

【ICDE 2013勉強会】

Session 7: Trajectory Databases

担当: 杉浦健人, 姜仁河, 趙セイ(名大)

Some figures are copied from ICDE 2013 proceedings.

Towards Efficient Search for Activity Trajectory

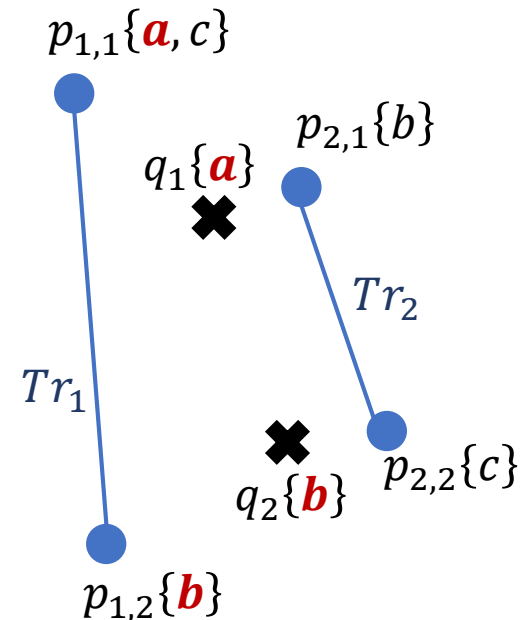
Kai Zheng, Shuo Shang, Nicholas Jing Yuan, Yi Yang
(U. Queensland / Aalborg U. / Microsoft Research Asia /
Carnegie Mellon U.)

問題設定

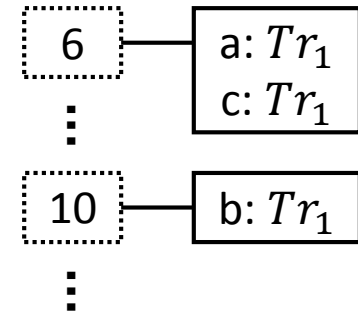
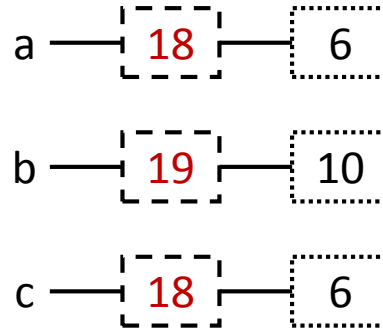
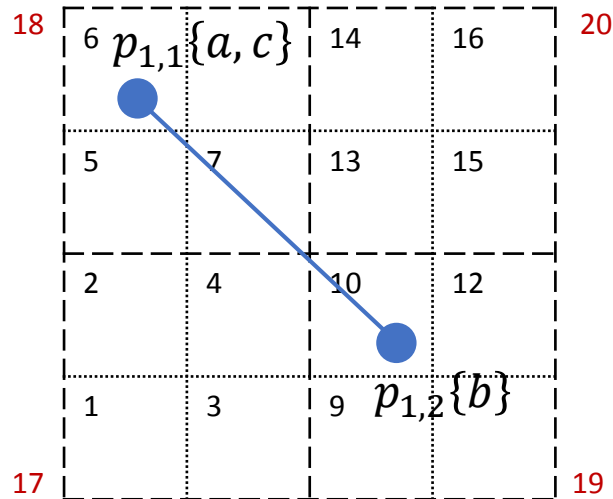
- 活動軌跡の集合Dとquery pointの集合Q
- 活動を考慮して適当な活動軌跡を選択
e.g. Tr_2 よりも Tr_1 の方が距離は遠いが、
ユーザの求める活動をしている

Activity Trajectory Similarity Query (ATSQ)

- 活動が一致し距離の近い活動軌跡を選択
- 活動が一致しているかの確認と距離の計算を
どうしたら効率よく同時に行えるか？



Grid index for Activity Trajectory (GAT) の利用



活動軌跡はそのままでは扱いづらいため、
空間を格子状に分割した階層インデックスを利用

- 「活動→地点, 地点→活動 (軌跡)」の変換が容易
- データ量が小さいためメモリ上で保持可能
- 階層構造を利用して効率的に枝刈り可能

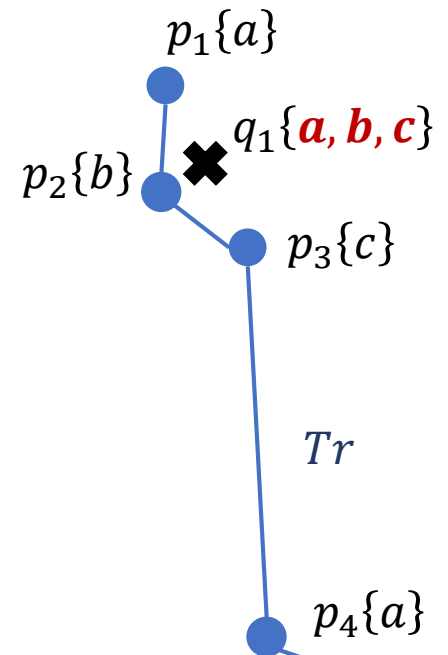
活動一致の確認と距離計算の同時実行

各query point $q_i \in Q$ の活動に一致し
かつ距離の近い活動軌跡 Tr を見つける問題に分割して考える
→ Tr と各 q_i の距離を合計して Tr と Q の距離とする

Tr と q_i の距離計算

$p_j \in Tr$ を距離の昇順にソートし,
順に結合して活動が q_i と一致するものを探す

- $p_1, p_2, p_3\{a, b, c\}$ は $q_1\{a, b, c\}$ と活動が一致
→ p_1, p_2, p_3 と q_1 の距離を最小距離として保持
- p_4 以降の点までの距離は最小距離より大きい
→ p_4 以降に解は存在しない → 距離計算終了



On Discovery of Gathering Patterns from Trajectories

▶ Kai Zheng, Yu Zheng, Nicholas Jing Yuan, Shuo Shang
(U. Queensland / Microsoft Research Asia)

▶ 背景:

▶ グループイベントから軌跡パターンを発見 → **Gathering**

• 例: **交通渋滞、抗議、パレード**



▶ 従来手法:

▶ **グループ**の概念に基づいてパターンを発見

• グループの定義によって、其々**Flock**, **Convoy**, **Swarm**手法がある

▶ 条件: **グループは終始同じ個体**を含む

▶ 問題点:

▶ 実際には、個体がグループイベントに**出入り**することが多い



グループ中で同じ個体を維持しないGathering手法を提案

Crowd, Participator, Gatheringの定義

▶ Crowdの定義

$\{C_1, C_2, C_3, C_4, C_5\}$

▶ k_c 時間内の密度ベースのクラスタのシーケンス

• 例: $\langle C_1, C_2, C_4 \rangle, \langle C_1, C_3, C_4 \rangle \rightarrow$ **Crowd** $k_c=3$

$\langle C_1, C_2, C_5 \rangle, \langle C_1, C_3, C_5 \rangle \rightarrow$ **✗** C_5 と $C_2(C_3)$ の距離が遠い

▶ Participatorの定義

▶ k_c 時間内に少なくとも k_p 個クラスタに現れるオブジェクト

• 例: $O_2 \rightarrow$ **Participator**, $O_1 \rightarrow$ **✗** $k_c=3, k_p=3$

▶ Gatheringの定義

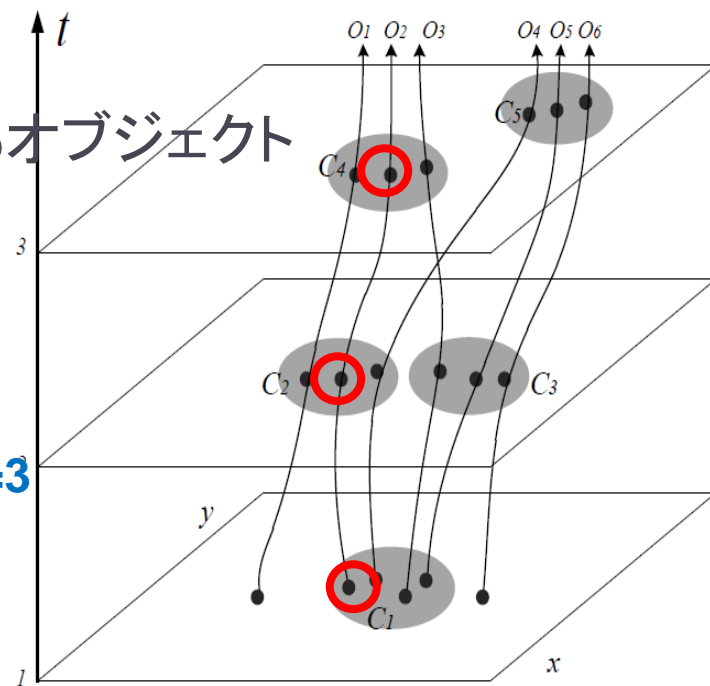
▶ m_p 個のParticipatorを含むCrowd

• 例: $\langle C_1, C_2, C_4 \rangle \rightarrow$ **Gathering** $k_c=3, k_p=2, m_p=3$

C_1, C_2, C_4 は O_1, O_2, O_3, O_4 Participator中の三つを含む

• $\langle C_1, C_3, C_4 \rangle \rightarrow$ **Crowd**, $\langle C_1, C_3, C_4 \rangle \rightarrow$ **✗ Gathering**

ただ C_1 は O_2, O_3, O_5 Participator中の三つを含む



Gathering発見のプロセス

▶ Crowdを発見: Cluster(入力) → Closed Crowd(出力)

super-crowdにならないcrowd

▶ 成長 (growth) 型のアルゴリズムの提案

- 短いCrowdシーケンスを長いCrowdシーケンスに拡張
- 拡張できない時、Closed Crowdになるか、Crowdにならないかを判別

						Time	Crowd	Closed Crowd
						1	<C ₁₁ >	
						2	<C ₁₁ , C ₂₁ >, <C ₁₁ , C ₂₂ >	
t1	t2	t3	t4	t5	t6	3	<C ₁₁ , C ₂₁ , C ₃₁ >, <C ₁₁ , C ₂₁ , C ₃₂ >, <C ₁₁ , C ₂₂ , C ₃₂ >	
					C ₆₁	4	<C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ >	
		C ₃₁	C ₄₁	C ₅₁		5	<C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ , C ₅₁ >, <C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ , C ₅₂ >, <C ₅₃ >	
C ₁₁	C ₂₁			C ₅₂		6	<C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ , C ₅₁ , C ₆₁ >, <C ₅₃ , C ₆₂ >	<C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ , C ₅₂ >
	C ₂₂	C ₃₂		C ₅₃		7		<C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ , C ₅₂ >, <C ₁₁ , C ₂₁ , C ₃₁ , C ₄₁ , C ₅₁ , C ₆₁ >
					C ₆₂			

▶ Gatheringを検出: Closed Crowd(入力) → Gathering(出力)

▶ Test-And-Divideアルゴリズムの提案

- 十分なParticipatorを含めていないClusterを削除 **Test段階**
- 全部のCrowdを部分列に分割し、再帰的に部分列からGatheringを検出 **Divide段階**

Destination Prediction by Sub-Trajectory Synthesis and Privacy Protection Against Such Prediction

▶ Andy Yuan Xue et al. (University of Melbourne, MSRA)

▶ 研究の動機

▶ 目的地の予測により、観光地や目的地に関する広告などをユーザに推薦

▶ 問題点

▶ Ziebart et al.¹による既存の目的地予測技術

▶ 例: クエリ軌跡 $\{l_1, l_4, l_5\}$ に対する予測目的地なし

▶ 原因: data sparsityの問題

▶ 目的地予測はプライバシー問題を生み出しうる

▶ 提案手法

▶ Sub-trajectory Synthesis (SubSyn) アルゴリズム

▶ SubSynアルゴリズムにおけるプライバシー保護

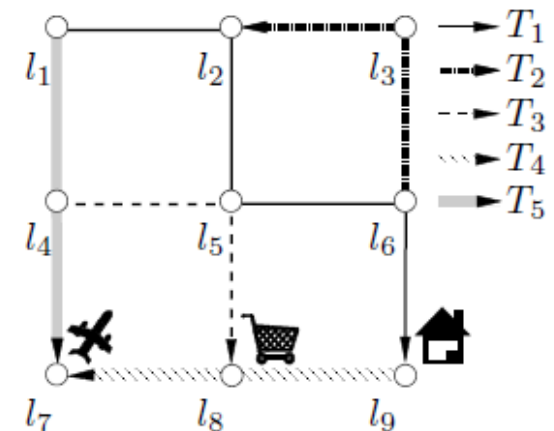


Fig. 1. An example of destination prediction

Destination prediction based on Sub-Trajectory synthesis

▶ SubSynアルゴリズム

- ▶ 考え方: 履歴軌跡をそれぞれ長さ2の部分軌跡に分解し, 合成軌跡として結合

▶ 目的地予測確率の計算

- ▶ 部分軌跡の一階マルコフモデルを構築 (オフライン)

- ▶ グリッドグラフによる表現

- ▶ p_{ij} : ノード n_i から n_j までの遷移確率

- ▶ 遷移行列 M (隣接ノード間の遷移確率) を計算

- ▶ 任意のノード間の完全遷移行列 M^T を M から計算 (オフライン)

- ▶ 問合せ軌跡とマッチした合成軌跡の目的地確率を M^T により計算 (オンライン)

- ▶ 確率が高い順に k 個の目的地を返す

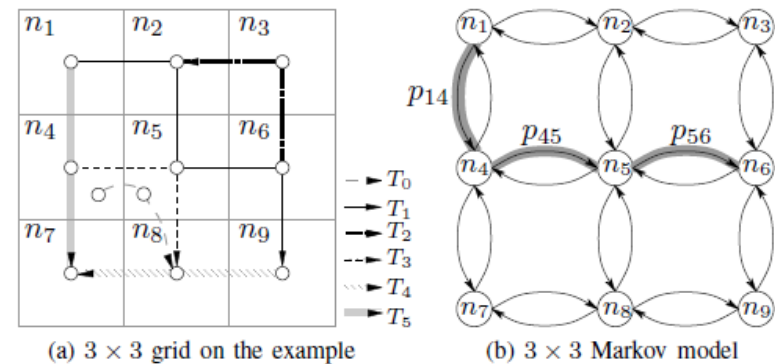


Fig. 2. Grid graph representation and Markov model

$$M = \begin{pmatrix} 0 & p_{12} & 0 & p_{14} & 0 & 0 & 0 & 0 & 0 \\ p_{21} & 0 & p_{23} & 0 & p_{25} & 0 & 0 & 0 & 0 \\ 0 & p_{32} & 0 & 0 & 0 & p_{36} & 0 & 0 & 0 \\ p_{41} & 0 & 0 & 0 & p_{45} & 0 & p_{47} & 0 & 0 \\ 0 & p_{52} & 0 & p_{54} & 0 & p_{56} & 0 & p_{58} & 0 \\ 0 & 0 & p_{63} & 0 & p_{65} & 0 & 0 & 0 & p_{69} \\ 0 & 0 & 0 & p_{74} & 0 & 0 & 0 & p_{78} & 0 \\ 0 & 0 & 0 & 0 & p_{85} & 0 & p_{87} & 0 & p_{89} \\ 0 & 0 & 0 & 0 & 0 & p_{96} & 0 & p_{98} & 0 \end{pmatrix}$$

Privacy Protection Against Prediction Based on SubSyn Algorithm

- ▶ SubSynアルゴリズムによるプライバシー問題
 - ▶ 予測された目的地はユーザの**センシティブな位置情報**である可能性がある
 - ▶ 例: **住宅位置**など
- ▶ **保護手法**: センシティブな位置を予測できる軌跡情報を削除

ユーザの**センシティブな位置情報**を保護するために、**削除される軌跡データが最小となるプラン**を求める

▶ **網羅的な生成方法**

- ▶ 全ての削除案を**列挙**し、ユーザの要求に満足できる案を選択
- ▶ **非効率**

コストが高い

▶ **End-Points生成方法**

- ▶ **始点と終点だけを削除する**
- ▶ 理論的裏付け: 一階マルコフモデルの特性により、任意の予測目的地の確率がただ開始ノードと現在ノードのみに依存