【ICDE 2015勉強会】

Session 1: Data Integration
Session 21: Trajectories

担当：趙　セイ（名大）

Some figures are copied from ICDE 2015 proceedings.

Cleaning Uncertain Data with a Noisy Crowd

- Chen Jason Zhang, Lei Chen, Yongxin Tong , Zheng Liu (HKUST, China)
- 目的：クラウドソーシングを用いて，データの質を改善
- Uncertainty Model: X-relation model
  - 確率的に独立な**x-タプル**セット
  - x-タプル：確率付きタプルの集合
    - $T_1\{t_1(0.4), t_2(0.3), t_3(0.2), t_4(0.1)\}$ ）

例，条件付き確率：
Pr(House no=51|Area=East)=1)

Location address of T_1: 51A Hayward East New York

Xid	Tid	House no	Area	City	$Pr(t_i)$
T_1	t_1	51	Hayward	New York	0.4
T_1	t_2	51A	Hayward East	New York	0.3
T_1	t_3	51A	Hayward East	York	0.2
T_1	t_4	51	East	York	0.1

Table1: running example-uncertain data generated by an information extraction tool

Crowdsourcing Model

- 質の尺度
  - Negative value of the *Shannon entropy*: $\sum_p plog_p$
    - $Q(T_1) = 0.4 * \log 0.4 + 0.3 * \log 0.3 + 0.2 * \log 0.2 + 0.1 * \log 0.1 = -0.55$
  - データの**不確実性が高い**ほど，**質が低い**
- 質を改善
  - 手段：HITs(Human Intelligent Tasks)を公開
  - 例，“I: City=New York” (HIT6) is true
    - $\Pr(t_1|I) = \frac{\Pr(t_1)\Pr(I|t_1)}{\Pr(I)=\Pr(t_1)+\Pr(t_2)} = 0.57$
    - $Q(T_1)$ = -0.3 > -0.55
- 問題の定義
  - 予算（答え数）B以内にHITsを選んで，Q(T)を最大化する
- チャレンジ
  - 答えの誤り率
  - HITsの関連性

質問形式：Is the *c.att c* ?
c.att：属性　*c*: セルの値

答え：yes or no

ID	HIT content	ground truth $Pr(h_c)$	crowdsourced answer $Pr(A_c)$
HIT1	Is the House no 51?	y(0.5) n(0.5)	y(0.5) n (0.5)
HIT2	Is the House no 51A?	y(0.5) n (0.5)	y(0.5) n(0.5)
HIT3	Is the Area Hayward?	y(0.4) n (0.6)	y(0.45) n (0.55)
HIT4	Is the Area Hayward East?	y(0.5) n (0.5)	y(0.5) n (0.5)
HIT5	Is the Area East?	y(0.1) n (0.9)	y(0.3) n (0.7)
HIT6	Is the City New York?	y(0.7) n (0.3)	y(0.6) n (0.4)
HIT7	Is the City York?	y(0.1) n(0.9)	y(0.3) n (0.7)

TABLE II. RUNNING EXAMPLE - CANDIDATE HITS FOR CLEANING WITH CROWD, AND DISTRIBUTIONS OF THE GROUND TRUTH AND CROWDSOURCED ANSWER (CROWD ACCURACY = 0.75)

NP-hardness

提案手法

- (Theorem4.1) 目標関数： $S_h := \arg\max_{S_h} H(A_{S_h})$
  - $H(A_{S_h})$：答えのセットA_{S_h}のエントロピー（答えのランダム性の尺度）
- 基本的なアイデア：
  $H(A_{S_h})$の上・下限値に基づき，枝刈りと推定を行う
  - 上・下限値の計算（Sec. IVを参照）
- 近似アルゴリズム（small k）
  - インスタンス-レベル枝刈り
    - HITs: h_0, h_1, 次の式が成立すると，h_1を枝刈りする

      $$H(A_{(S_{k-1}\cup\{h_0\})}).lb \geq H(A_{(S_{k-1}\cup\{h_1\})}).ub$$
  - アルゴリズム-レベル枝刈り
    - 上限値の**降順**でHITを選らんで，枝刈りを行う
- 発見的なアルゴリズム（large k）
  - 下限値と上限値の**平均値**を用いて，$H(A_{(S_{k-1}\cup\{h_1\})})$の値を推定

Approximate Keyword Search in Semantic Trajectory Database

- Bolong. Zh, Nichalas J.Y., Kai. Zh, Xing X., Shazia. S., Xiaofang Zh. (UQ, MSRA)
- 空間キーワード検索
  - ユーザの位置とキーワードにより，関連するPOIsを見つける
  - 例：旅行プランなど
- 既存研究
  - 空間ウェブオブジェクト（Cong et al.）
  - ATSQ(Activity trajectory similarity query) (Zheng et al.)
    - クエリのアクティビティをカバーし，最短的最小なマッチ距離をもたらす
- 問題点
  1. ATSQ: **正確な**キーワード検索しか支持しない
  2. **ユーザに指定**された位置により，クエリを処理
- 目的
  - **意味的**軌跡データベースにおける**近似**キーワード検索を行う

Problem Statement

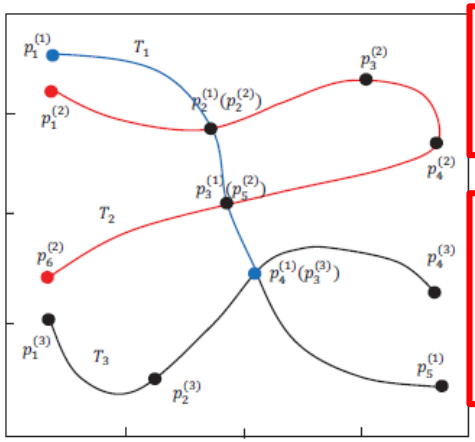
- Spatio-textual utility function

$$D_{st}(Q,T) = \min_{T[s,e] \subseteq T} \{D_{ft}(Q,T[s,e])\} \quad (1)$$

テキスト距離　　移動距離

  - Fixed length textual distance: $D_{ft}(Q,T[s,e]) = \alpha \cdot D_{td}(Q,T[s,e]) + (1-\alpha) \cdot D_{tr}(T[s,e])$
- Approximate keyword query of semantic trajectory（AKQST）
  - 意味的軌跡セットD, クエリ Q=$\{q_1, q_2, \ldots, q_{|Q|}\}$が与えられ，
    spatio-textual utility functionの最小値を持つ
    k個の異なる軌跡を返す

    移動距離が短い，かつクエリに類似なキーワードを含める意味的な軌跡
  - 実行例（図1）
    - Q={“Restaurant”,“Theatre”}
    - Dst(T1[1,4])<Dst(T2[1,6])
    - Top-1 AKQST:T1（青い線）



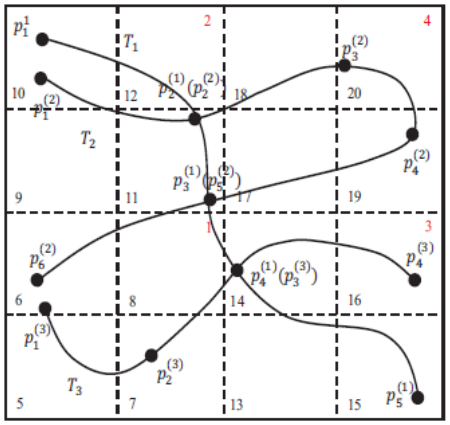
T_1	$p_1^{(1)}$	w_1^1:Restaurant, w_1^2:Club, w_1^3:Shop
	$p_2^{(1)}$	w_2^1:Cafe
	$p_3^{(1)}$	w_3^1:Park
	$p_4^{(1)}$	w_4^1:Theater, w_4^2:Pub
	$p_5^{(1)}$	w_5^1:Market, w_5^2:Station
T_2	$p_1^{(2)}$	w_1^1:Restaurant, w_1^2:Gym
	$p_2^{(2)}$	w_2^1:Cafe
	$p_3^{(2)}$	w_3^1:School, w_3^2:Pizza, w_3^3:Spa
	$p_4^{(2)}$	w_4^1:Bakery, w_4^2:Hotel, w_4^3:Mall
	$p_5^{(2)}$	w_5^1:Park
	$p_6^{(2)}$	w_6^1:Theatre, w_6^2:Bar
...	...	...

図1

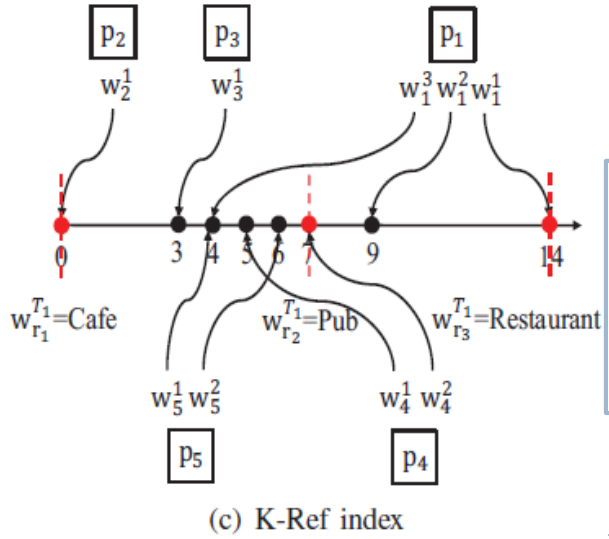
Hybrid Index Structure

Grid-Keyword index (GiKi)

- スペース分割
  - d-Grid：四分グリッド（図(a)）
- SQ-Tree Index（図(b)）:意味的軌跡データを格納
- K-Ref Index（図(c)）：各軌跡に参考キーワード（reference keywords）セット R(T)を格納



(a) Grid partition of space

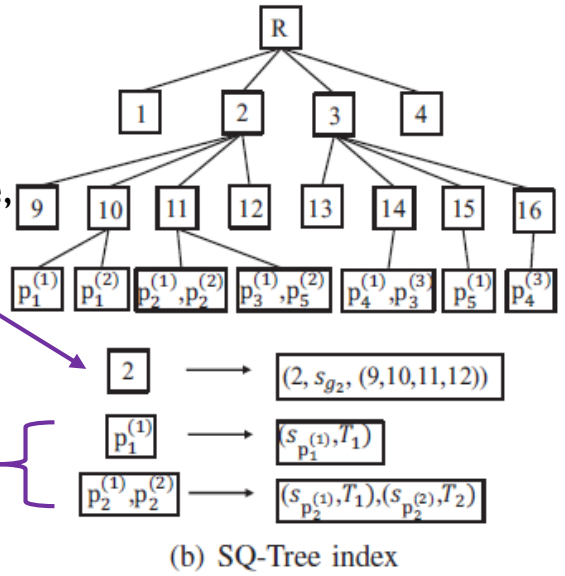


(c) K-Ref index

（実行例）
軌跡T1の参考キーワード：”Cafe”, “Pub”,“Restaurant”

内部ノード
（ID, keyword signature, Sub entries）

葉ノード
（キーワード付き空間オブジェクト）



(b) SQ-Tree index

Approximate Keyword Query Processing

- 検索手法
  - ステップ1：候補軌跡セットを抽出
    - SQ-Treeにおいて，
      生成したsignatureがクエリとの類似度が高いグリッドが優先
  - ステップ2：候補における軌跡の下限値$D_{st}(Q,T)_L$を計算
    - K-Ref indexに基づき，R(T)とQのedit distanceの下限値を計算
    - **動的計画法**（DPAアルゴリズム）
  - ステップ3：各軌跡のSpatio-textual utility functionを計算
    - $D_{st}(Q,T)_k$（k番目の$D_{st}(Q,T)$の最小値）
    - $D_{st}(Q,T)_L$（残りの軌跡の下限値）
- 候補検証（下限値による枝刈り）
  - 次の式に満たす場合，処理が終了

$$D_{st}(Q,T)_k < D_{st}\ (Q,T)_L$$