ICDE勉強会 [session 6]
Top-k and Pattern Mining

佐々木　勇和

NAGOYA
UNIVERSITY

Session6

[DE150609.pdf]

– Mining Maximal Cliques from an Uncertain Graph
Arko Provo Mukherjee, Pan Xu, Srikanta Tirthapura

[DE150241.pdf]

– Temporal Spatial-Keyword Top-k Publish/Subscribe
Lisi Chen, Gao Cong, Xin Cao, Kian-Lee Tan

[DE150440.pdf]

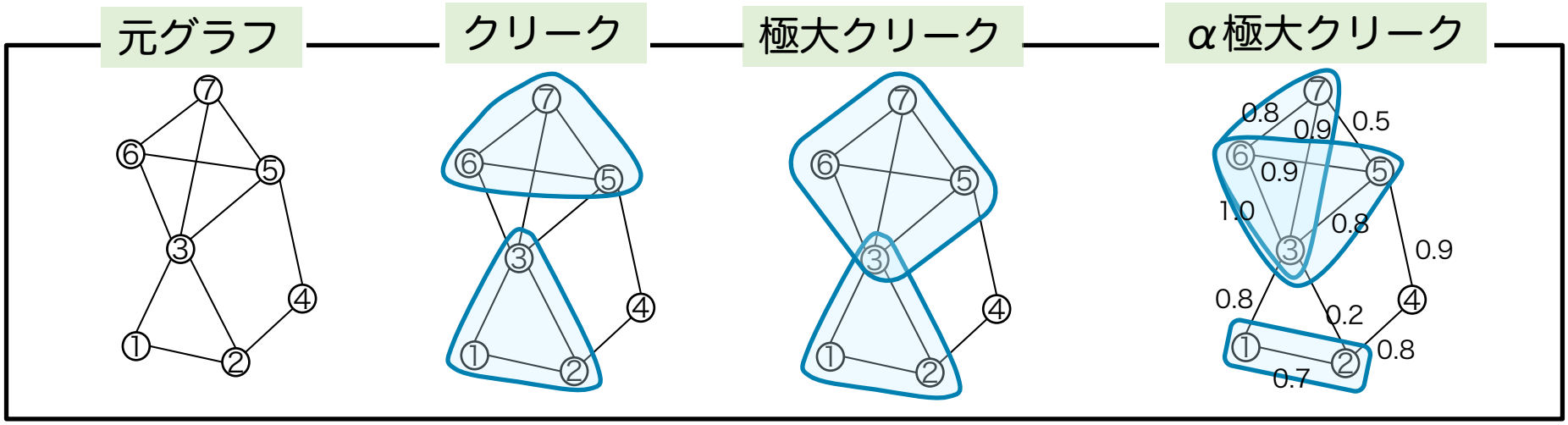
– Finding Top-k Local Users in Geo-Tagged Social Media Data
Jinling Jiang, Hua Lu, Bin Yang, Bin Cui

[DE150361.pdf]

– Answering Why-Not Questions on Spatial Keyword Top-k Queries
Lei Chen, Xin Lin, Haibo Hu, Christian S. Jensen, Jianliang Xu

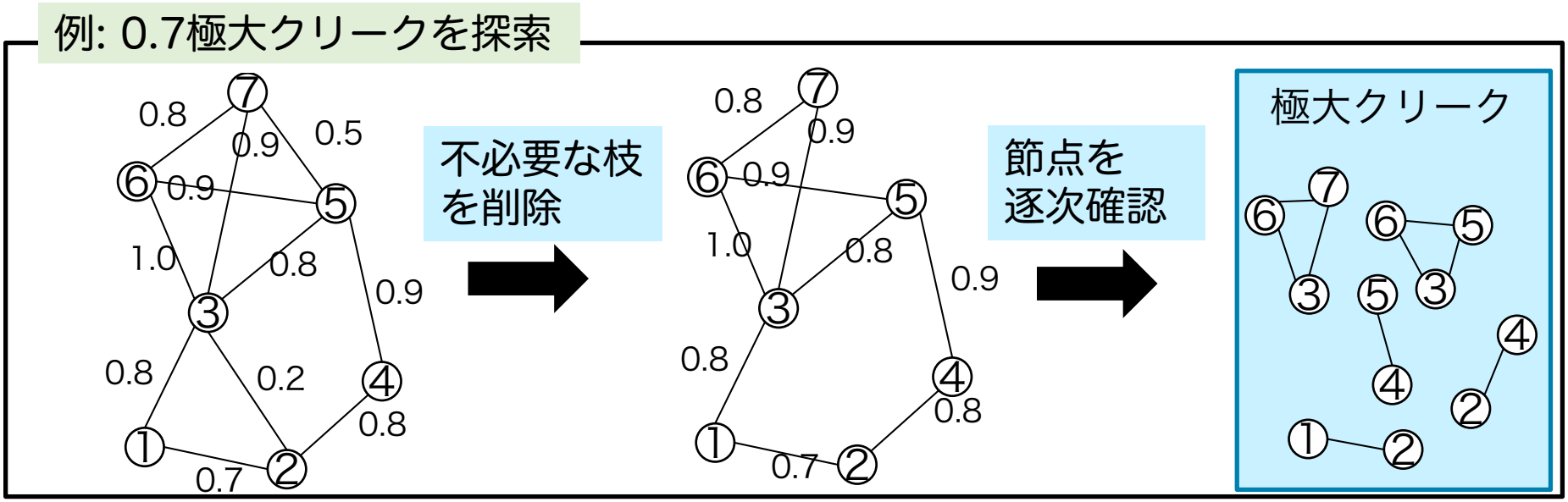
Mining Maximal Cliques from an Uncertain Graph

- 曖昧なグラフにおける極大クリーク探索問題
  - ソーシャルネットワークの解析などに重要
- 課題：
  - 確率を考慮した極大クリークが未定義および探索法が必要
- 貢献：
  - α極大クリークの定義
  - α極大クリークの最大数 $\binom{n}{\lfloor \frac{n}{2} \rfloor}$ を証明（割愛）
  - MULE(Maximal Uncertain cLique Enumeration)の提案



α極大クリークとは？

- 存在確率がα以上の極大クリーク
  - 存在確率α以下の枝は不必要
  - クリークに追加する枝
    - クリークの存在確率がα以上となる枝
    - クリークになってる全ての節点と隣接してる枝

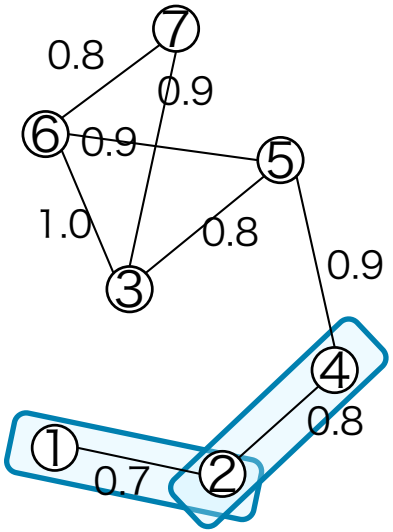


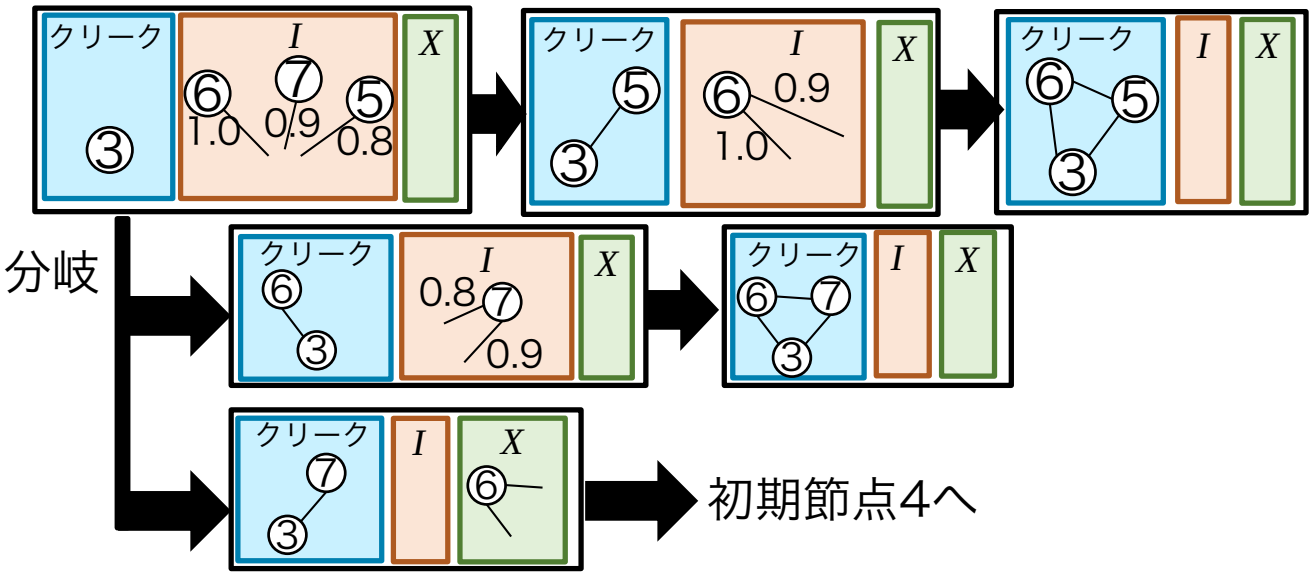
重複なく，無駄なくクリークを発見する方法は？

探索アルゴリズム

- **MULE**(Maximal Uncertain cLique Enumeration)
  - 全ての節点から再帰的に実行
  - インクリメンタルに節点を追加
    - 節点のIDの昇順に追加。IDが小さい節点は無視。
  - 追加可能な節点Iと極大クリークになった隣接節点Xを管理
    - IとXが$NULL$なら，極大クリーク完成

例: 0.7極大クリークを探索：節点1と2の処理が終わった後．節点3から実行





Temporal Spatial-Keyword Top-k Publish/Subscribe

- **TaSKクエリ** (Temporal Spatial-Keyword Top-k Subscription)
  - 継続的にtop-kのオブジェクトを出力
    - オブジェクトのスコアは，キーワード，地理，時間から算出

例："food poisoning vomiting"に関する自宅近くの最近のツイートを知りたい。

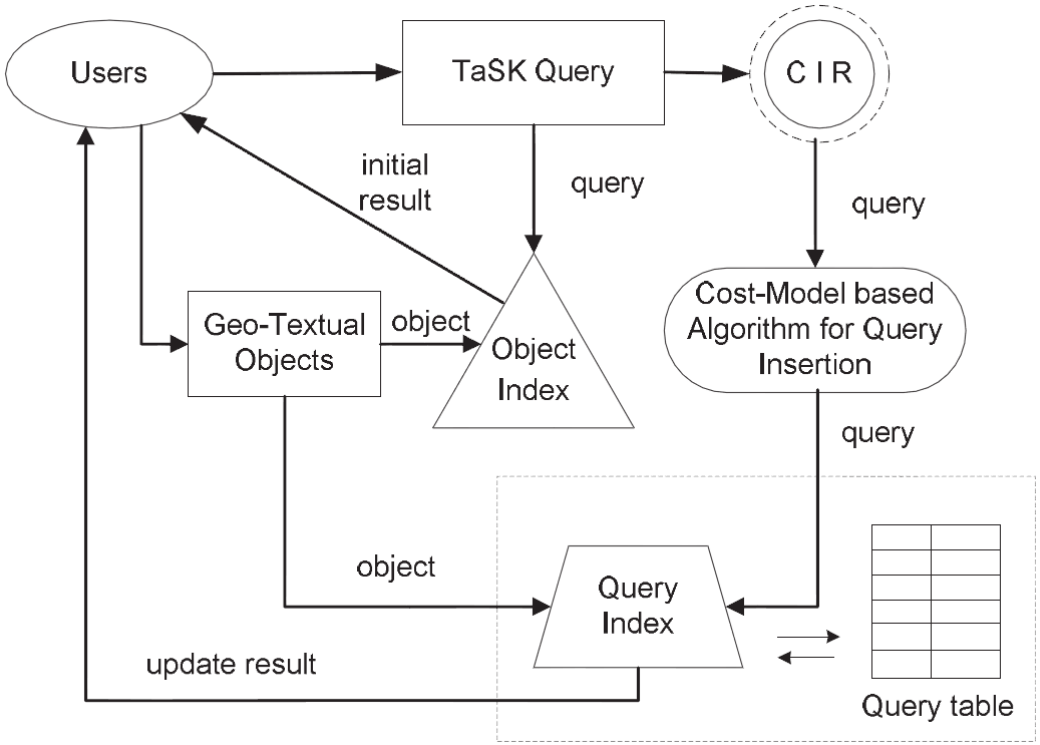
- 課題：
  - クエリの数やデータの発生頻度が大きいと計算量が大きい

新しいオブジェクトがきたときの効率的な処理が重要

- 貢献：
  - TaSKクエリの提案
  - インデックス構造や計算アルゴリズムの提案
  - 実データを用いた実験（割愛）

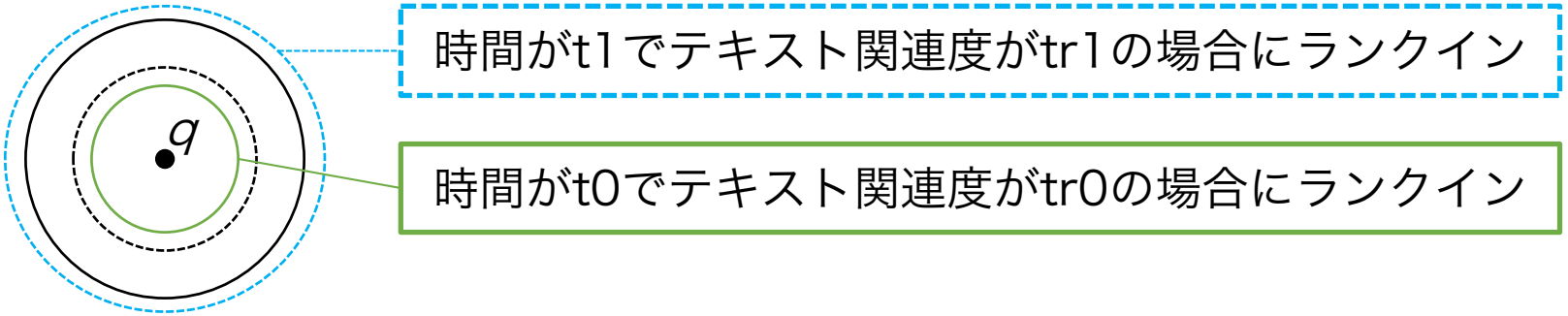
システムの概要

- **CIR** (Conditional Influence Region)
- クエリインデックス
- クエリ挿入アルゴリズム
- オブジェクト処理



キーアイディア①

- CIR：top-kに影響する地理的範囲
  - CIRをもとに，閾値計算やコスト予測を行う。



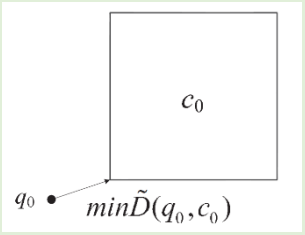
- クエリインデックス
  - クエリをセルに格納（Quad-treeを採用）
    - クエリは全ての領域をカバーするように複数のセルに割当られる。
  - セル内に転置インデックス
    - 転置インデック内のクエリをブロック単位に分割
    - ブロック内のクエリからブロック毎に閾値を設定
      - セルとクエリ間の最小距離，最小スコアから計算

キーアイディア②

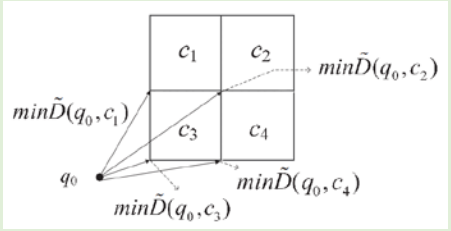
- クエリ挿入：
  - クエリは全ての領域をカバーするようにセルを関連付け
  - どの粒度のセルに格納すると効率的かコスト計算

どっちに格納すべき？

(A)



(B)



- オブジェクト処理
  - オブジェクトの位置に対応するセルを探索
  - ブロック毎のキーワード集合とオブジェクトのテキスト関連度を計算
    - 閾値以上であれば，ブロック内のクエリとの関連度を計算

TaSKクエリの効率化のために，いろいろやってます。