Seminar aus maschinellem Lernen: A General Framework for Mining Frequent Subgraphs from Labeled Graphs

Timo Schneider

17. Dezember 2008

Übersicht

- Graph Mining Allgemein
- Probleme und Lösungsansätze
- Das General Framework
- Experimentelle Ergebnisse
- Zusammenfassung

Graph Mining Allgemein

- Gegeben: Ein beschrifteter Graph $G = (V, E, L_V, L_E)$
- Gesucht: Frequente Muster
- Bisher: Unterschiedliche Algorithmen für unterschiedliche Muster

Graph Mining Allgemein

- B-AGM kann mit unterschiedlichen Suchkriterien parametrisiert werden
- Hat somit mehrere Einsatzgebiete
- Getestet auf chemischen Datensätzen und Webseiten-Logdateien

Probleme und Lösungsansätze

- Wie neue Kandidaten erzeugen?
  - Zwei Graphen mit gleichem Kern zusammenfügen
  - Nur einen Subgraph mit einem Knoten erweitern
- Wie die Vorkommen überprüfen?
  - Bereits erkannte Teilgraphen merken...
  - ... Obergraphen davon können nur an diesen Stellen liegen

Probleme und Lösungsansätze

- AGM kennen wir schon
- Kandidatenerzeugung per Join-Operation
- Dann abzählen: Recht speicherintensiv

Das General Framework

Das Framework soll durch Austauschen einiger Funktionen andere Strukturen erkennen.

- AGM: Teilgraphen
- Verbundene Teilgraphen
- geordnete Teilbäume
- Pfade

Das General Framework

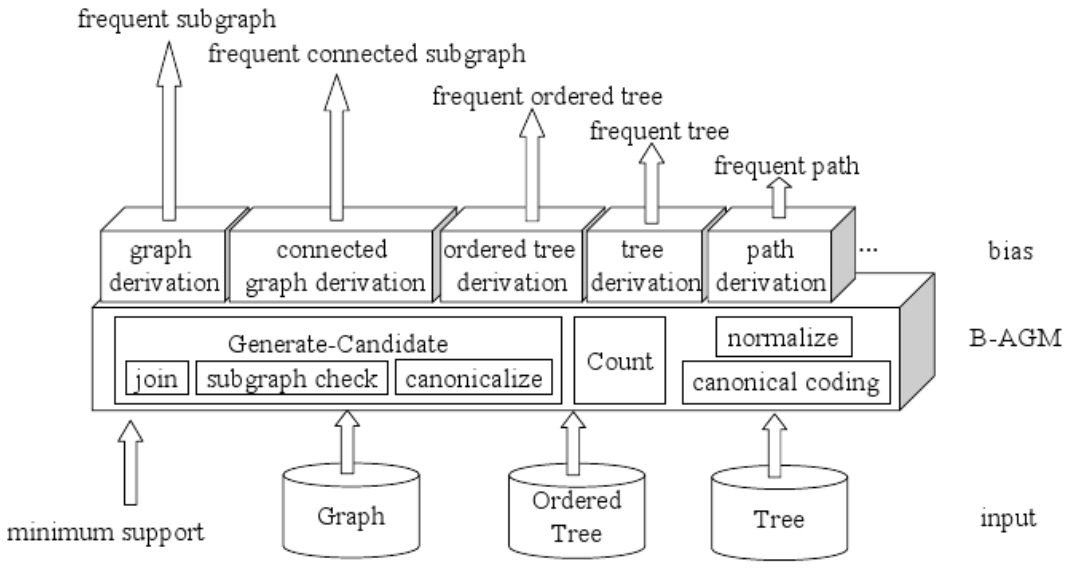


Figure 11. B-AGM Framework.

Das General Framework

Damit zwei Graphen X_k und Y_k zusammengefügt werden, müssen folgende Bedingungen eingehalten werden:

- 1.) X_k und Y_k besitzen die gleiche Erzeugermatrix X_{k-1}
- 2.) X_k ist die kanonische Form von $G(X_k)$

Das General Framework

Bei AGM kam als dritte Bedingung hinzu:

- 3.) $CODE(X_k) >= CODE(Y_k)$
- Damit werden ganz allgemein Teilgraphen erzeugt
- CODE ist hier: $num(lb(v_1))...num(lb(v_k))code(X_k)$

Das General Framework

Beispiel einer Join-Operation bei AGM

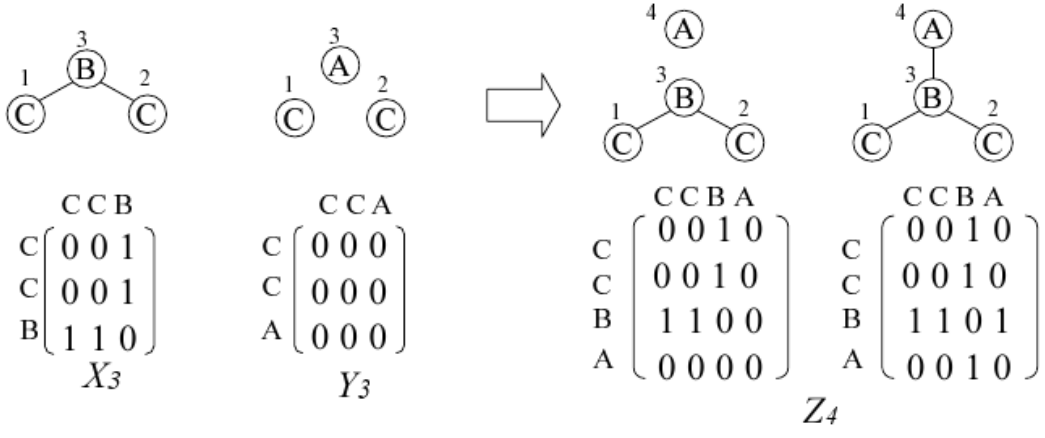


Figure 6. Example of Join Operation.

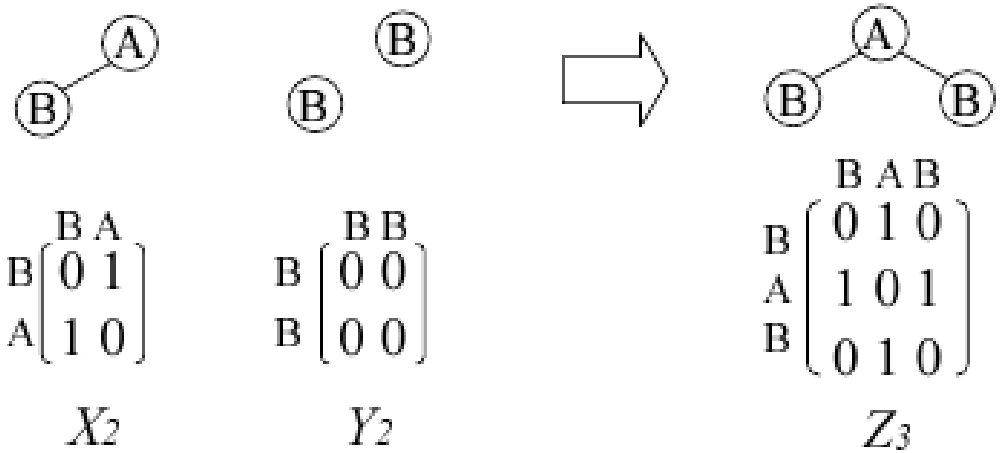
Das General Framework

Erzeugung verbundener Teilgraphen

- 3.) $G(X_k)$ ist ein verbundener Graph
- 4.) $CODE(X_k) >= CODE(Y_k)$ oder $G(Y_k)$ ist nicht verbunden

Das General Framework

Join-Operation für verbundene Teilgraphen



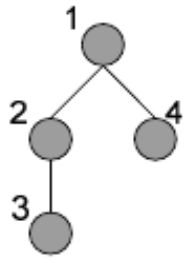
Das General Framework

Erzeugung von geordneten Teilbäumen

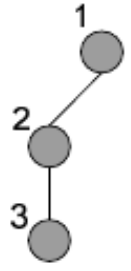
- 3.) $code(X_k) <= code(Y_k)$ oder $G(Y_k)$ ist nicht verbunden
- 4.) $G(X_k)$ ist verbunden

Das General Framework

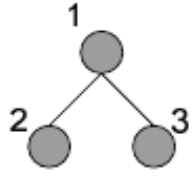
Join-Operation für geordnete Teilbäume



G_1



G_2



G_3

$$Z_4 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad X_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad Y_3 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$code(Z_4) = 101100 \qquad code(X_3) = 101 \qquad code(Y_3) = 110$

Das General Framework

Pfade erzeugen:

- 3.) $G(X_k)$ ist verbunden
- 4.) $CODE(X_k) <= CODE(Y_k)$ oder $G(Y_k)$ ist nicht verbunden
- 5.) Bei Join verbiete neue Verbindungen

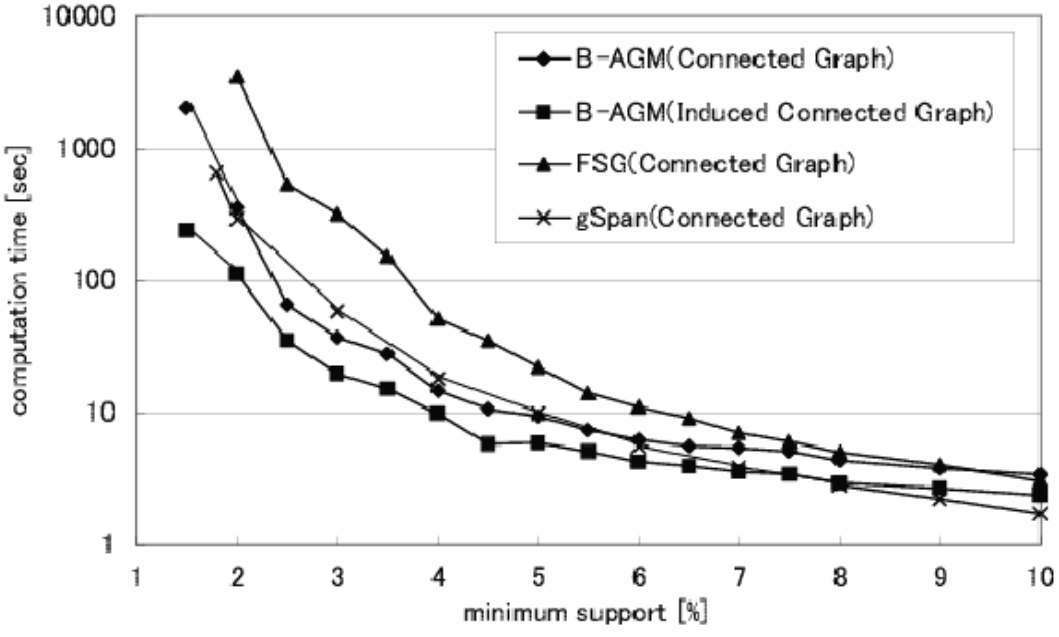
Das General Framework

- Im Prinzip das gleiche wie Bäume erzeugen
- Aber: Hier werden die Label der Knoten beachtet
- Und wir lassen keine Verbindungen zu,
  um Verzweigungen zu unterbinden

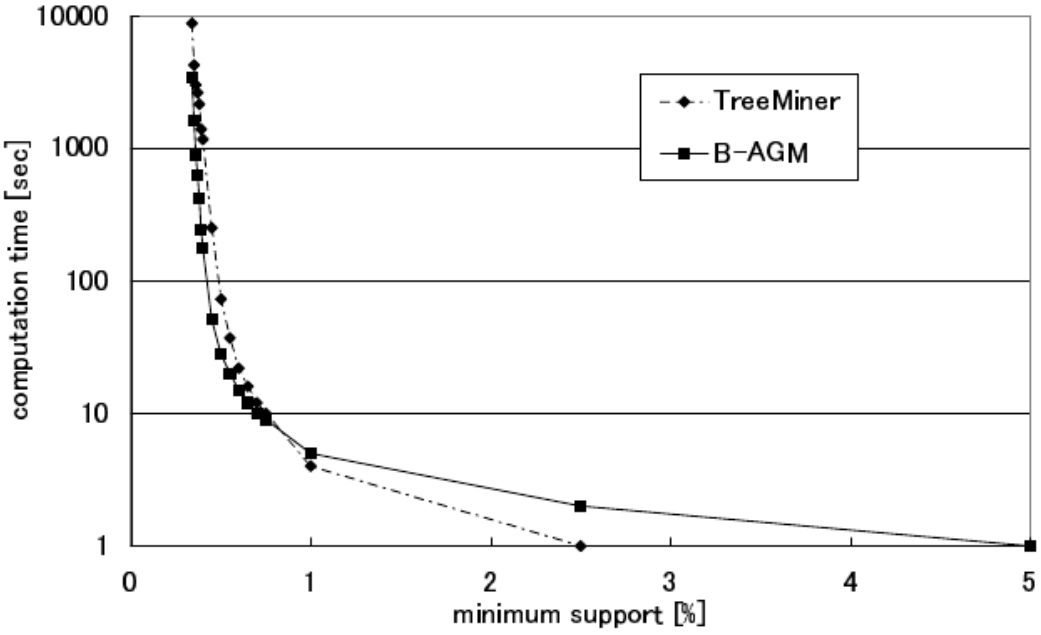
Das General Framework

- Somit lassen sich mit einem Algorithmus unterschiedliche Strukturen minen
- Die Autoren beweisen im Artikel auch die Vollständigkeit in Hinsicht auf die jeweilige Struktur

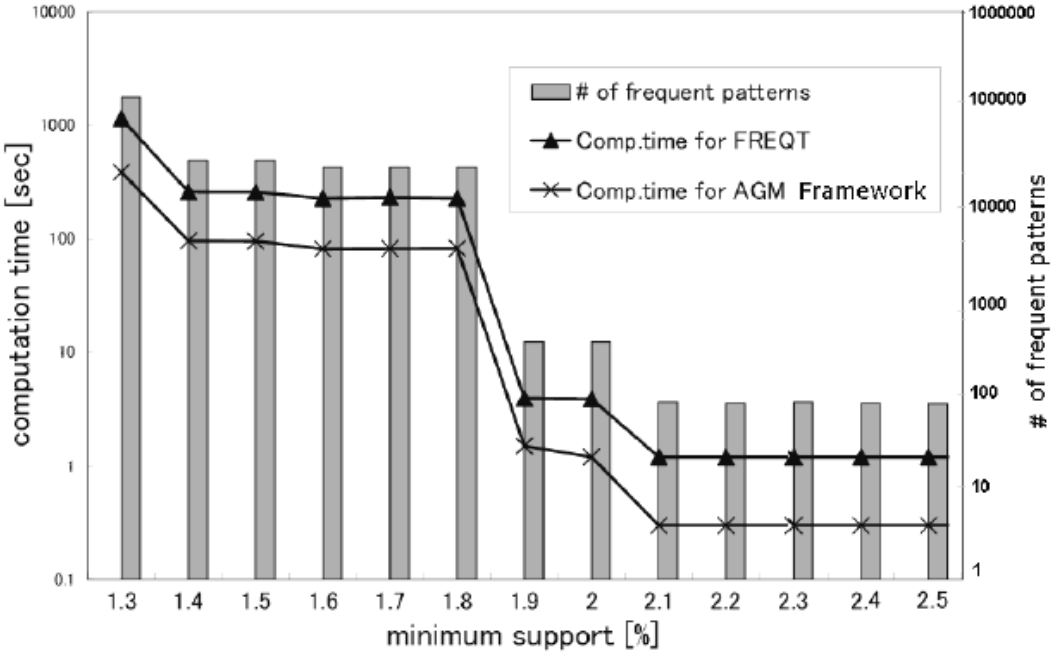
Experimentelle Ergebnisse



Experimentelle Ergebnisse



Experimentelle Ergebnisse



Zusammenfassung

- Trotz höherer Abstraktionsebene schlägt sich B-AGM nicht schlecht
- Der Speicherbedarf liegt in $O(n^2)$ (bei anderen Algorithmen meist $O(n)$)
- Wir speichern ja eine Matrix, wo andere nur eine Knotenmenge speichern

Zusammenfassung

- B-AGM kann unterschiedliche Strukturen ableiten:
  - (verbundene) Teilgraphen
  - geordnete Teilbäume
  - Pfade
- Die Autoren schlagen ebenfalls eine Erweiterung vor, um ungeordnete Teilbäume zu finden
- Wurde jedoch im Dokument nicht näher untersucht


Ingrid Fischer and Thorsten Meinl.
Graph based molecular data mining - an overview.


Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda.
An apriori-based algorithm for mining frequent substructures from graph data.


Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda.
Complete mining of frequent patterns from graphs: Mining graph data.


Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda.
A general framework for mining frequent subgraphs from labeled graphs.

Fragen?

Vielen Dank für Ihre Aufmerksamkeit.