VLDB2015勉強会
[session 4]Graph Mining

佐々木　勇和

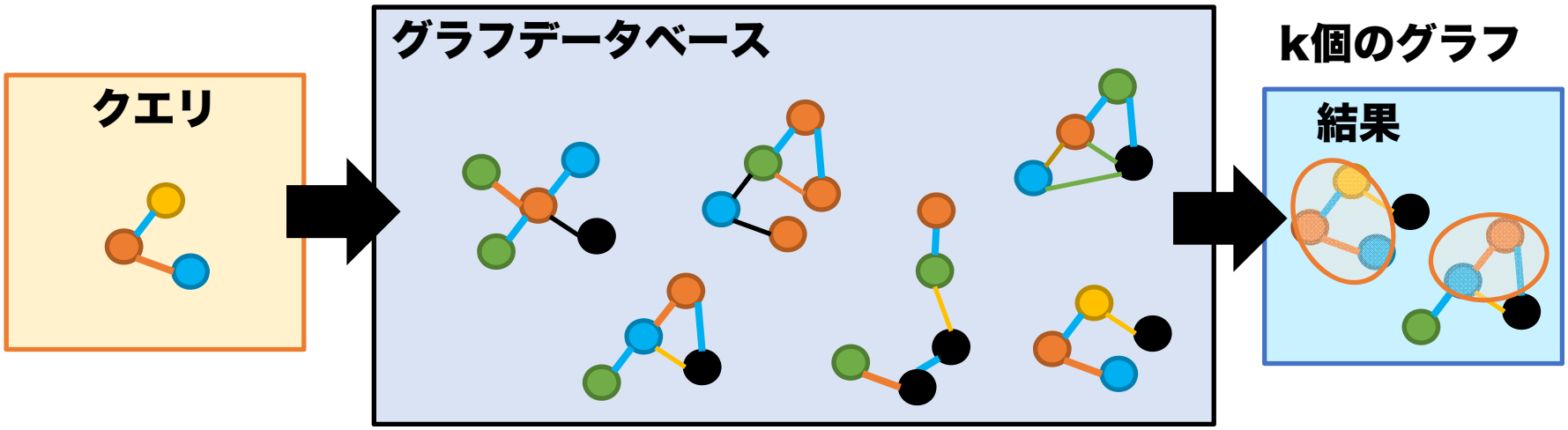
NAGOYA
UNIVERSITY

Session6

1. **Leveraging Graph Dimensions in Online Graph Search**
   Yuanyuan Zhu, Jeffrey Xu Yu, LuQin
2. **Event Pattern Matching over Graph Streams**
   Chunyao Song, Tingjian Ge, Cindy Chen, Jie WangLisi
3. **An Efficient Similarity Search Framework for SimRank over Large Dynamic Graphs**
   Yingxia Shao, Bin Cui, Lei Chen, Mingming Liu, Xing
4. **Growing a Graph Matching from a Handful of Seeds**
   Ehsan Kazemi, Seyed Hamed Hassani, Matthias Grossglauser
5. **Association Rules with Graph Patterns**
   Wenfei Fan, Xin Wang, Yinghui Wu, Jingbo Xu

Leveraging Graph Dimensions in Online Graph Search

- 背景：
  - グラフ操作は処理が重たいので，なんとかしたい．
- 問題：
  - Maximum Common Subgraph(SCM)
    - 2つのグラフの最大の重複部分を探索
  - 類似度が上位k個のグラフを探索



アプローチ

重いグラフ操作から多次元データベースへの軽い処理に変換

- グラフを多次元データベースに射影
- 距離と構造を保存

グラフをバイナリベクタに変換

- $\phi(g_i) = \boldsymbol{y_i} = (y_{i1}, y_{i2}, \cdots, y_{ip})$
  - もし y_{ir}=1なら, f_r はg_i のサブグラフ
- ベクトルの距離比較でグラフの距離をはかる

課題

- Q1: どのグラフを要素とするか？（全部は多すぎ）
- A1: 頻出サブグラフから距離を保存できるものをp個選択
- Q2: 効率的な探し方は？

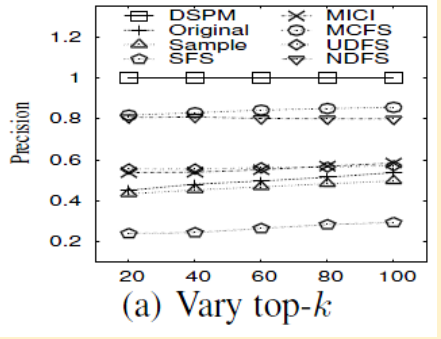
アルゴリズム

- DSPM法: 距離のずれが小さくなるまで，最大k回の繰り返しを実行
- DSPMap法：データベースを分割して，コスト削減

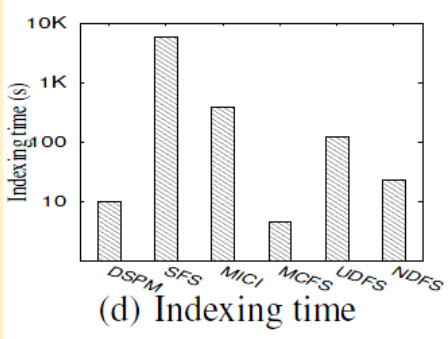
実験

- 実験データ
  - 化学化合物データベース from PubChem
  - 人工データ using Graphgen
- 実験結果
  - DSPMが既存手法より精度が良くてインデックス構築が早い
  - クエリ実行時間はどの手法もほとんど一緒
  - 近似手法は精度の低下が少しで，インデックス構築が効率的

既存手法との比較

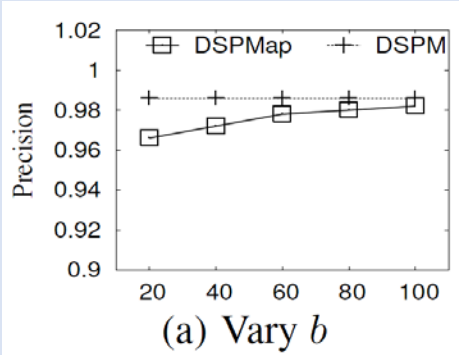


(a) Vary top-k

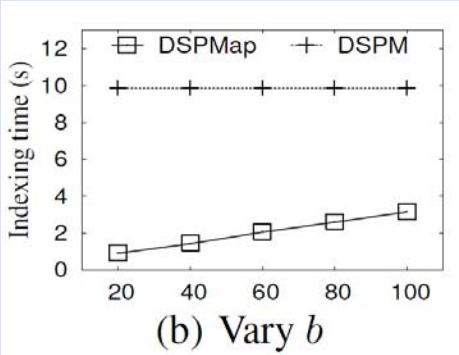


(d) Indexing time

近似手法との比較



(a) Vary b



(b) Vary b

b : グラフ分割

Event Pattern Matching over Graph Streams

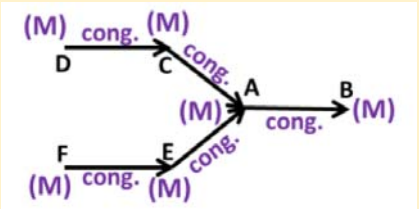
- グラフストリーム
  - 枝と節点の追加，削除，更新が動的に発生する．
- クエリ
  - 順序制約付きのサブグラフマッチング

道路ネットワークの例

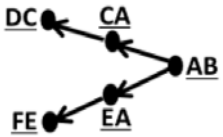
渋滞の原因となる道路を発見したい

クエリ

サブグラフ



順序制約



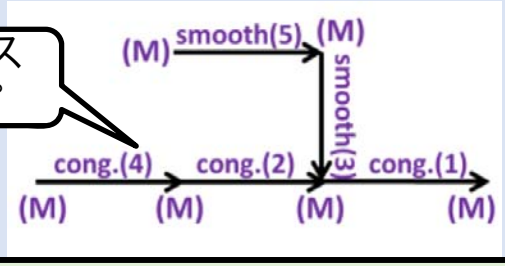
時間幅

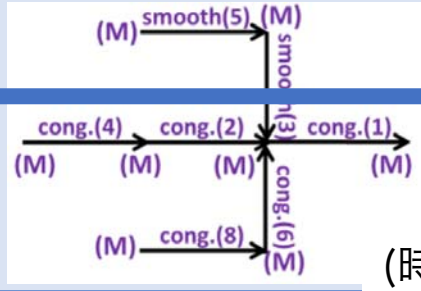
$w = 5$ minutes

発生から5分経過した枝は無視

グラフストリーム

タイムスタンプ





マッチ

(時間1と8が5min以内なら)

問題定義

- 課題
  - インデックスは使えない
  - グラフが変化する度にサブグラフマッチングするのは無駄
- ベースライン手法
  - グラフに変化がある度にサブグラフの候補集合全部列挙
  - 順序制約を考慮して，正解集合を計算

計算回数をいかに減らすかが重要

提案手法

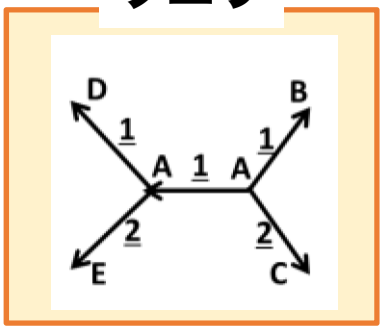
Number Theoretic Signature Method

- クエリグラフとグラフストリームを数値化
  - 枝と節点に数値ラベル付け，入出次数を数値化し，総乗する.
- グラフストリームがクエリで割切れるならマッチの可能性
  - 割切れる場合，Baselineを実行
    - クエリの要素は全て出現するが，バラバラに存在するかも.

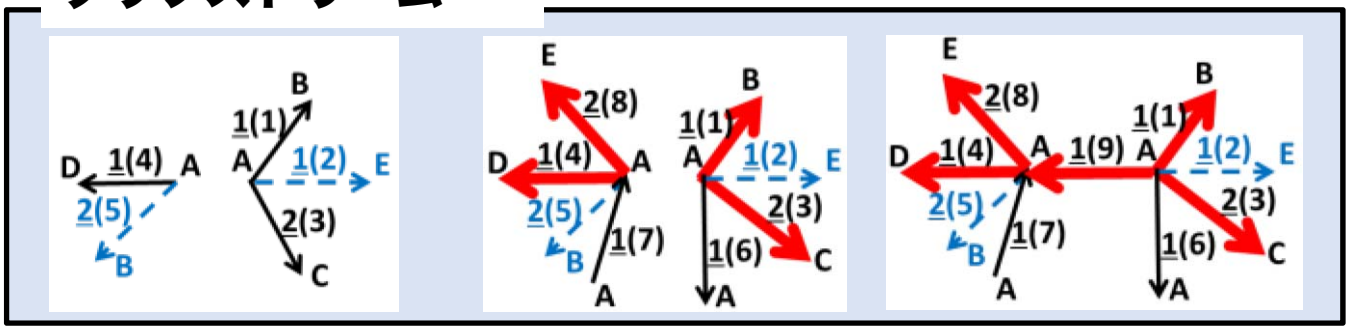
Coloring Algorithm

- 枝を条件に合わせて色づけ
  - クエリグラフに合う枝->黒
  - 全ての隣接枝もクエリグラフに合う枝->赤
- マッチを見つけたら，時間制約のチェック

クエリ



グラフストリーム



実験

- 実験データ
  - 道路ネットワーク
  - 電話回線ネットワーク
  - 特許データ
  - Twitter
- 実験結果
  - Coloringのスループットが最もよい.
  - パラメータが変わると，Signatureの方がよくなることもある.

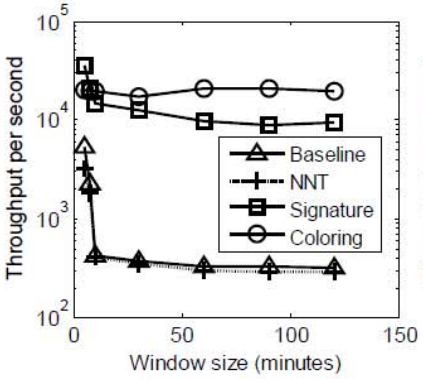


Fig 9 Varying w (Road)

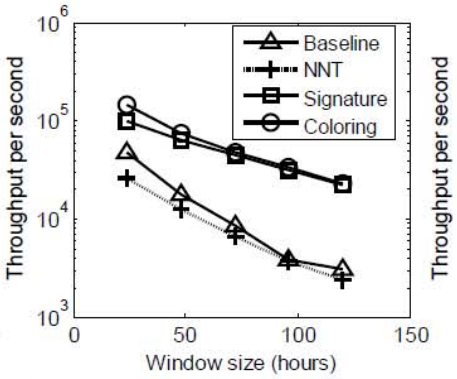


Fig 10 Varying w (Phone)

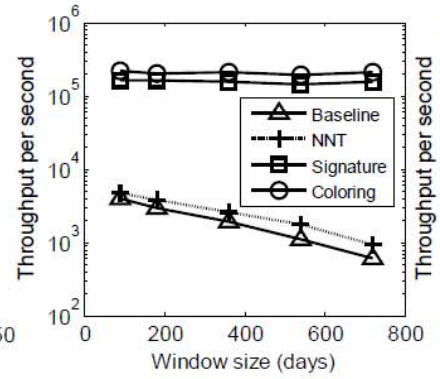


Fig 11 Varying w (Patent)

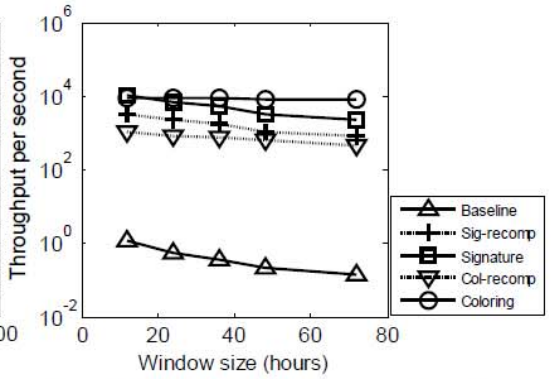


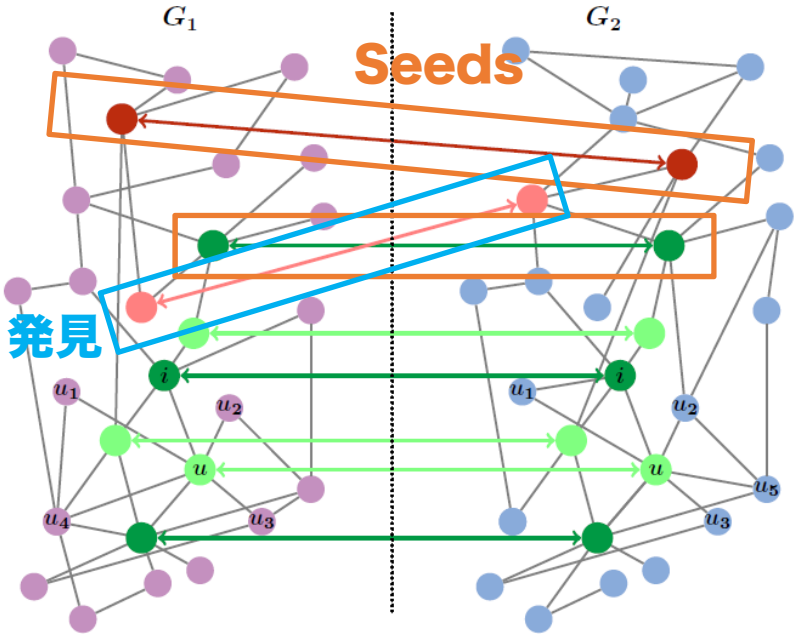
Fig 12 Various settings (Twitter)

Growing a Graph Matching from a Handful of Seeds

- 2つのグラフを比較して節点のマッチを発見
  - 例：TwitterとFlicker上で同じユーザを見つける.

- **Percolation Graph Matching Method**が有効
  - 事前のマッチ（Seed）から徐々にマッチを探していく.
    - 例：ユーザAのTwitterとFlickerのアカウントが最初から既知
  - Seedがたくさん必要で，多いほうが性能いい.

- **Seedを見つけるのが大変!!**



提案アルゴリズム

ExpandOnce アルゴリズム

- 正しい既知のSeedから NoisySeedsを生成して,
- Seedが間違ってる場合を考慮して実行する.

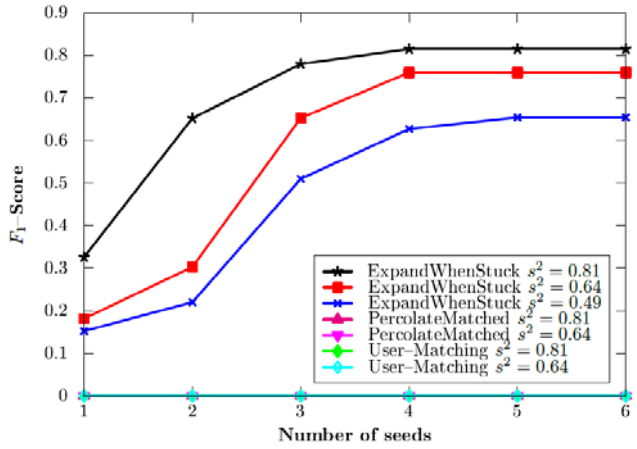
ExpandWhenStuck アルゴリズム

- 正しい既知のSeedの隣接の中から,
- 最もよさそうなのを一つずつ追加していく.

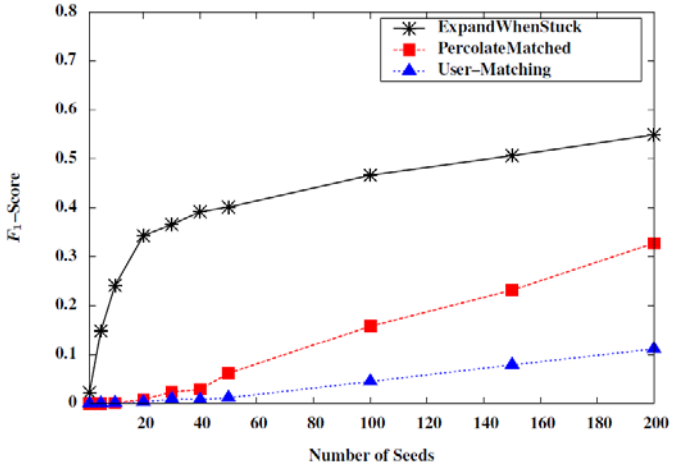
実験

- 実験データ
  - Youtube graph
  - EPFLのメールのスナップショット
  - Pokec (スロバキアで最も人気のSNS)
  - Gowalla
- 実験結果
  - 少ないSeedでマッチを見つけることができる.
    - 割愛：ExpectedWhenStuckの方がExpectedOnceよりもよい.

youtube

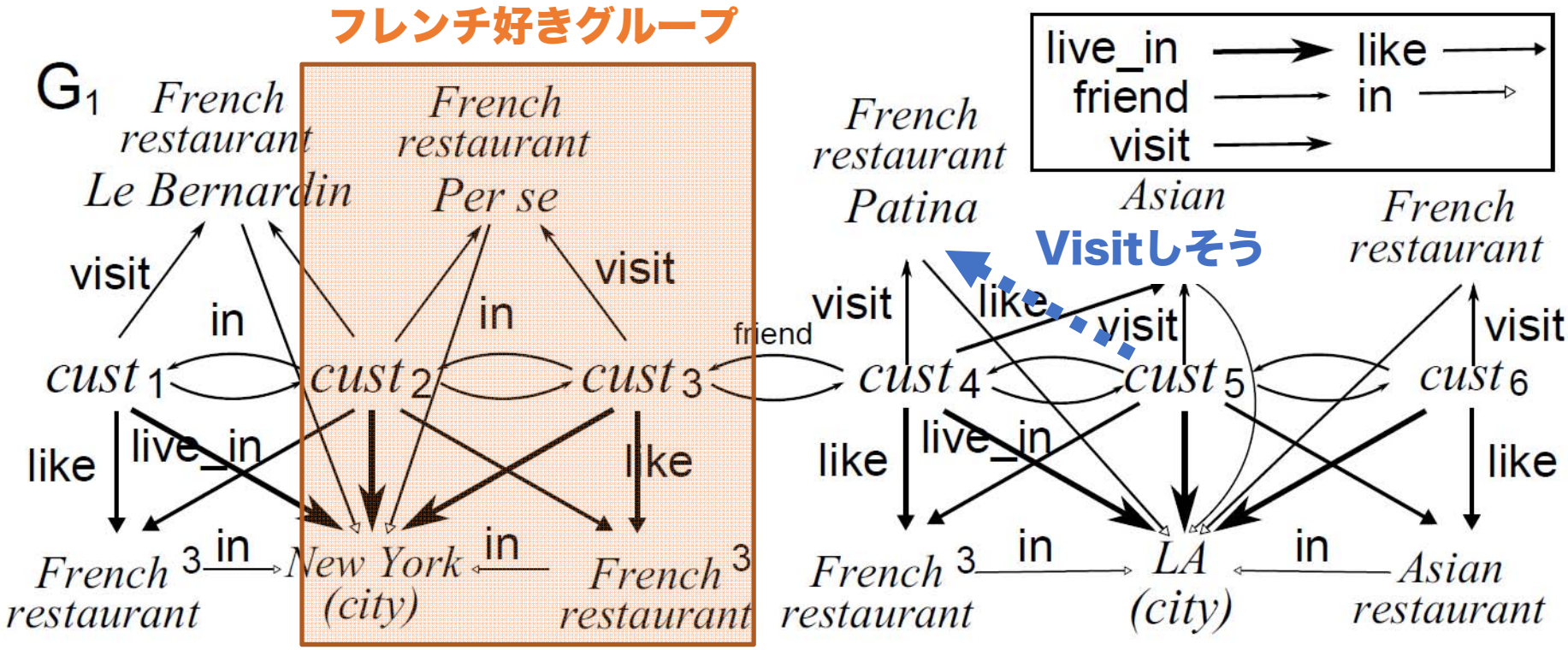


Gowalla



Association Rules with Graph Patterns

- ソーシャルグラフのためのGraph-pattern association rules (GPARs)の提案
- **よくあるパターン**の発見と**潜在的な枝**を発見する



提案

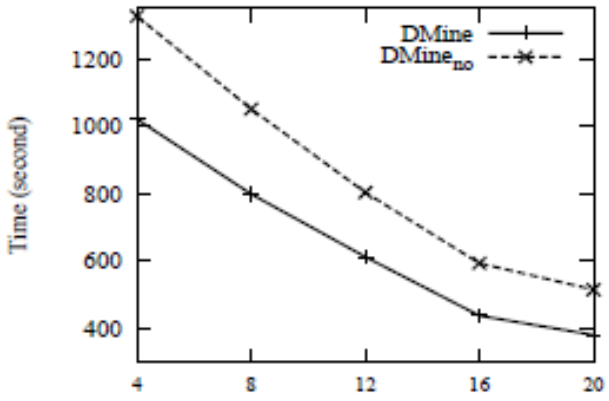
- ソーシャルグラフ用に定義し直し.
  - Support (支持度)：サブグラフとの関連を考慮
  - Confidence (確信度)：未入力の可能性（本当は枝があるかも）を考慮

- **Diversified Mining Problem**を提案
  - 多様性を考えた，k 個のパターンを発見
- **Entity Identification Problem**を提案
  - 確信度が閾値以上の潜在的な枝を発見

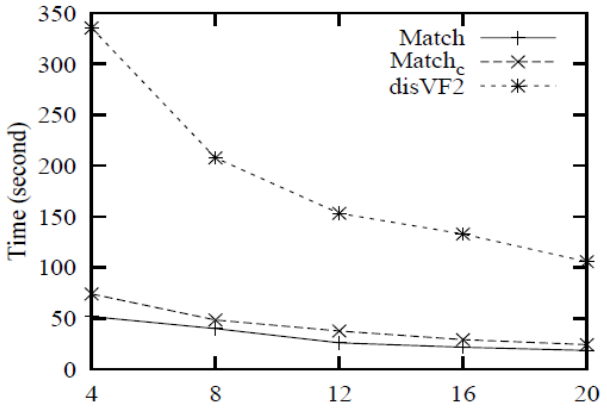
それぞれの並列処理アルゴリズム
DmineとMatchを提案

実験

- 実験データ
  - Pokec
  - Google+
- 実験結果
  - プロセッサ数に比例して計算時間が減少



(b) DMine: Varying n (Google+)



(i) Match: Varying n (Google+)