_k) = 0101100101$$

Gerichtet Graph:

$$code(X_k) = c_{1,2}c_{1,3}c_{2,3}c_{1,4}\cdots c_{k-2,k}c_{k-1,k}$$

$$wobei$$

$$c_{i,j} = 2x_{j,i} + x_{i,j}$$

Extraktion häufigen induced Subgraph

- Häufig Subgraph mining ist ähnlich zum Itemset Mining
  - Subgraph G mit k Knoten kann nur frequent sein, wenn alle Subgraphen von G mit k-1 Knoten auch frequent sind
- Zwei Subgraph können nur kombinieren, wenn die folgend drei Bedingung erfüllt sind
- Constraint 1:

$$X_K = \begin{pmatrix} \boxed{X_{K-1}} & x_1 \\ x_2^T & 0 \end{pmatrix} \qquad Y_K = \begin{pmatrix} \boxed{X_{K-1}} & y_1 \\ y_2^T & 0 \end{pmatrix}$$

$$Z_{K+1} = \begin{pmatrix} \boxed{\begin{matrix} X_{K-1} & x_1 \\ x_2^T & 0 \end{matrix}} & \begin{matrix} y_1 \\ z_{k,k+1} \end{matrix} \\ \begin{matrix} y_2^T & z_{K+1,k} \end{matrix} & 0 \end{pmatrix} = \left(\begin{array}{cc|c} \multicolumn{2}{c|}{X_k} & \begin{matrix} y_1 \\ z_{k,k+1} \end{matrix} \\ \hline y_2^T & z_{K+1,k} & 0 \end{array}\right)$$

Constraint

Constraint 2:

$N(X_k, i)$: Label von i-ten Knoten in Matrix X_k.

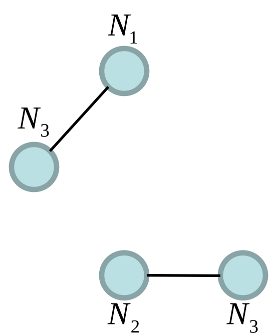
$$N(X_k, i) = N(Y_k, i) = N(Z_{k+1}, i) \; and \;\; N(X_k, i) \leq N(X_k, i+1) \;\; for \; i = 1, \ldots, k-1$$

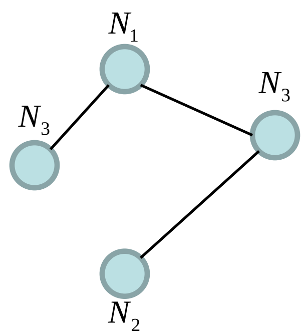
$$N(X_k, k) = N(Z_{k+1}, k), \;\; N(Y_k, k) = N(Z_{k+1}, k+1), and \; N(X_k, k) \leq N(Y_k, k)$$

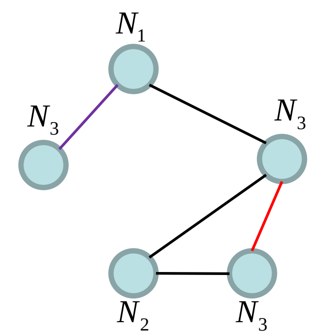
Constraint

- Constraint 3: code(the first matrix) ≤ code(the second matrix)
- Die drei Constraint heißt „join operation“
- Matrix, die durch „join operation“ erzeugt ist, „normal Form“

  z.B.:







$$X_4 = \begin{array}{c c} & \begin{array}{cccc} N_1 & N_2 & N_3 & N_3 \end{array} \\ \begin{array}{c} N_1 \\ N_2 \\ N_3 \\ N_3 \end{array} & \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

$$Y_4 = \begin{array}{c c} & \begin{array}{cccc} N_1 & N_2 & N_3 & N_3 \end{array} \\ \begin{array}{c} N_1 \\ N_2 \\ N_3 \\ N_3 \end{array} & \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

$$Z_5 = \begin{array}{c c} & \begin{array}{ccccc} N_1 & N_2 & N_3 & N_3 & N_3 \end{array} \\ \begin{array}{c} N_1 \\ N_2 \\ N_3 \\ N_3 \\ N_3 \end{array} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & ? \\ 1 & 1 & 0 & ? & 0 \end{pmatrix} \end{array}$$

Normalform

- Direktes Vergrößern der Graphen um jeweils
  - Finde alle Graphen mit *k Knoten und erweitere diese um einen weiteren* Knoten => Kandidaten für *k+1 elementare Graphen*

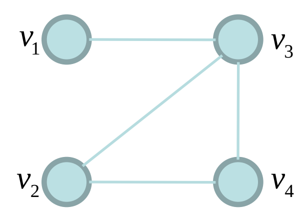
- Generierung durch entfernen dem i-th Knoten => non-normal form Matrix

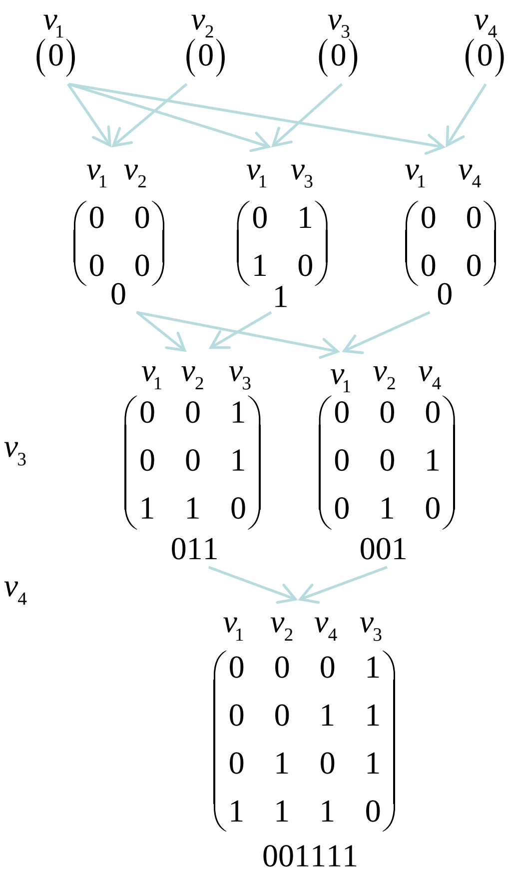
- Ein Methode wandelt non-normal form zu normal Form um

$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 0 & 1 & 0 \\ v_2 & 0 & 0 & 1 & 1 \\ v_3 & 1 & 1 & 0 & 1 \\ v_4 & 0 & 1 & 1 & 0 \end{array}$$

011011

X_4





Kanonisch Form

- Nach alle Kandidaten Graph erzeugt werden, überprüfen ihre Supportwert
- Problem: mehre Normalform bilden identisch Graph ab (z.B.)

$$
X_5 = \begin{array}{c} \\ N_1 \\ N_1 \\ N_1 \\ N_1 \\ N_1 \end{array}
\begin{array}{c} \begin{array}{ccccc} N_1 & N_1 & N_1 & N_1 & N_1 \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}
\qquad
Y_5 = \begin{array}{c} \\ N_1 \\ N_1 \\ N_1 \\ N_1 \\ N_1 \end{array}
\begin{array}{c} \begin{array}{ccccc} N_1 & N_1 & N_1 & N_1 & N_1 \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}
$$

- Falsche Ergebnis von Supportwert
- Lösung: Kanonisch Form

$$\Gamma(G) = \{X_k \mid X_k \textit{is an nomal form matrix and } G \equiv G(X_k)\}$$

$$C_k \; w.r.t\; code(C_k) = \min_{X_k \in \Gamma(G)} code(X_k)$$

Kanonisch Form

- Erzeugung durch Permutation der Zeile und Spalten der normalen Form

$$C_k = W_k^T X_k W_k$$

- Annahme: Permutationsmatrix für jeden häufigen induced k-1 Supgraph sind bekannt
  - Theorem 1: Die erste Matrix einem Kanonischen Form Matrix ist auch ein Kanonischen Form Matrix

$$C_k = \begin{pmatrix} C_{k-1} & C_1 \\ C_2^T & 0 \end{pmatrix}$$

  - Theorem 2:

$$code(C_{k-1}) \leq code(can(X_{k-1}^m)) \qquad (1 \leq m \leq k, N(X_k, m) = N(X_k, k))$$

  - Definition: „pseudo- canonical form“

$$C_k^p = \begin{pmatrix} C_{k-1} & C^p{}_1 \\ C^{pT}{}_2 & 0 \end{pmatrix}$$

$$G(C_k^p) \equiv G(C_k) \equiv G(X_k) \qquad code(C_k) \leq code(C_k^p)$$

Kanonisch Form

- X^m_{k-1} ($1 \le m \le k$) wird zum normale Form umgewandelt. Permutationsmatrix: T^m_{k-1}.
- Zum Kanonische Form umgewandelt. Permutationsmatrix: S^m_{k-1}
- T^m_k und S^m_k wird wie folgende Gleichung definiert.

$$s_{i,j} = \begin{cases} s^m_{i,j} & 0 \le i \le k-1 \, and \; 0 \le j \le k-1 \\ 1 & i = k \; and \; j = k \\ 0 & otherwise \end{cases} \qquad t_{i,j} = \begin{cases} t^m_{i,j} & i < m \; and \; j \ne k \\ t^m_{i-1,j} & i > m \; and \; j \ne k \\ 1 & i = m \; and \; j \ne k \\ 0 & otherwise \end{cases}$$

- nach Theorem 3: ein „pseudo-canonical form“

$$C^p_k \; w.r.t \; code(C^p_k) = \min_m code((T^m_k S^m_k)^T X_k (T^m_k S^m_k))$$

- Und ein Kanonische Form wie folgend

$$C_k \; w.r.t \; code(C_k) = \min_{U_k \in \Lambda(C^p_c)} code((U_k{}^T C^p_k U_k)$$

Beispiel

$$
X_5 = \begin{array}{c} \\ N_1 \\ N_1 \\ N_1 \\ N_1 \\ N_1 \end{array}
\begin{array}{c}
\begin{array}{ccccc} N_1 & N_1 & N_1 & N_1 & N_1 \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}
\end{array}
\qquad
Y_5 = \begin{array}{c} \\ N_1 \\ N_1 \\ N_1 \\ N_1 \\ N_1 \end{array}
\begin{array}{c}
\begin{array}{ccccc} N_1 & N_1 & N_1 & N_1 & N_1 \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}
\end{array}
$$

$$
X_4^m s \ for \ m = 1,2,3,4,5 \qquad
X_4^1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}
X_4^2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}
X_4^3 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}
X_4^4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}
X_4^5 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}
$$

$$
T_4^1 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} and \ S_4^1 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}
\Rightarrow T_5^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},
S_5^1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}
$$

- Für m=1,2,3,4 und 5 $code((T_5^m S_5^m)^T X_5 (T_5^m S_5^m))$ =0011010011, 0011110010,0 011110010, 0000111011, 0000111101
- m=4 ist Minimum C_5^p ist gefunden
- Y_5 ist kanonische Form

Algorithmus

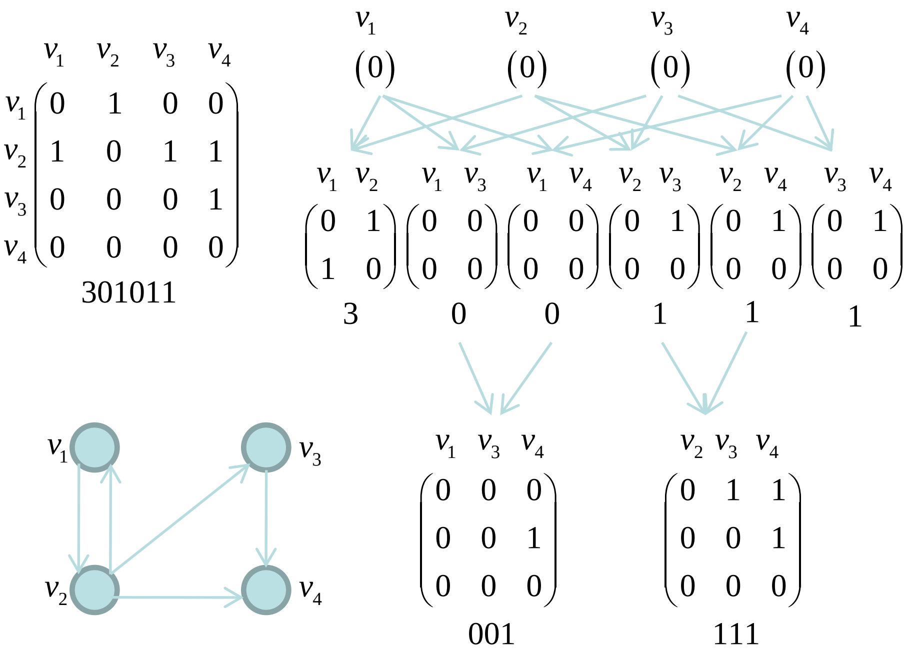
1) **forall** X_k in a set of candidate frequent induced subgraphs
2) $X'_k = X_k,\ m \leq 1$
3) **while** $m \leq k$ **do begin**
4) **if**$(N(X_k, m) = N(X_k, k))$ **then do begin**
5) **if**$(code(X'_k) > code((T_k^m S_k^m)^T X_k (T_k^m S_k^m))$ **then do begin**
6) $X'_k = (T_k^m S_k^m)^T X_k (T_k^m S_k^m));$
7) **if**(the transformation matrix of X'_k to the canonicalform is known) **then do begin**
8) $X'_K = S_K^T X'_k S_K;$
//where *Sk* is the matrix to transform X'_k in r.h.s. to its canonicalform./
9) **break**;
10) **end**
11) **end**
12) **end**
13) m=m+1
14) **end**
15) **if**(the canonicalform of X_K has not been derived in the step 8)
16) X'_k =permutation(X'_k);
17) **end**
18) Canonicalform of X_k is X'_k;
19) **end**

Generierung häufigen Subgraphs

- Annahme: 2 sitz Subgraph, die code 3 ist, ist nicht häufig Subgraph

- Kombination wenn first Matrix gleich ist

- Kombination wenn Knotenlabel in erst Matrix gleich sind.



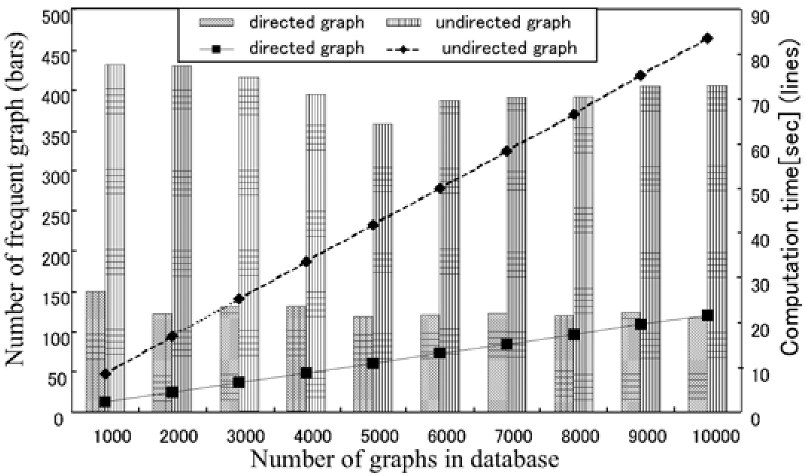
Leistungsbewertung

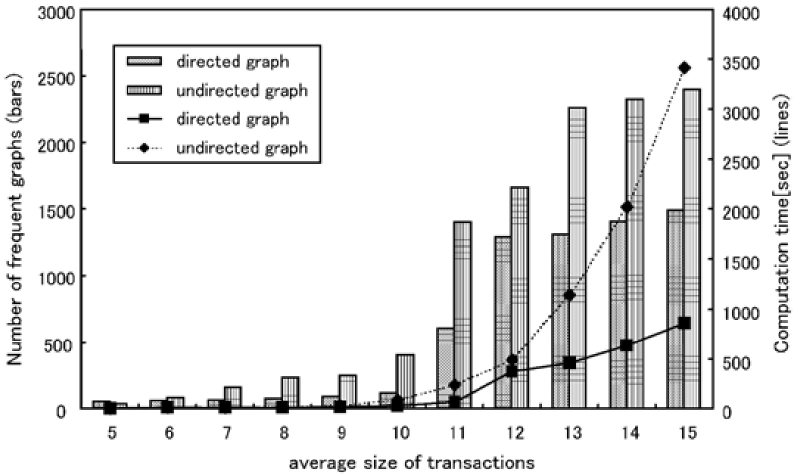
Pentium CPU 400MhHz, Ram 128MB

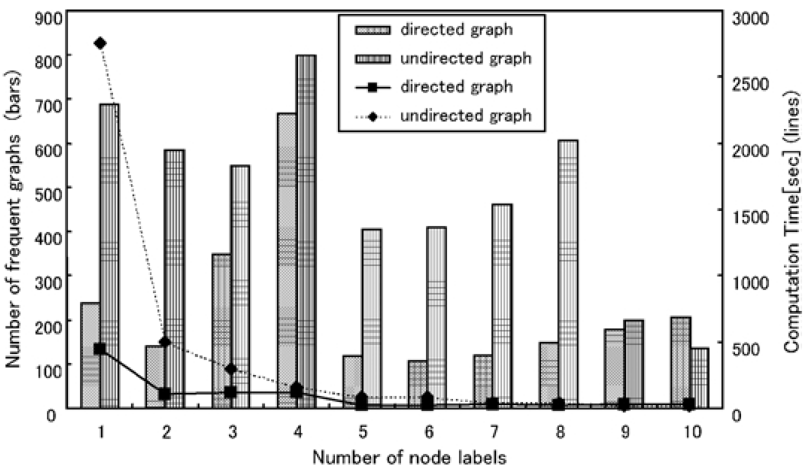
Parameter	Definition	Default value
D	Number of transaction	10,000
|T|	Average transaction size	10
L	Number of basic patterns	10
|I|	Average basic patterns size	4
N	Number of node labels	5
p	Link existence probability	50%
minsup	Minimum support	10%

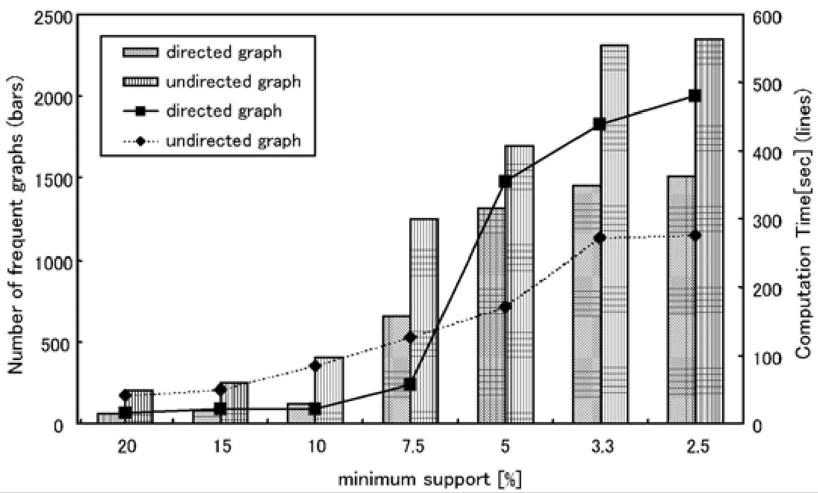
Leistungsbewertung

TECHNISCHE UNIVERSITÄT DARMSTADT



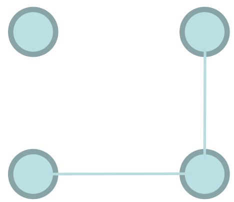






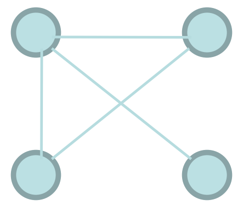
Leistungsbewertung

- Ein komplett Graph ist ein Graph, in dem ein Link für jeder Knoten existiert.



000011
000101
000110
010100
100010

(a)



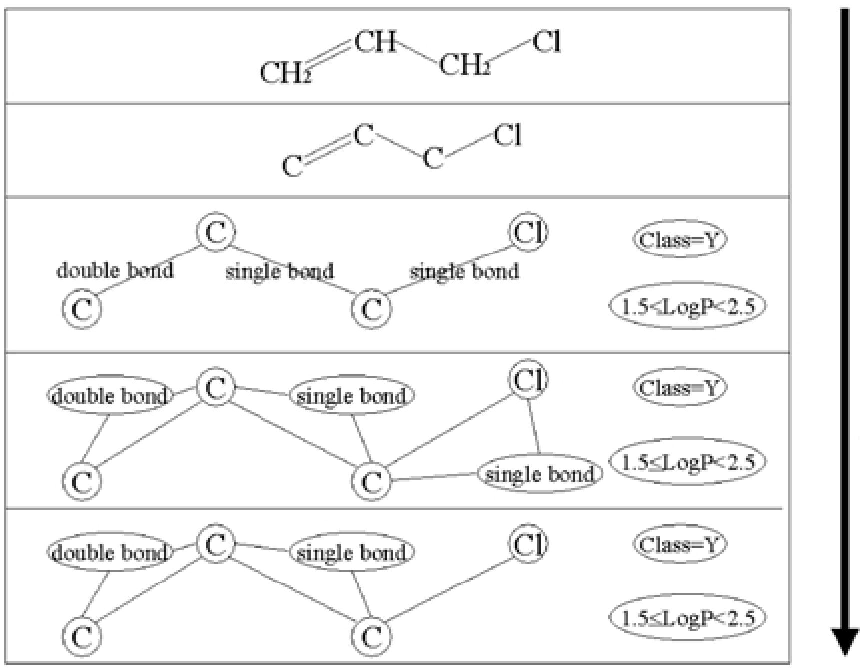
111001
110110
010111
001111

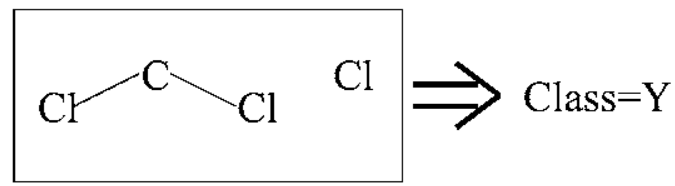
(b)

Komplementär

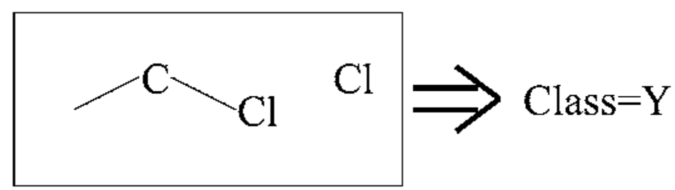
Anwendung

- Web Browsing Analyse
- Chemische Karzinogenes Analyse





Ex1. support=31.7%, confidence=86.7%



Ex2. support=36.6%, confidence=83.3%

Danke für Ihre Aufmerksamkeit