HOKKAIDO UNIVERSITY

Title	クリークマイニングとその応用 ： 大規模データの活用
Author(s)	宇野, 毅明
Description	ERATO湊離散構造処理系プロジェクトシンポジウム(第2回) ： 第73回情報処理学会全国大会イベント企画. 2011年3月2日(水). 東京工業大学 大岡山キャンパス.
Relation	2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録. p.496-499.
Issue Date	2011-06
Doc URL	https://hdl.handle.net/2115/48330
Type	conference presentation
File Information	00.keynote03.uno.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

クリークマイニングとその応用
～ 大規模データの活用 ～

宇野 毅明 （国立情報学研究所
＆総合研究大学院大学）

2011年3月2日 情報処理学会 湊ERATOシンポジウム

どうして大規模データ？

- Web サーチのように**「全てのデータを取っておくこと」**に意味があるのでない限り、データを大量に保存せずともいいんじゃない？

← データ解析して全体的な傾向を見るのなら、ある程度たくさんあつまれば十分でしょう。（世論調査、出口調査、、、）

- 統計理論が発達してるので、少ないデータでも全体の傾向を十分に捕らえられるようになってきている （*1億件 → 1万件*）

→ 大規模データの蓄積は不要？？？

- 全体的なものに関しては、確かにその通り。
でも、部分的な特徴は、大規模データが必要

エラーの訂正

- OCR（スキャナ）で文章を読みました
「実は大規**撲**データの解析で、、、」 ← まちがってる

- と思いきや、「実は大規**撲**データの解析で大関の、、、」だったりすると、「実は大**相**撲データの解析で大関の、、、」となるべき

→ 前後を見ないと、何が正解か分からない
→ 意味を解析しないと分からない？？？

- でも、事例があれば推測はできる
「先月の大相撲データの解析で大関の取り組みが、、、」
「実は大相撲データを解析を大関の成績にあてはめて、、、」
- 「**大**の次に何が来る可能性が高いか」程度の統計では難しい

数の暴力

- ゲノム情報の読み取りはエラーがつきもの。いろいろな方法で精度を高めようとするが、、、

自信がない　自信がない

....ATCCGCTAGGTGAATATGCGC...
....ATCCGCTAGGAGAATATTCGC...
....ATCCGCTAGGAGAATATGCGC...
....TTCCCCTAGGGGAATATTCGC...
....ATCCGCTAGGAGAATATTCGC...
....ATCCGCAAGGAGAATATTCGC...
....ATCCGCTAGGAGAATATCCGC...

- **気にせずに、たくさん読んでしまえばいい！！**

どうして使わないの？

＋ 大規模データの利用で（ある種自動的に）精度を高められる
＋ 非常にまれな（1万分の1）事象でも、1億件のデータでは1万件もある （解析するのに十分）

- なのに、それほど大規模データをがんがんと解析しているわけではない （企業には、眠っているデータが山ほど）

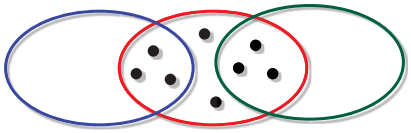
- 大きな理由は、***「計算が大変だから」***
← 1億件のデータの、どれとどれが似ているかを調べるには、1億×1億／2のチェックが必要。一秒に1億回比較しても2年くらいかかる

- 実際はもっといい方法を使うが、それでも1週間とか。
（大量の似たものグループを見つけるなど、とてもとても、となる）

似たものグループ

- 「数の暴力」をするには、似たものグループを作る必要がある
← **「クラスタリング」**という問題

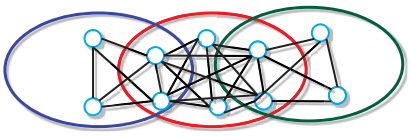
- しかし、通常の「クラスタリング」では、データを完全に分割してしまう （2つ以上のグループに属する人がいない）



- 重なりを許したグループを列挙したい

クリーク

- 似たものの間に線を引いて、グラフを作る
  - ➔ 同じグループに属するものは、線で結ばれているだろう
  - ➔ 全てのペアがお互い結ばれているもの達がグループ？
    （こういう頂点の集合を**クリーク**という）



- 他のクリークに含まれない、極大なクリークを見つければ、グループが見つけられるだろう

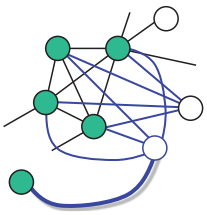
問題 与えられたグラフの極大クリークを全て列挙せよ

一個見つけるだけでも…

- 極大クリークは、$O(n^2)$ 時間で見つかる （**多項式時間**）
  （全ての頂点に隣接する頂点を1つずつ追加する）
- しかし、大規模データでは、n^2 時間はとんでもなく長い

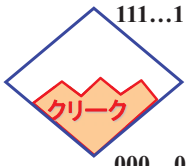
- 現在のクリークに隣接する頂点の集合を保持する
  頂点を追加したら、その頂点に隣接するものだけ残す
- $O(平均次数^2)$ 時間程度になるので、疎なグラフなら大丈夫

- あとは、探索の工夫が必要
  - ➔ 適当に探すと、同じのばかり何回も見つけてしまって、いつまでたっても全部見つけられない



クリークの単調性

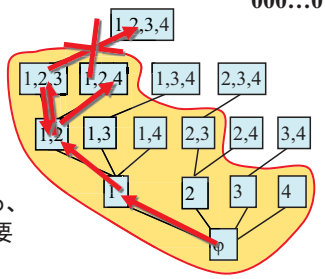
- クリークの部分集合はクリーク
  - ➔ 単調性が成り立つ



➔ 原点を出発して山を登り、クリークでなくなったら、戻って、他の方向に登る、というバックトラック式の探索で列挙できる

- しかし、極大が少なくても、たくさんのクリークが含まれることが多く、現実的な時間で終わらない

- 極大クリークだけ見つけるには、なんらかの工夫（うまくジャンプする、不要なところをスキップする）が必要



工夫をする

- 工夫の仕方はいくつかあるが、代表的なもの

＋ 富田アルゴリズム *（元電気通信大学 富田悦次先生）*
（先の探索を改良し、効果的な枝刈り法を組合わせる）

＋ 牧野宇野アルゴリズム *（with 東京大学 牧野和久先生）*
（極大クリークの隣接性を上手に定義し、効率良く探索）

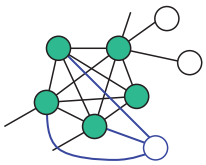
- 世界的に見て、優秀なのはこの2つと思って良いだろう
（大規模グラフデータでも、現実的に高速に動く、という意味で）

- 両方とも、実装（プログラム）が宇野のホームページにある
(http://research.nii.ac.jp/~uno/codes-j.html)「宇野毅明」「公開プログラム」「データマイニングの簡単」「mace クリーク」などで検索

隣接性の導入

- クリークに「えらい順」を決める （頂点添え字の辞書順）

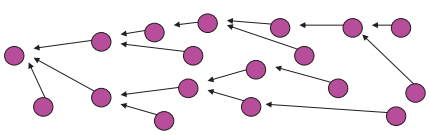
- 極大クリーク Kの頂点を、添え字（ID）の大きい方から順に抜いていく



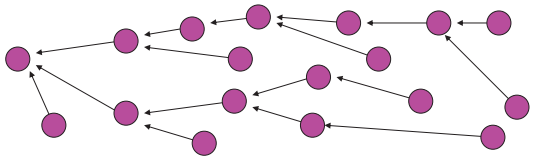
➔ 自分よりえらい極大クリークに含まれるようになったら、それを隣（親）とする

- （一番偉いクリーク以外）、どの極大クリーク にも一つ親がある

➔ 巡回的でない（木型の）探索路ができる



「逆探索」



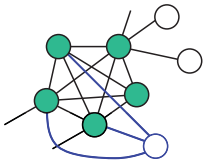
- こういう探索ルートが（暗に）決まっているのなら、あとはこのルートをなぞればよい
- なぞるには、（今訪れている頂点の）子供が見つかればよい

（実は、子供を見つけるのは普通簡単ではなく、だから効率的な列挙は難しい

- この親子関係の場合、「頂点を追加、邪魔者を削除」で求まるのでうまくいく

頂点を追加して子供を見つける

- **「親」**に1つ頂点を追加する
- つながっていない頂点はじゃまなので、消す
- それを含む一番偉い極大クリークを見つける
  （**「子供の候補」**になる）



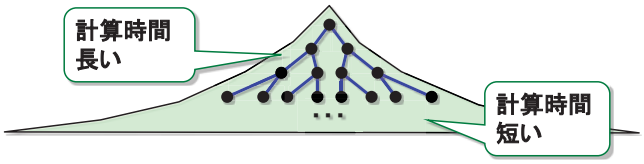
- 「子供の候補」の「親」が自分だったら、それは自分の子供である
  （「子供候補」が他人の子供である可能性もある、ということ）

子供候補が少ないので効率的に計算できる
（クリークにつながる頂点だけ）

- 一般に、現実データでは、クリークにつながる頂点は少ない

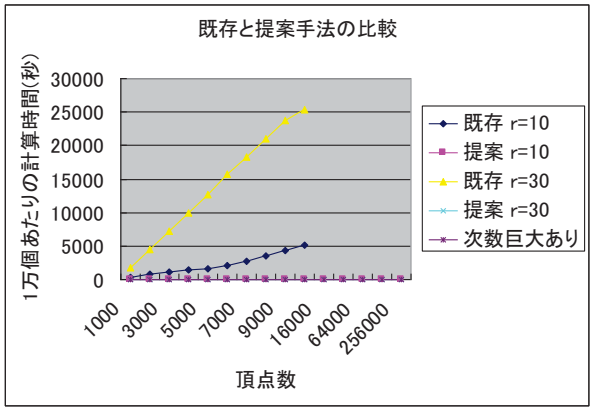
末広がり性

- 列挙は、各反復で複数の再帰呼び出しをする
  - ➔ 計算木は、下に行くほど大きくなる
  - ➔ 計算時間を支配するのは一番下の数レベル

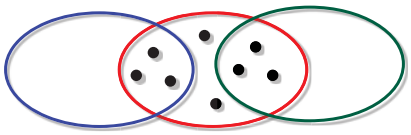


平均してしまうと、1つあたりの計算時間は非常に短くなる。
そのため、1秒間に1万個以上の極大クリークを見つけることが可能。（どんな大きさのデータでも、だいたい）

新旧アルゴリズムの比較



クラスタ マイニング



- 類似する部分に線を引いたグラフの極大クリークを見つけ、クラスタの列挙を行う

＋ 現実データでは、似ている物は似ているし、似ていないものは似ていない、ということが多く、データにムラがあるため、比較的キレイにまとまることが多い
（アイテムセット、画像、文字列、ゲノム・・・）

＋（2乗時間かけないグラフ作成が重要）

Webテキストからデータマイニング

- Webテキストの、類似する部分に線を引いてグラフを作り、クリークを見つけて多数決を取ると (200万文字、10分)

#年0#月200#年0#月2006年0#月2006年0#月2006年0#月2006年0#月2006年0#月200#年0#月200#年12月2005年…

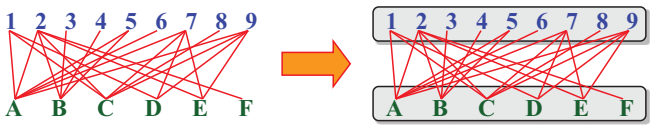
##, ###円(税別、送料別)Paul Smith Men's／ポールスミス メンズサイズフェイス:H約4. 5cm×W約3. #cm、厚み約#. #cm(リューズ除く)ベルト:幅約2. #cm腕まわり:最大約###

#組の成立となりました。## #月1#日(土)男性12名:女性1#名のご参加で、5組の成立となりました。## 4月7日(土)男性#0名:女

- 面白い物が見つかる。既存のデータマイニング手法では、ちょっとずつ違う解が大量に出るし、何年かかっても計算が終わらない

2部クリークの列挙

- 2部グラフのクリークは、2部それぞれの頂点集合をクリークにすることで、クリークと1対1の対応が付く

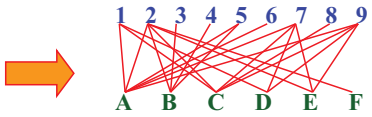


- 変換したグラフの極大クリークを列挙
  - ➔ 極大2部クリークの列挙
- クリーク化でグラフが密になると困るので、仮想的に枝を追加

アイテム集合マイニングができる

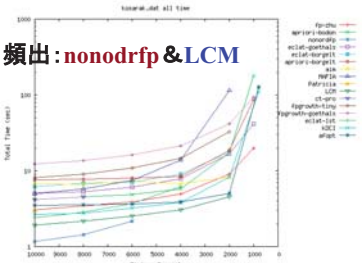
- 多く(閾値以上)の項目に含まれるアイテム集合を**頻出アイテム集合**という
- アイテム、項目を頂点とし、包含関係を枝とした2部グラフで、項目側が閾値以上の頂点を含む2部クリークは、頻出アイテム集合に対応

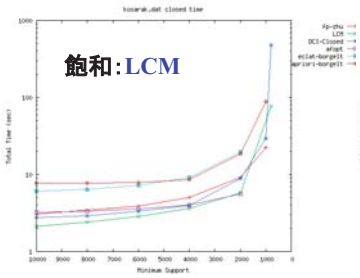
$D=$
A: 1,2,5,6,7,9
B: 2,3,4,5
C: 1,2,7,8,9
D: 1,7,9
E: 2,7,9
F: 2

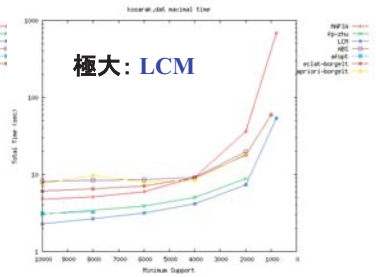


- データベースの再帰的縮約操作が末広がり性と実に良くフィットするため、非常に高速で列挙できる

クリックストリームデータ(疎)



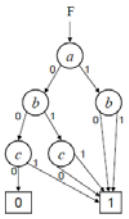




ZDD に “ピッタリ”

- 頻出アイテム集合や極大クリークは比較的似た共通の構造、規則的な構造を沢山持つ
  - ➔ ZDD と相性が良い。出力は ZDDでかなり小さくなる

Record ID	Tuple
1	a b c
2	a b
3	a b c
4	b c
5	a b
6	a b c
7	c
8	a b c
9	a b c
10	a b
11	b c



- 複数のデータの頻出アイテム集合の間で、差分をとったり様々な演算ができる ➔ 多様な解析が可能になる

まとめ

- 大規模データの「数の暴力」で効果的なデータ利用をしよう
- 何も知識がないところでは、「相互関係」しか頼るものがない。似たものグループを見つけて利用しよう
- 似たものグループを見つけるために、極大クリークを列挙しよう
- 隣接性をうまく定義して、上手に探索しよう
- 変化球で、頻出文字列や頻出アイテム集合を見つけよう
- ZDDとの組合せで、多様で深い解析を効率的に行おう