【SIGMOD2013勉強会】

Session 24: Road Networks and Trajectories

担当：董・早矢仕（名古屋大学）

Calibrating Trajectory Data for Similarity-based Analysis

- Han Su, Kai Zheng, Haozhou Wang (UQ), Jiamin Huang(Nanjing U), Xiaofang Zhou (UQ)
- Problem:軌跡校正（trajectory calibration）
  - 異種の軌跡データセットを，サンプリング戦略が統合される軌跡に転換
    - 様々なサンプリング戦略による異種軌跡
      - サンプリング手法：距離，時間，予測など
      - パラメータ：頻度，バッテリーなどの要素
- Contribution: an anchor-based calibration system
  - 参照システム（reference system, offline）
    - 同種のanchor点の集合：$\{a_1,\ldots,a_{10}\}$
      - anchor点：空間上の固定位置（交差点など）
  - 軌跡校正（trajectory calibration, on/offline）
    - 整列：軌跡のサンプル点をanchor点にmap
    - 補足：anchor点を追加

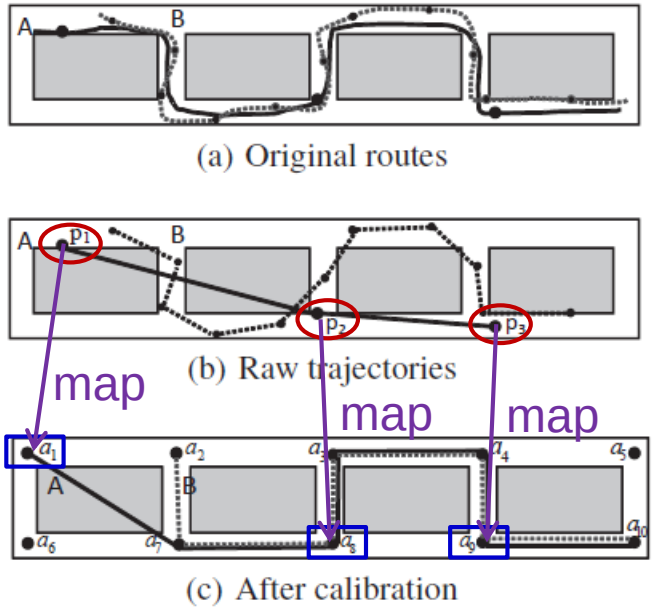


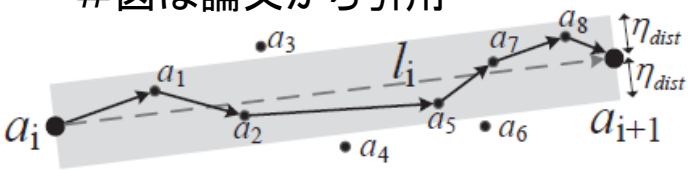
Figure 1: Motivation of calibration

＃図は論文から引用

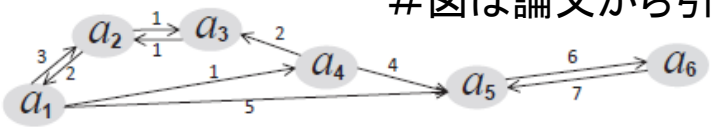
Calibrating Trajectory Data for Similarity-based Analysis

- 参照システム
  - anchor点の種類
    - 空間に基づくanchor (GC)：グリッドの重心，Realm法[13]
    - データに基づくanchor (AS)：過去軌跡のサンプル点
    - POIに基づくanchor (POI)：Point of Interest (POI, 店など), DBSCAN[9]
    - 特徴に基づくanchor (TP)：交差点，[6]
- 軌跡校正
  - 幾何学に基づく校正(G)
    - 整列：サンプル点→最近傍anchor点
    - 補足：連続したサンプル点の線の近くのanchor点，変化角度<π/2
  - モデルに基づく校正(M)：Hidden Markov Model (HMM)
    - anchor遷移確率行列を前計算
    - 整列：確率の高い整列を選ぶ
    - 補足：連続したサンプル点間の可能なパスを列挙し確率の高いパスを選ぶ

#図は論文から引用



#図は論文から引用



Calibrating Trajectory Data for Similarity-based Analysis

- 評価実験
  - 軌跡データ：タクシー, 11208, sample rate<10s;
    re-sample 50s, 100s, 150s
  - 比較対象：SP, GC, AS, POI+G, POI+M, TP+G, TP+M
  - 類似度尺度：Euclidean Distance, DTW, LCSS, EDR
  - 実験結果（下の図から）
    - POI, TP > GC, AS > SP
    - TP+M > POI+M > POI+G, TP+G

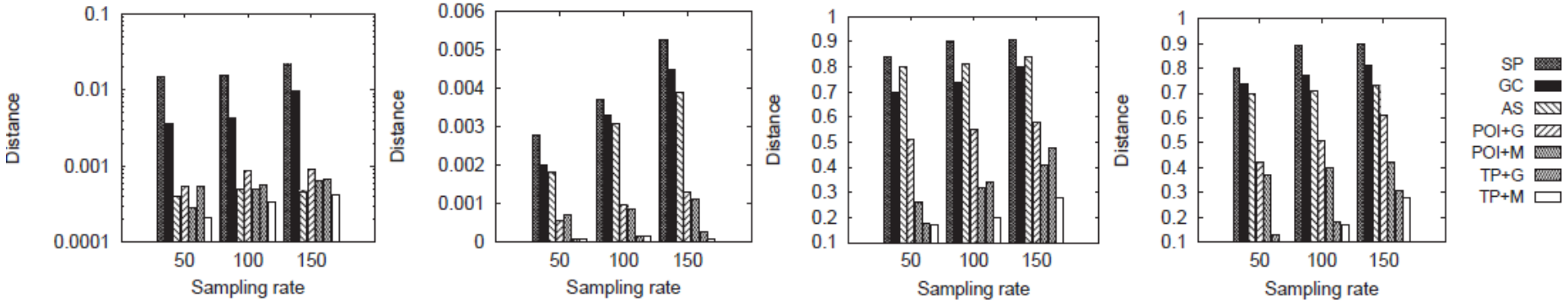
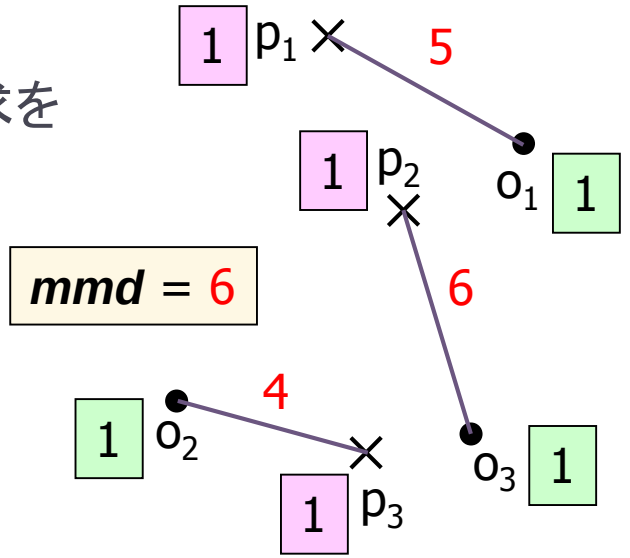


Figure 8: Distance between the trajectories referring to the same original route

On Optimal Worst-Case Matching

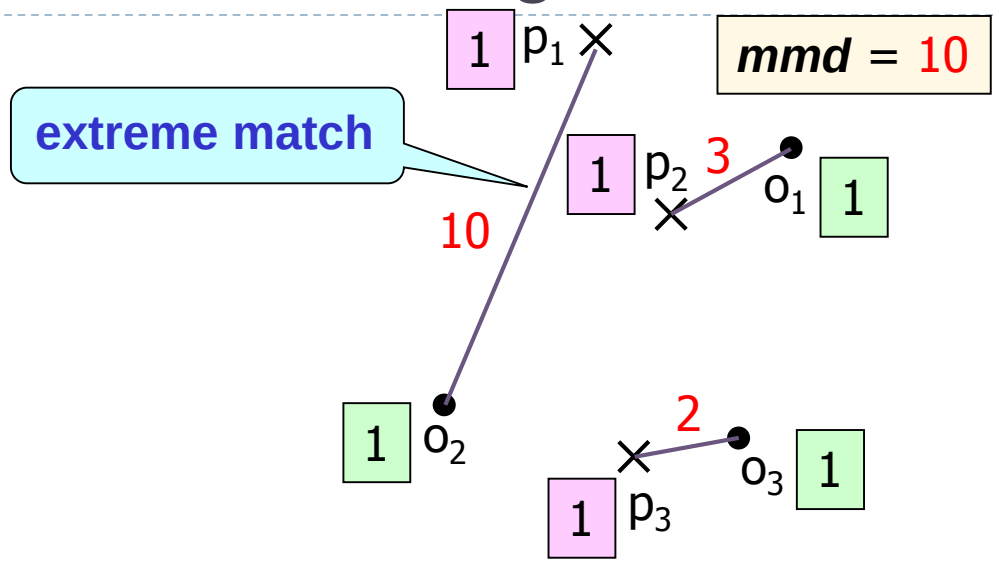
- Cheng Long, Raymond Chi-Wing Wong (HKUST), Philip S. Yu (UIC), Minhao Jiang (HKUST)
- Problem: Spatial Matching for Minimizing Maximum matching distance (SPM-MM)
  - サービス提供者の許容量と顧客の要求を考慮した割当手法
    - 例:
      - サービス提供者(病院):P = {p_1, p_2, p_3}
      - 顧客(住宅):O = {o_1, o_2, o_3}
      - 許容量と要求は**任意**の整数(1で説明)
      - Maximum matching distance (***mmd***) = 6
    - 目標:***mmd***を最小化する

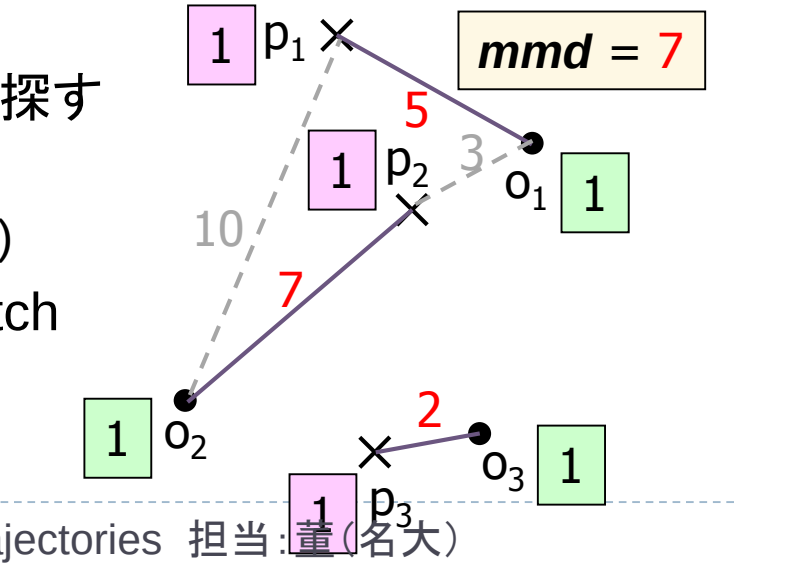


- Contribution: two algorithms for SPM-MM
  - *Threshold-Adapt*:既存手法に基づく,拡張性がない
  - ***Swap-Chain***:効率良い,拡張性がある

On Optimal Worst-Case Matching

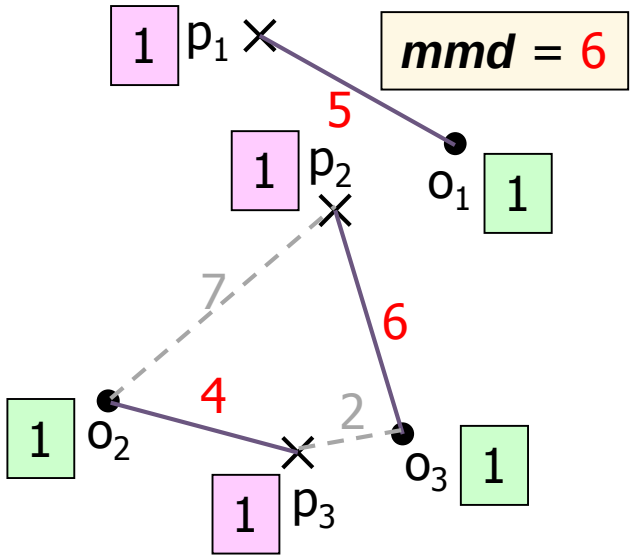
- ***Swap-Chain***
  - Step 1：割当初期化
    - 戦略
      - Fair assignment [25]
      - Globally optimized [22]
      - Randomなど
  - Step 2：割当調整
    - Extreme match o_2p_1をbreak
    - o_2のchainを範囲問合せ(r=***mmd***)で探す
      - o_2 p_2—o_1 p_1：**短い**
      - o_2 p_3—o_3 p_2—o_1 p_1 (o_3p_1=11>10)
    - 元のchainをbreak, 新なchainをmatch
      - o_2 p_2 o_1 p_1
      - ***mmd***: 10 -> 7





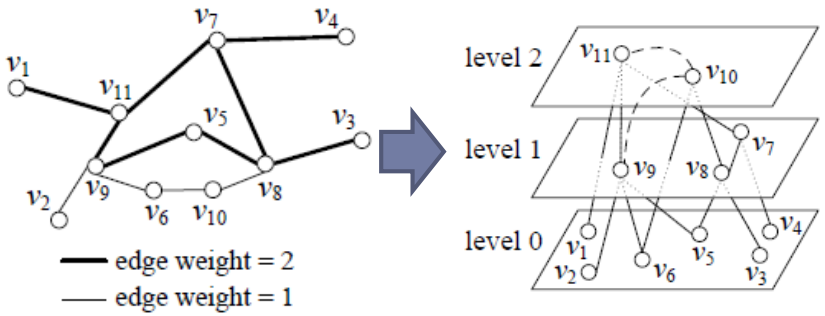
On Optimal Worst-Case Matching

- ***Swap-Chain***
  - Step 3 : 反復
    - Extreme match: o_2p_2
    - Chain: o_2 p_3 — o_3 p_2
    - ***mmd***: 7 -> 6
  - 出力結果 : ***mmd*** = 6
  - 許容量と要求が任意の整数
    - 元のchainをbreakする前後，顧客の要求の変化と提供者の許容量の変化に基づき，新なchainをmatch
  - 時間計算量の理論的な分析
  - 距離尺度がnon-metric又はnon-spatialの場合への拡張
- 実験による時空間的な効率性の検証
  - 消防署と人口集中区域の実データと人工データ



Shortest Path and Distance Queries on Road Networks : Towards Bridging Theory and Practice

- Andy Diwen Zhu[1], Hui Ma[1], Xiaokui Xiao[1], Siqiang Luo[2], Youze Tang[1], Shuigeng Zhou[2] ([1]Nanyang Technological University, [2]Fudan University)
- 問題設定
  - ロードネットワークにおける最短経路の問合せとグラフ上の最短距離の問合せ
- 従来手法の問題点
  - 実際のパフォーマンスがベストな手法はヒューリスティック[Geisberger et al, 2008]
    - 理論上の計算量・空間計算量は$O(n^2)$
  - 計算量が良い手法では前処理のコストとオーバヘッドが大きい[Samet et al, 2008]
    - 巨大なロードネットワークを対象としたアプリケーションでの利用が困難
- 提案手法
  - Arterial Hierarchyの提案
    - ロードネットワークのノードを階層的に扱う
    - 階層関係から探索するノードの枝刈を行う
  - 計算量
    - 最短距離問合せ:$O(h\log h)$ h: AHの深さ
    - 最短経路問合せ:$O(\mathrm{k}+\mathrm{h}\log h)$ k：最短経路のノード数



図は論文より引用

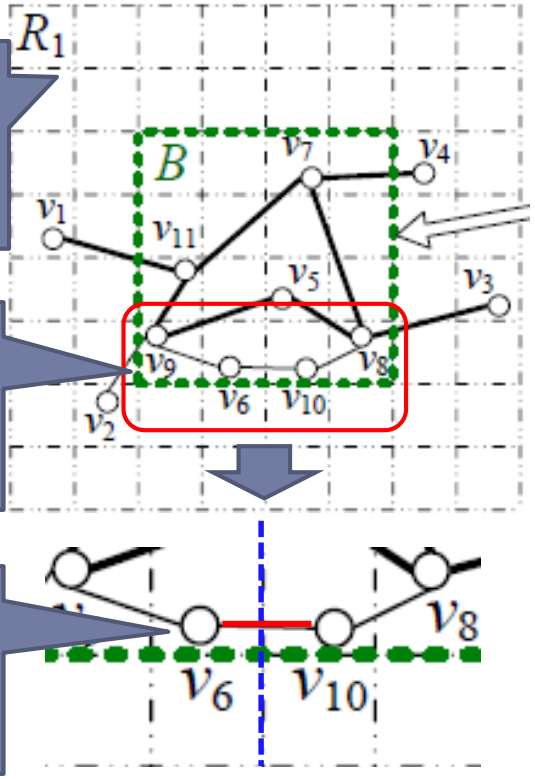
First-Cut(FC)：ベースとなる索引構造

レベルの決め方

４×４のグリッドを
２×２で再帰的に分割

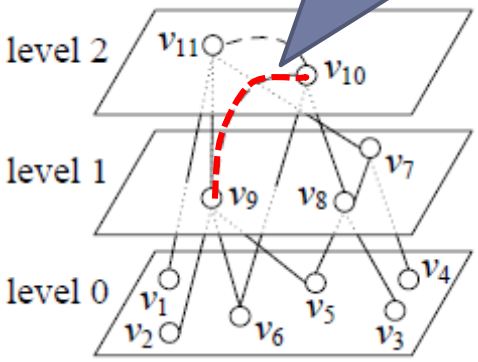
パスの両端が
グリッドの対極に位置

二等分線と交差する
エッジ→arterial edge



グリッドの粒度を大きくして次のレベルの構築

パスの両端より
レベルが低い場合
ショートカットを作成



問合せ処理
双方向からダイクストラを適用

低レベルへの
遷移の禁止

３×３のセル内の
ノードへ遷移

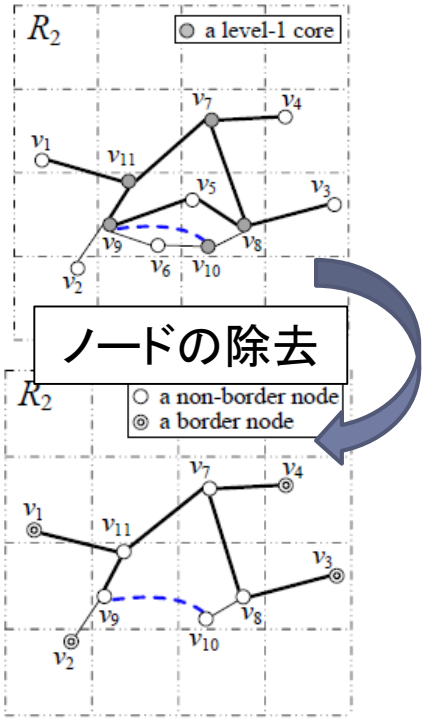
図は論文より引用，加筆

Arterial Hierarchy（AH）

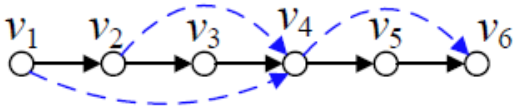
- FCの前処理のコストを減らすように改良
- FCとの違い

1. レベル決定手順
   - FC:全ノードのレベルを決定　→　全ショートカットの作成
   - AH：各階層毎に「ノードレベルを決定 → ショートカットの作成」
     - 低レベルのノードを除去したグラフを逐次作成
     - 各階層で処理するグラフを小さくするので処理が速い
2. ショートカットの作成
   - ショートカットを”2-hop path”で保持
     - 最短経路問合せの処理が速い

- その他，細かい制限の追加（詳細は論文を参照）
  - レベル決定手順の違いによる変更



ショートカット



2-hop path

$$< v_1, v_4 > \rightarrow < v_1, v_2, v_4 >$$
$$< v_2, v_4 > \rightarrow < v_2, v_3, v_4 >$$
$$< v_4, v_6 > \rightarrow < v_4, v_5, v_6 >$$

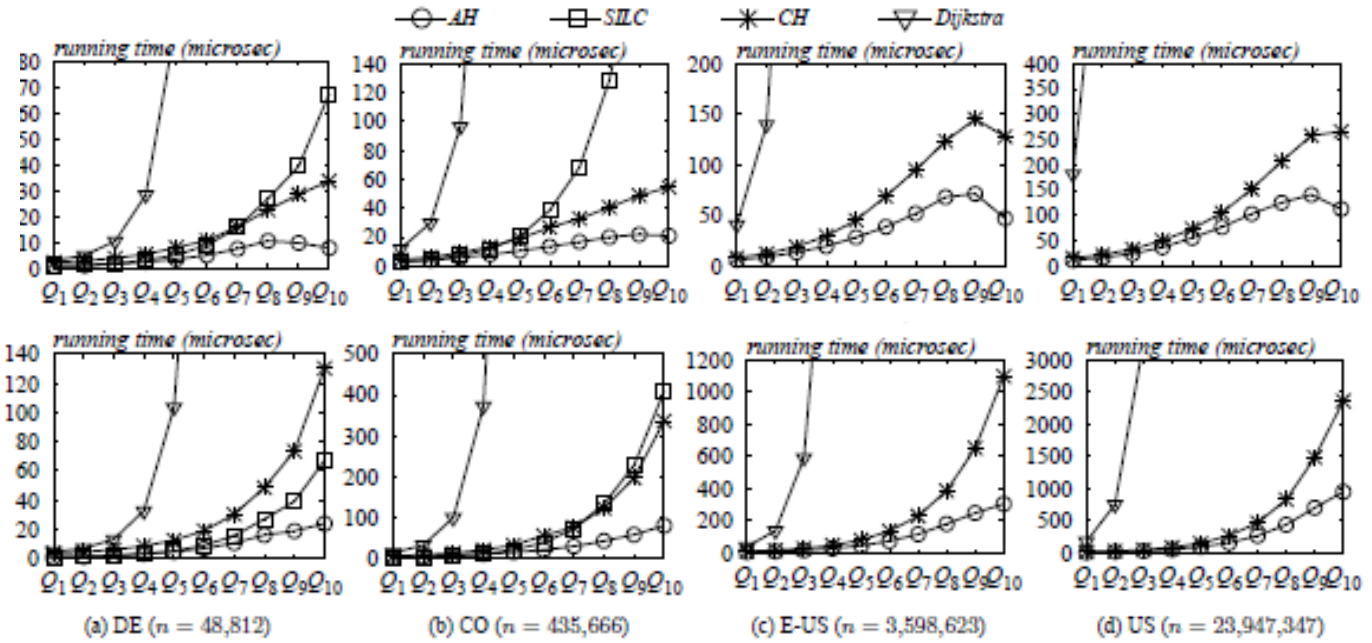
図は論文より引用，加筆

評価結果

- distance query，shortest path query共に提案手法が高速

distance query

shortest path query



- AH：提案手法
- SILC：ワーストケースを高速に処理
- CH：ヒューリスティックでベストな手法

図は論文より引用