Algorithmen in der Dokumenten-Clusterung

Assoziationsregeln und Hypergraph-Partitionierung

Sascha Grau

28. November 2006

Inhalt

Zielstellung

Gruppierung von Text-Dokumenten zu inhaltlich verwandten Domänen, unter ausschließlicher Nutzung der durch Wortmenge und -vorkommen gegebenen Informationen.

Motivation

- Unabhängigkeit von Metainformationen (Links, ...)
- Entdeckung verborgener Zusammenhänge
- Verbessertes Erschließen großer Textsammlungen (Browsing)
- Einbeziehung in Suchverfahren
  - Einschränkung der Treffermenge von Polynymen
  - Erweiterung der Treffermenge auf Gesamtkategorie

Zielstellung

Gruppierung von Text-Dokumenten zu inhaltlich verwandten Domänen, unter ausschließlicher Nutzung der durch Wortmenge und -vorkommen gegebenen Informationen.

Motivation

- Unabhängigkeit von Metainformationen (Links, ...)
- Entdeckung verborgener Zusammenhänge
- Verbessertes Erschließen großer Textsammlungen (Browsing)
- Einbeziehung in Suchverfahren
  - Einschränkung der Treffermenge von Polynymen
  - Erweiterung der Treffermenge auf Gesamtkategorie

Vorverarbeitung

- Aufbereitung
  - Spracherkennung
  - Stemming
  - Feststellung der Worthäufigkeiten
  - Termgewichtung
  - Normalisierung der Dokumentenlänge
- Ausschluss unbenötigter Information
  - Stopwort-Entfernung
  - Frequenzbasierte Wortfilterung

Der vorverarbeitete Datensatz

- Wort-Dokument-Matrix, mit skalierten Auftretenshäufigkeiten
- enormer Umfang (abhängig von Dokumentenlängen & Zahl der Themengebiete)
- dünn besetzt

Weiterverarbeitung

- prinzipiell sind nun beliebige Clusterungsalgorithmen einsetzbar
- wünschenswert sind Verfahren, welche auf die speziellen Eigenschaften des Datensatzes abgestimmt sind (PDDP, ARHP)

Der vorverarbeitete Datensatz

- Wort-Dokument-Matrix, mit skalierten Auftretenshäufigkeiten
- enormer Umfang (abhängig von Dokumentenlängen & Zahl der Themengebiete)
- dünn besetzt

Weiterverarbeitung

- prinzipiell sind nun beliebige Clusterungsalgorithmen einsetzbar
- wünschenswert sind Verfahren, welche auf die speziellen Eigenschaften des Datensatzes abgestimmt sind (PDDP, ARHP)

Assoziationsregeln

Einführung

- Instrument des Data Mining
- Erschließung simpler Objektrelationen, durch Betrachtung einer Menge von Transaktionen über diesen Objekten
- Im Folgenden Konzentration auf *boolsche Assoziationsregeln*, mit Aussagen zum gemeinsamen Auftreten von Objekten
- Beispiel: Warenkorbanalyse

Items & Transaktionen

Sei $\mathcal{I}$ die Menge aller Items. Eine Menge $t \subseteq \mathcal{I}$ heißt Transaktion. Die Menge aller Transaktionen heißt $\mathcal{T}$.

Support einer Itemmenge

Jede Menge $X \subseteq \mathcal{I}$ heißt *Itemmenge*.
Der *Support* $\sigma(X) = |\{t \in T | X \subseteq t\}|$ einer Itemmenge, ist die Anzahl der Transaktionen aus $\mathcal{T}$ in denen X als Teilmenge auftritt.

Items & Transaktionen

Sei $\mathcal{I}$ die Menge aller Items. Eine Menge $t \subseteq \mathcal{I}$ heißt Transaktion. Die Menge aller Transaktionen heißt $\mathcal{T}$.

Support einer Itemmenge

Jede Menge $X \subseteq \mathcal{I}$ heißt *Itemmenge*.
Der *Support* $\sigma(X) = |\{t \in T | X \subseteq t\}|$ einer Itemmenge, ist die Anzahl der Transaktionen aus $\mathcal{T}$ in denen X als Teilmenge auftritt.

Assoziationsregeln

Der Ausdruck $A \xrightarrow{\sigma,\alpha} B$ mit $A, B \subseteq \mathcal{I}; A, B \neq \emptyset$, heißt *Assoziationsregel*.

Support & Konfidenz einer Assoziationsregel

Der Support σ der Regel ist $\frac{\sigma(A \cup B)}{|T|}$.
Der gemeinsame Transaktionsanteil.

Die Konfidenz α entspricht $\frac{\sigma(A \cup B)}{\sigma(A)}$.
Die Bedingte Auftretenswahrscheinlichkeit $P(B \subseteq t | A \subseteq t)$.

Assoziationsregeln

Der Ausdruck $A \xrightarrow{\sigma,\alpha} B$ mit $A, B \subseteq \mathcal{I}; A, B \neq \emptyset$, heißt *Assoziationsregel*.

Support & Konfidenz einer Assoziationsregel

Der Support σ der Regel ist $\frac{\sigma(A \cup B)}{|T|}$.
Der gemeinsame Transaktionsanteil.

Die Konfidenz α entspricht $\frac{\sigma(A \cup B)}{\sigma(A)}$.
Die Bedingte Auftretenswahrscheinlichkeit $P(B \subseteq t | A \subseteq t)$.

Begrenzung der Regelanzahl

- Die Menge aller Items $\mathcal{I}$ besitzt $2^{|\mathcal{I}|} - |\mathcal{I}| - 1$ Teilmengen über deren Bipartitionen sich nicht-triviale Assoziationsregeln bilden lassen.
- Um diese enorme Menge einzuschränken, werden nur *häufige Itemmengen* betrachtet.

Häufige Itemmengen

Eine Itemmenge X gilt als *häufig* (*engl.*: frequent item set), falls ihr Support $\sigma(X)$ oberhalb eines benutzerdefinierten minimalen Support-Schwellwertes liegt.

Begrenzung der Regelanzahl

- Die Menge aller Items $\mathcal{I}$ besitzt $2^{|\mathcal{I}|} - |\mathcal{I}| - 1$ Teilmengen über deren Bipartitionen sich nicht-triviale Assoziationsregeln bilden lassen.
- Um diese enorme Menge einzuschränken, werden nur *häufige Itemmengen* betrachtet.

Häufige Itemmengen

Eine Itemmenge X gilt als *häufig* (*engl.*: frequent item set), falls ihr Support $\sigma(X)$ oberhalb eines benutzerdefinierten minimalen Support-Schwellwertes liegt.

Bildung von Assoziationsregeln

Input: $\mathcal{I}$, $\mathcal{T}$ und untere Häufigkeitsschranke ϵ
Output: Assoziationsregeln über den häufigen Itemmengen

$M :=$ häufige_Itemmengen($\mathcal{I}$, $\mathcal{T}$, ϵ);
foreach $B \in M$ **do**
  foreach $A \subset B$ **do**
    $\sigma(A \rightarrow B - A) := \sigma(B)/|\mathcal{T}|$;
    $\alpha(A \rightarrow B - A) := \sigma(B)/\sigma(A)$;
  end
end

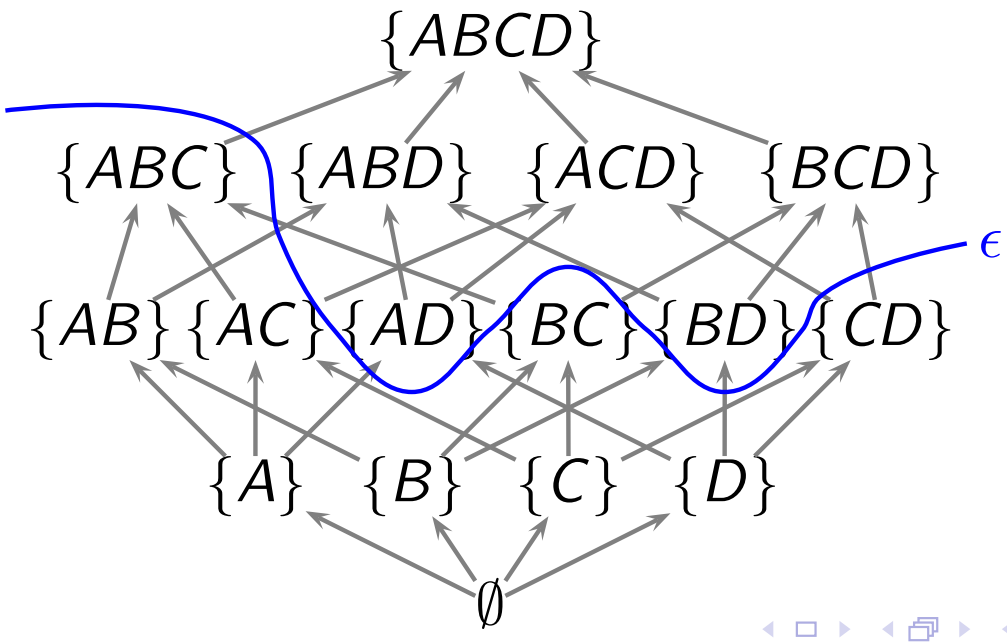
Laufzeitfaktoren

- Berechnung der häufigen Itemmengen dominiert Laufzeit
  - üblicherweise sehr umfangreiche Datenbasis
  - IO-Zugriffe auf externe Medien
  - exponentielle Zahl von Kandidatenmengen
- Mit häufigen Itemmengen steht auch deren Support fest, so dass sich die Werte der restlichen Berechnungen ergeben

Der Verband der Kandidatenmengen

Die Kandidatenmengen $\mathcal{P}(\mathcal{I})$ bilden mit der Teilmengenrelation $\subseteq$ einen Verband. In diesem gelten folgende Beobachtungen:

- Der Support fällt monoton mit steigender Kardinalität.
- Alle Teilmengen einer häufigen Itemmenge sind auch häufig.
- Es existiert eine Menge maximal häufiger Itemmengen Bd^+.



Geeignete Algorithmen, berechnen also nicht die Support-Werte aller möglichen Itemmengen, sondern erzeugen Kandidaten mit zunehmender Kardinalität durch Erweiterung bereits gefundener häufiger Itemmengen.

Der Apriori-Algorithmus

Input: $\mathcal{I}$, $\mathcal{T}$ und untere Häufigkeitsschranke ϵ
Output: Menge F der häufigen Itemmengen

$F := \emptyset$;
$C_1 := \mathcal{I}$;
$i := 1$;
while $C_i \neq \emptyset$ **do**
  foreach $c \in C_i$ **do**
    Scanne $\mathcal{T}$, berechne $\sigma(c)$;
    if $\sigma(c) < \epsilon$ **then** $C_i := C_i \backslash c$;
  end
  $F := F \cup C_i$;
  $C_{i+1} := \{\{a \cup b\} \,|a, b \in C_i \wedge |a \cap b| = i - 1\}$;
  $i := i + 1$;
end

Anzahl der Datenbankzugriffe

- Zugriffe auf $\mathcal{T}$ teuer (externe Datenbank)
- Apriori vollführt kompletten Scan pro getesteter Kardinalitätsstufe

Reduzierung der Datenbankzugriffe

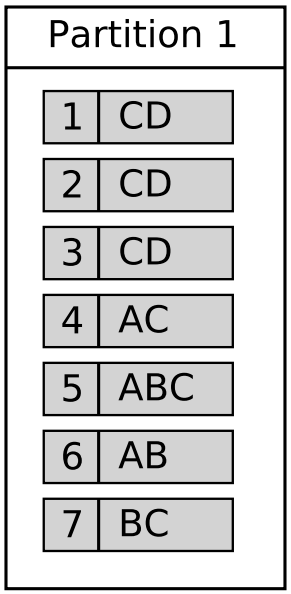
- Einführung von Transaktions-IDs
- Datenbank-Scan ordnet jede TID den in der Transaktion vorkommenden einelementigen Teilmengen zu
- Support der Mengen höherer Kardinalität folgt simpel durch Schnittbildung der TID-Mengen

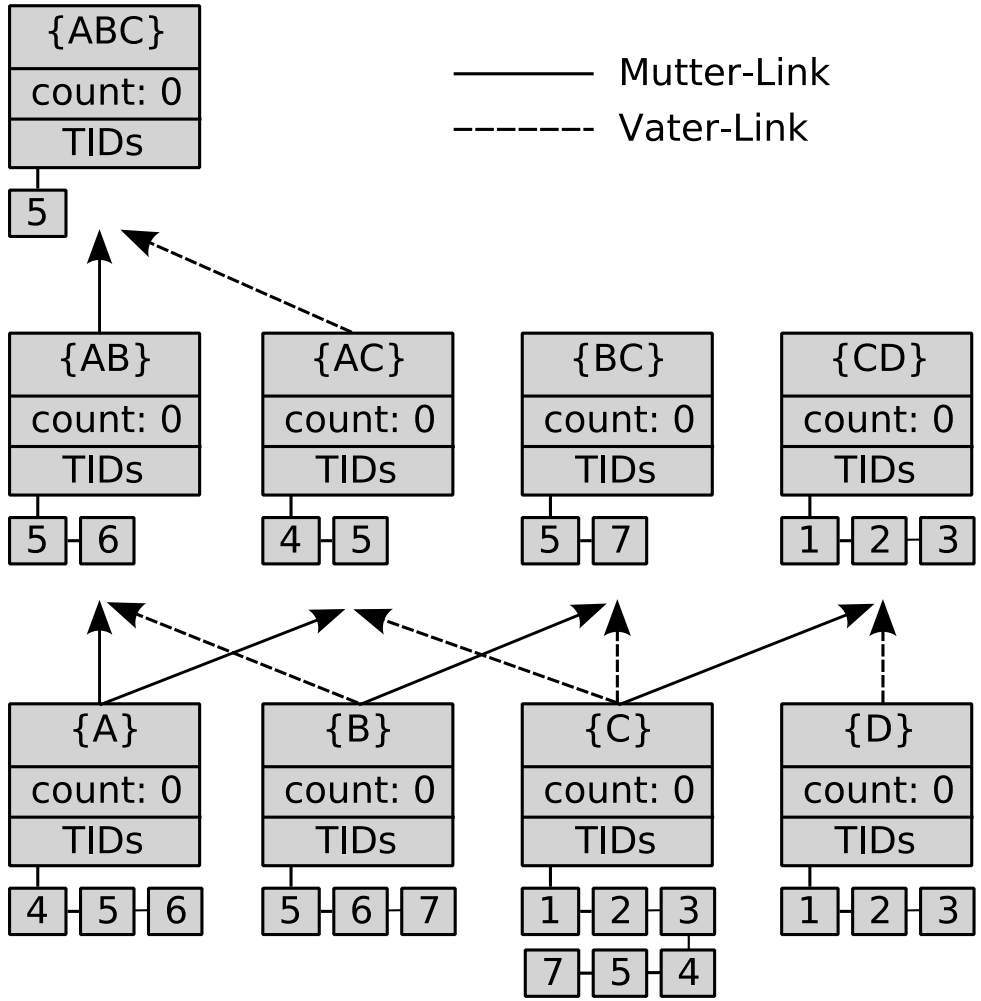
Anzahl der Datenbankzugriffe

- Zugriffe auf $\mathcal{T}$ teuer (externe Datenbank)
- Apriori vollführt kompletten Scan pro getesteter Kardinalitätsstufe

Reduzierung der Datenbankzugriffe

- Einführung von Transaktions-IDs
- Datenbank-Scan ordnet jede TID den in der Transaktion vorkommenden einelementigen Teilmengen zu
- Support der Mengen höherer Kardinalität folgt simpel durch Schnittbildung der TID-Mengen


Partition 1
1 CD
2 CD
3 CD
4 AC
5 ABC
6 AB
7 BC


{ABC}
count: 0
TIDs
5
Mutter-Link
Vater-Link
{AB}
count: 0
TIDs
5 6
{AC}
count: 0
TIDs
4 5
{BC}
count: 0
TIDs
5 7
{CD}
count: 0
TIDs
1 2 3
{A}
count: 0
TIDs
4 5 6
{B}
count: 0
TIDs
5 6 7
{C}
count: 0
TIDs
1 2 3
7 5 4
{D}
count: 0
TIDs
1 2 3

Der ARMOR-Algorithmus

- Verarbeitung von $\mathcal{T}$ in Partitionen um Speicher zu sparen
- 2 Datenbankdurchläufe (jeweils mit TID-Schnittmengenbildung):
  1. Schätzung einer Obermenge der enthaltenen häufigen Itemmengen und Bd^-
     - Beginn mit allen einelementigen Itemmengen
     - Hinzufügen neues Kandidaten, sobald eine seiner Teilmengen als häufig eingeschätzt wird
     - Min-/Max-Abschätzung des Supports des eingefügten Kandidaten über Teilmengen-Support
  2. Support-Bestimmung für Kandidatenmengen und Ausschluss nicht-häufiger Kandidaten

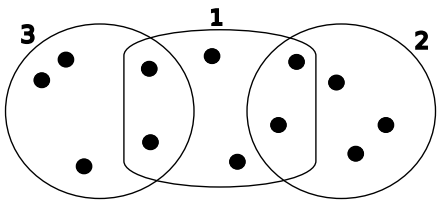
Hypergraphen

Definition

Ein (ungerichteter) *Hypergraph* ist ein Tupel $G = (V, E)$ bestehend aus einer Knotenmenge V und einer Menge $E \subset \mathcal{P}(V)$ von Hyperkanten, wobei $\mathcal{P}(V)$ die Potenzmenge von V bezeichnet.

Gewichtsfunktionen

Die Funktion $w : E \rightarrow \mathbb{R}$ heißt Kantengewichtsfunktion, analog lässt sich eine Knotengewichtsfuntion c definieren.



Partition

Eine *Partition* P eines Hypergraphen $G = (V, E)$ ist eine nicht-leere Menge von Teilmengen $P_1, \ldots, P_r$ aus V für die gilt: $\forall\ P_i, P_j \in P : P_i \cap P_j = \emptyset; \bigcup_{i=1}^{r} P_i = V$.
Die Menge der Hyperkanten welche Überdeckungen mit mehr als einer dieser Teilmengen besitzen, heißt *Schnitt*.

Schnittkosten

Die Kosten $w(S)$ eines Schnitts $S \subseteq E$ entsprechen der Summe $\sum_{e \in S} w(e)$ der Gewichte der Kanten in S.

BALANCED MIN MULTIWAY PARTITION

Eingabe: Hypergraph $G = (V, E)$, c, w, $(u_1, \cdots, u_k)$, $(o_1, \cdots, o_k)$
Aufgabe: Berechne Partition $P = \{P_1, \cdots, P_k\}$ mit
$u_i \leq c(P_i) \leq o_i$ mit minimalen Schnittkosten.

BALANCED MIN BIPARTITION

Eingabe: Hypergraph $G = (V, E)$, c, w, Balancefaktor ϵ
Aufgabe: Berechne Partition $P = \{P_1, P_2\}$ mit

$$\left| \frac{1}{2} - \frac{c(P_i)}{c(V)} \right| \leq \epsilon.$$

Komplexität

BALANCED MIN MULTIWAY PARTITION

- NP-vollständig

BALANCED MIN BIPARTITION

- mit nicht-trivialem Balancefaktor ϵ NP-vollständig
- mit trivialem Balancefaktor $\epsilon = \frac{1}{2}$ äquivalent zu MINCUT.

Komplexität

BALANCED MIN MULTIWAY PARTITION

- NP-vollständig

BALANCED MIN BIPARTITION

- mit nicht-trivialem Balancefaktor ϵ NP-vollständig
- mit trivialem Balancefaktor $\epsilon = \frac{1}{2}$ äquivalent zu MINCUT.

Aus diesem Grund besitzen Lösungs-Heuristiken in der praktischen Anwendung große Bedeutung.

Im Folgenden werden ausschließlich Algorithmen zur Lösung von BALANCED MIN BIPARTITION vorgestellt, da sich BALANCED MIN MULTIWAY PARTITION durch (k-1)-fache Anwendung bzw. einfache Modifikationen dieser Heuristiken bearbeiten lässt.

Vorgestellte Strategien

- Iterative Verbesserung (Fiduccia-Mattheyses)
- Multilevel-Algorithmen (hMETIS)

Iterative Verbesserung

Strategie der Iterativen Verbesserung

- Erzeuge zufällige, gültige Initialpartition
- Führe eine Serie schnittkostenminierender Operationen aus, bis eine Abbruchbedingung erreicht ist

Vertreter

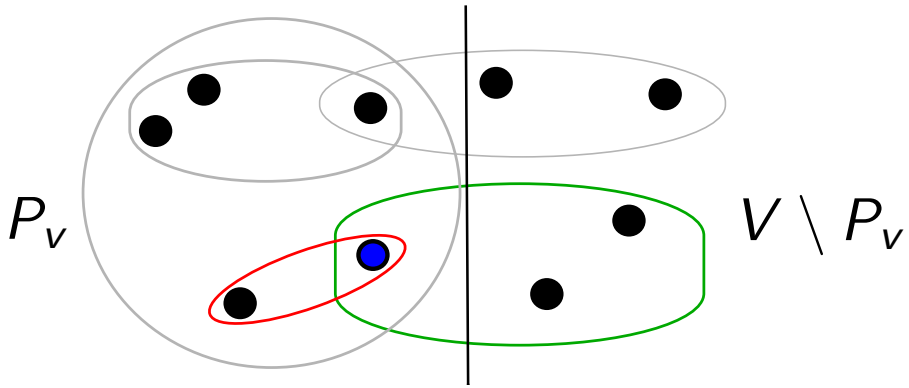
- Kernighan-Lin (BISECTION)
- Fiduccia-Mattheyses (BIPARTITION)

Fiduccia-Mattheyses

P := balancierte Initialpartition;
while *true* **do**
  Markiere alle Knoten als unverschoben;
  while $\exists$ *unverschobener Knoten* **do**
    Berechne Verschiebungseignungen (Kosten/Balance);
    v := unverschobener Knoten max. Güte;
    Verschiebe v virtuell in andere Partitionsmenge;
    Notiere Gesamtverbesserung;
  end
  if *Verbesserung beobachtet* **then**
    Ausführung max. schnittverbessernder Tauschsequenz;
  else break;
end

Die Gütefunktion

- Ursprünglich nur für Graphen definiert
- Kostendifferenz der inzididenten, die Partition verlassenden Kanten E_{ext} und partitionsinterner Kanten E_{int}
- Adaption für Hypergraphen redefiniert E_{ext} und E_{int}:
  - $E_{ext} = \{e \in E | e \cap P_v = \{v\} \wedge e \cap V \setminus P_v \neq \emptyset\}$
  - $E_{int} = \{e \in E | v \in e \cap P_v \wedge |e| > 1 \wedge e \cap V \setminus P_v = \emptyset\}$



Verwaltungsstrukturen

- Die potentiellen Schnittverbesserungen pro Knoten nach jeder Verschiebung neu zu berechnen ist unbefriedigend
- Fiduccia-Mattheyses verwendet Bucket-Datenstruktur
- Initialisierung:
  - Gewinnberechnung aller initialen Knotenverschiebungen
  - Eintrag der Knotens in Gewinnbucket
  - $\mathcal{O}(|V|)$
- Knotenauswahl:
  - Entnahme aus höchstem, gefülltem Bucket
- Pro virtueller Knotenverschiebung:
  - Adaption der Gewinnwerte für nicht-verschobene adjazente Knoten

Eigenschaften Fiduccia-Mattheyses

- Schneller greedy Algorithmus mit begrenzten Hillclimbing-Fähigkeiten
- Ergebnis ist lokales Minimum des Lösungsraumes
- Zufallseinflüsse durch Initialpartition und Auswahlentscheidungen zwischen Knoten gleicher Güte
- 'Kurzsichtige' Gewinnbewertung für Hyperkanten mit > 1 Knoten in beiden Partitionsmengen

Ergebnisverbesserung

- Mehrfachausführung und Wahl des besten Ergebnisses
- Lookahead-Gewinnbewertung mit Einbeziehung folgend möglicher Aktionen

Eigenschaften Fiduccia-Mattheyses

- Schneller greedy Algorithmus mit begrenzten Hillclimbing-Fähigkeiten
- Ergebnis ist lokales Minimum des Lösungsraumes
- Zufallseinflüsse durch Initialpartition und Auswahlentscheidungen zwischen Knoten gleicher Güte
- 'Kurzsichtige' Gewinnbewertung für Hyperkanten mit > 1 Knoten in beiden Partitionsmengen

Ergebnisverbesserung

- Mehrfachausführung und Wahl des besten Ergebnisses
- Lookahead-Gewinnbewertung mit Einbeziehung folgend möglicher Aktionen

Multilevel-Algorithmen

Motivation

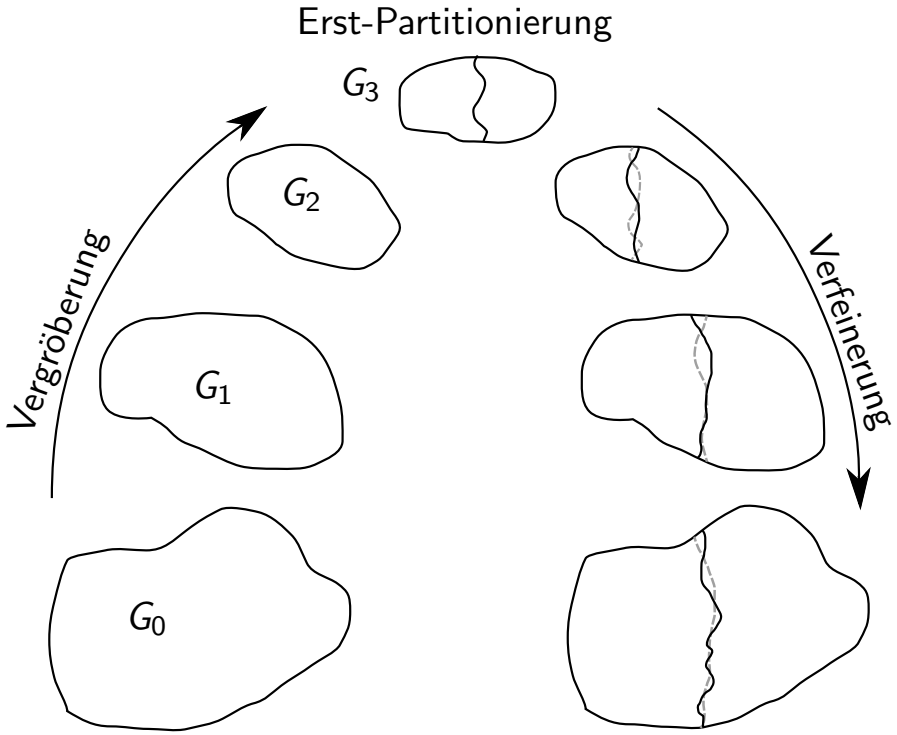
- Bisher Einsatz ausschließlich „lokal“ geprägter Entscheidungsfaktoren (Knotennachbarschaften, Kantenzugehörigkeiten)
- Multilevel-Algorithmen betrachten ein Problem auf mehreren Abstraktionsstufen
- Ursprünglich zur Beschleunigung der Graph-Partitionierung entwickelt
- Einteilung in 3 Phasen:
  Vergröberung, *Erst-Partitionierung*, *Verfeinerung*

Die Phasen eines Multilevel-Algorithmus

- Vergröberung
  - Erstellung mehrerer Abstraktionsstufen des Ausgangsproblems mit sinkender Komplexität
- Erst-Partitionierung
  - Partitionierung auf einfachstem Stellvertreter
- Verfeinerung
  - Rückpropagierung der Zwischenlösung über alle Abstraktionsstufen
  - Lösungsoptimierung unter Einbeziehung der gewonnenen Freiheitsgrade

Das Musterbeispiel: hMETIS

- Hypergraph-Adaption der METIS -Partitionierers für Graphen
- 1997 von Karypis et. al vorgeschlagen



Die Vergröberungsphase

Grundprinzip: Knotenkontraktionen

Idee:

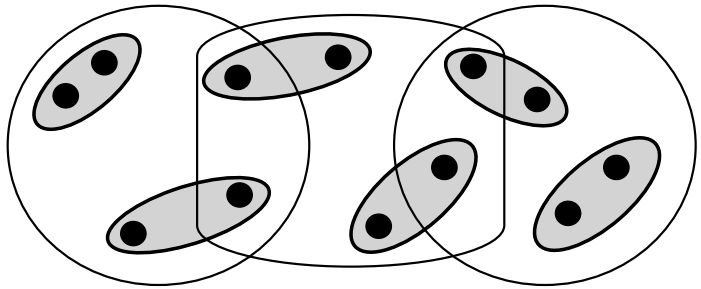
- Kontraktion durch teure Kanten verbundener Knoten, lässt Kantengewicht „aus dem Graphen fallen“, sobald Kanten verschwinden
- Dieses wird somit vom Schnitt ausgeschlossen
- Stellvertreterknoten erhalten Gesamtgewicht der kontrahierten Knoten

Konkurierende Ziele:

- Schnelle Abnahme des Gesamtkantengewichts
- Gewährleistung ausreichender Anzahl von Zwischenstufen für die Verfeinerungphase

Strategie 1: Kantenvergröberung

- Bildung disjunkter Knotenpaarungen in teuerstmöglichen Hyperkanten

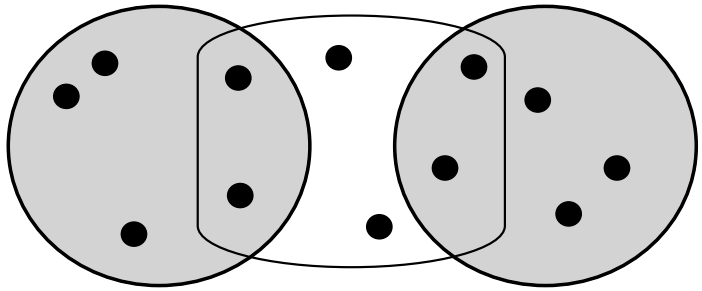


Eignung & Eigenschaften

- Halbierung der Knotenzahl
- gleichmäßiges jedoch sehr langsames Absinken des Kantengesamtgewichts

Strategie 2: Hyperkantenvergröberung

- Kontraktion disjunkter Hyperkanten in Kostenreihenfolge

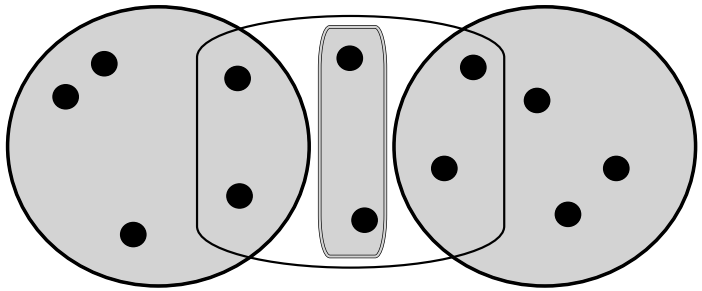


Eignung & Eigenschaften

- gute Kantengesamtgewichtsabnahme
- ungleichmäßige Knotengewichtsverteilung (Stellverteterknoten vs. unberührte Knoten)

Strategie 3: Modifizierte Hyperkantenvergröberung

- Hyperkantenvergröberung plus Kontraktion der in überdeckten Hyperkanten verbleibenden Knoten

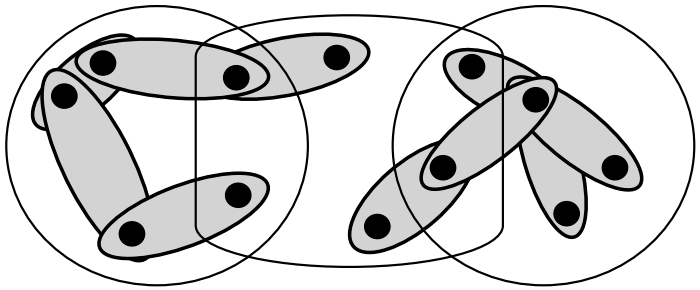


Eignung & Eigenschaften

- gute Kantengesamtgewichtsabnahme
- gleichmäßige Knotengewichtszunahme

Strategie 4: First-Choice-Vergröberung

- Pro Knoten randomisierte Paarungsentscheidung mit Nachbarknoten der teuersten inzidenten Kante



Eignung & Eigenschaften

- sehr schnelle Kantengesamtgewichtsabnahme (Begrenzung nötig)
- für bestimmte Hypergraphen besser geeignet

Abbruch der Vergröberung

- Keine weitere Vergröberung falls Schwellwert für Knotenanzahl unterschritten

Erst-Partitionierung

- Kleiner Hypergraphenumfang $\rightarrow$ exakte Lösung möglich
- Besser: Randomisierte Erstellung einer balancierten Lösung
  - schnell
  - Optimalität der Erst-Partition enthält keine Aussage für Originalgraphen
  - Vielzahl von folgenden Verfeinerungsschritten führt zu starker Veränderung
- Überschaubare Problemgrößen:
  Erzeugung mehrerer Erst-Partitionen, bei späterer Elimination schlechter Kandidaten, möglich

Die Verfeinerungsphase

- Rückpropagierung der Zwischenpartition über alle Komplexitätsstufen
- Auf jeder Stufe Iterative Verbesserung (*Fiduccia-Mattheyses*) um gewonnene Freiheitsgrade zu nutzen
- Wenige Iterationen nötig, aufgrund bereits guter Lösung der Vorstufe

Balancierung bei hoher Abstraktion

Problem:

- hohe Abstraktionsstufen enthalten schwere Knoten
- Eng gesetzte Balance-Anforderungen schränken die Anzahl der Partitionsmöglichkeiten sehr stark ein

Behandlung:

- Beginn mit stark relaxierten Balanceanforderungen
- Vorgaben erreichen erst nahe des Originalhypergraphen das Niveau der ursprünglichen Aufgabenstellung

Erweiterungen

- Noch immer große Zufallseinflüsse
- Mitführung mehrerer Kandidatenpartitionen auf hohen Abstraktionsebenen
- Erneute Vergröberungs-/Verfeinerungs-Iteration, jedoch unter Berücksichtigung der aktuellen Partition während der Vergröberungsphase
  - V-Durchlauf: vollständige Vergröberungs-/Verfeinerungs-Iteration ausgehnd vom Originalhypergraphen
  - v-Durchlauf: ausgehend von hoher Abstraktionsstufe, schnell, mehrfach wiederholbar

Ergebnisse

- hMETIS ist einer der erfolgreichsten Hypergraphen-Partitionierer

Erweiterungen

- Noch immer große Zufallseinflüsse
- Mitführung mehrerer Kandidatenpartitionen auf hohen Abstraktionsebenen
- Erneute Vergröberungs-/Verfeinerungs-Iteration, jedoch unter Berücksichtigung der aktuellen Partition während der Vergröberungsphase
  - V-Durchlauf: vollständige Vergröberungs-/Verfeinerungs-Iteration ausgehnd vom Originalhypergraphen
  - v-Durchlauf: ausgehend von hoher Abstraktionsstufe, schnell, mehrfach wiederholbar

Ergebnisse

- hMETIS ist einer der erfolgreichsten Hypergraphen-Partitionierer

Anwendung zur Dokumenten-Clusterung

Association Rule Hypergraph Partitioning (Han et al, 1997)

1. Vorverarbeitung
2. Extraktion häufiger Itemmengen (Dokumentengruppen) aus (boolscher) Wort-Dokument-Matrix. Worte bilden Transaktionen über Dokumenten
3. Modellierung der Dokumentenbeziehung in geeignetem Hypergraph, mit Hilfe der gefundenen häufigen Itemmengen
4. k-WAY Partition oder wiederholte Bipartition mit *fitness*-Maß als Abbruchkriterium

Modellierung des Hypergraphen

- Knoten sind mit Dokumenten assoziiert
- Hyperkanten korrespondieren mit häufigen Itemmengen
  - Gewicht ergibt sich als Funktion über den Konfidenzwerten der Assoziationsregeln, welche sich innerhalb der häufigen Itemmenge aufstellen lassen
  - In der Praxis Einschränkung auf Assoziationsregeln mit einelementigen Regelkopf
- Vorteil gegenüber Wortkanten:
  - adaptierbare Kantenzahl und inhaltserhaltende Reduktion

Partition

Einsatz nach Anwendungszweck:

- Direkte k-WAY Partition zur Erzeugung von k Dokumentenclustern
- Wiederholte Bipartition bis *fitness*-Kriterium aller Cluster über Schwellwert liegt:

$$fitness(C) = \frac{\sum_{e \subseteq C} w(e)}{\sum_{|e \cap C| > 0} w(e)}$$

Vielen Dank für ihre Aufmerksamkeit.

Alternativen zur Hypergraphpartitionierung

Abbildung auf Graphen

- Erstellung eines Stellvertretergraphen mit gleichen Schnitteigenschaften
- Anwendung einer Heuristik zur Graph-Partitionierung

Beispiel: Cliquenersetzung

- Ersetzung der Hyperkanten durch Cliquen
- gleichmäßige Aufteilung der Hyperkantenkosten
- naheliegend, einfach, *problematisch*
- schnittkostengleiche Gewichtung unmöglich, da je nach Schnittverlauf die Zahl der geschnittenen Stellvertreterkanten variieren kann

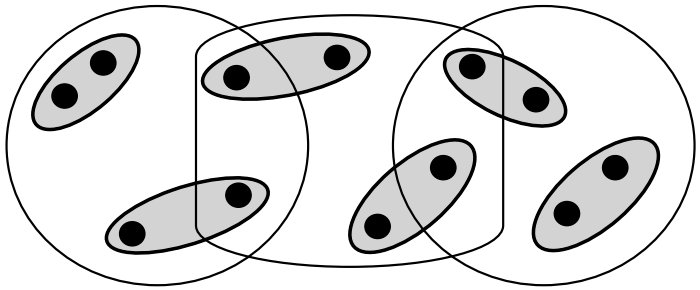
Abbildung auf Graphen

- Erstellung eines Stellvertretergraphen mit gleichen Schnitteigenschaften
- Anwendung einer Heuristik zur Graph-Partitionierung

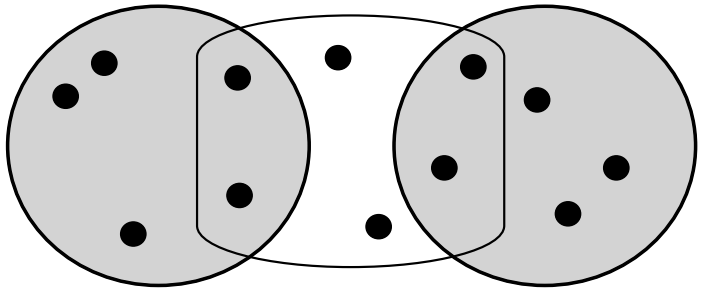
Bemerkungen

- Korrekte Ansätze nutzen zusätzliche Knoten und negativer Kantengewichte
- Die folgende Graph-Partitionierung kann jedoch auch nur heuristisch erfolgen (ist ebenfalls NP-vollständig)
- Besser: Direkte Übertragung von Graphen-Heuristiken auf Hypergraphen

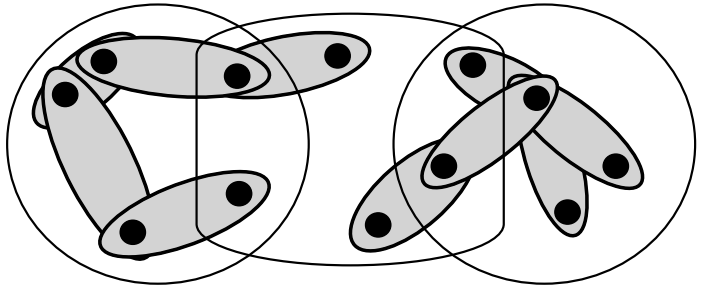
Vergröberungsstrategien



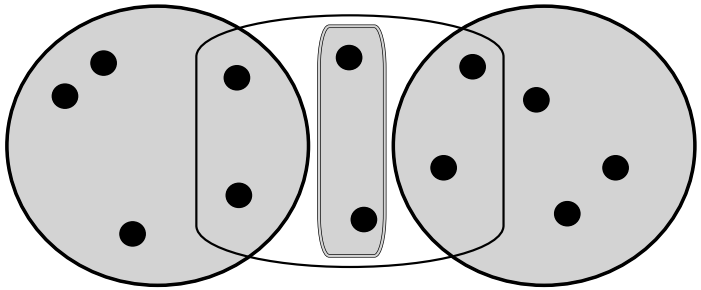
(a) Kantenvergröberung



(b) Hyperkantenvergröberung

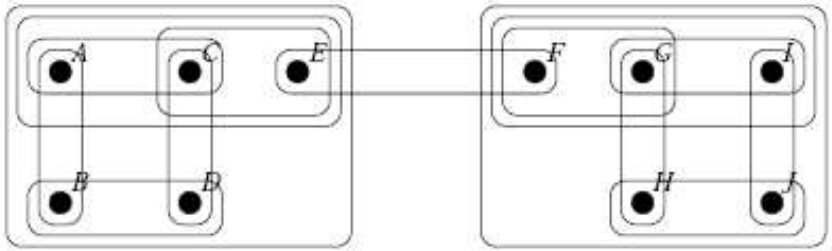


(c) First-Choice-Vergröberung

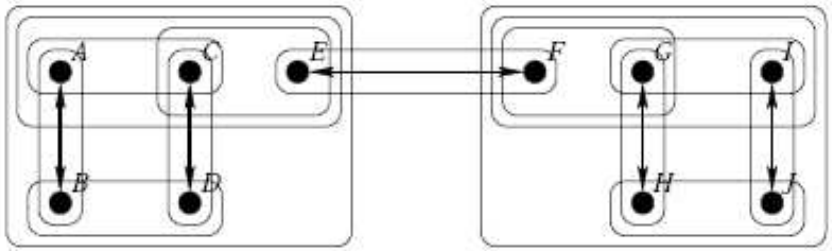


(d) Mod. Hyperkantenvergröberung

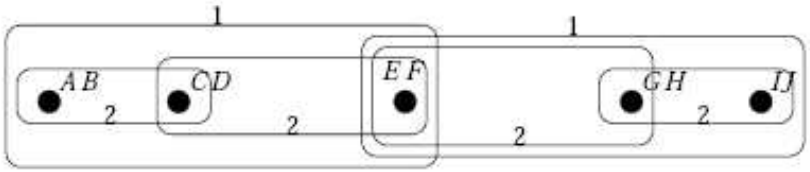
Vorteil EC-Vergröberung



(a) Initial Hypergraph



(b) Groups Determined by Edge–Coarsening



(c) Coarse Hypergraph