【ICDE 2015勉強会】

Session 24-3: Query Processing 3

担当：胡（名大）

Some figures are copied from ICDE 2015 proceedings.

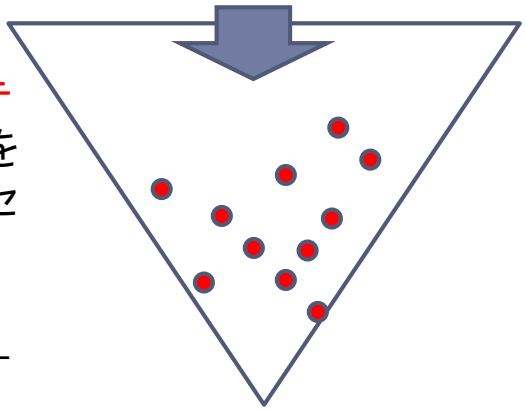
AP-Tree: Efficiently Support Continuous Spatial-Keyword Queries Over Stream

Xiang Wang(University of New South Wales), Ying Zhang(University of Technology Sydney), Wenjie Zhang, Xuemin Lin, Wei Wang(University of New South Wales)

- ストリームと書いているが、実際はSpatial-keyword問題の変種
- 僕の理解では, クエリとデータセットの構成を逆にしているのは区別のところ

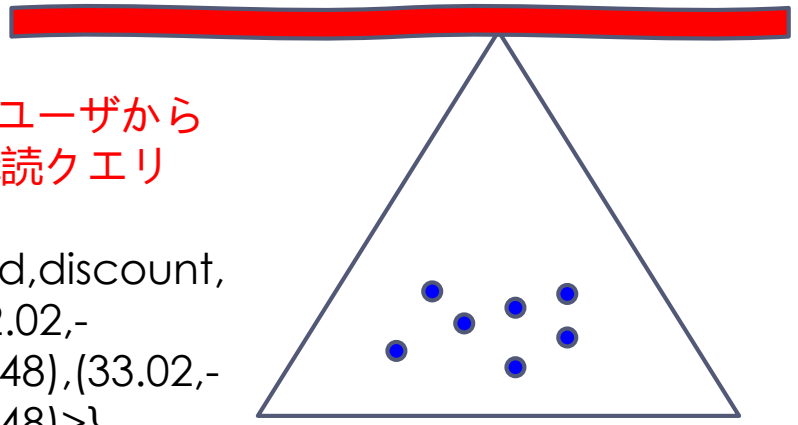
一括クエリQで,結果を引き出す

● は空間とテキスト情報を含むデータセットe.g. <Century Park, 32.02,-112.48>



Spatial-Keyword問題

ストリームからのobjectを一個一個処理する



●はユーザからの購読クエリ e.g. {ipad,discount, <(32.02,-112.48),(33.02,-111.48)>}

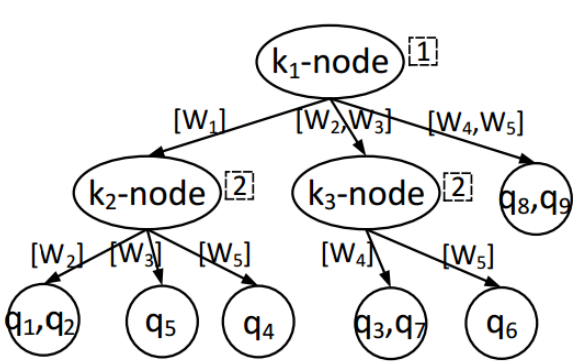
本論文streamカッコカリ問題

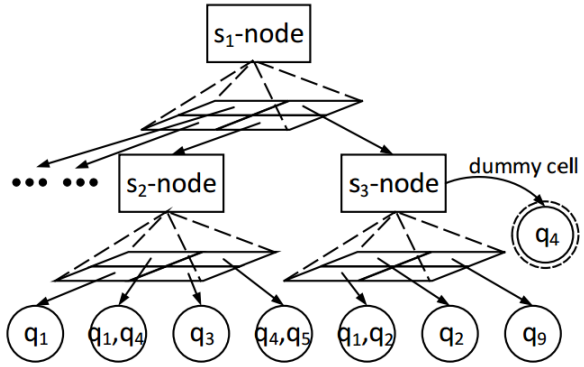
- 典型的なSpatial-Keyword問題: 電子地図上の店検索
- 典型的なstreamカッコカリ問題: 店の電子招待券を特定興味の客への配り

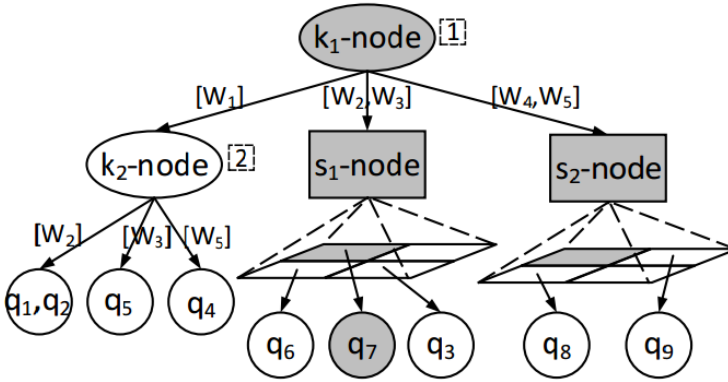
Core-technique

- 普通のSpatial-keywordの問題と同じく, Spatial側のデータとKeyword側のデータはまったく異なるデータなので, 索引を作った時に区別した処理は求められる
- 実際の場合では, filtering-refinementの手法によって, 単なるspatial側またはkeyword側のデータに基づいて索引を作るのもそんなに悪くもない
- 単なる索引手法といえば
  - Spatial系はR-tree, Quad-treeなど
  - Keyword系はTrie, Inverted Listなど

本論文ではQuad-treeとTrie(変種)の組合せ







Keyword側のTrie索引　Spatial側のQuad-tree索引　TrieとQuad-treeの組合せ

Cost Model

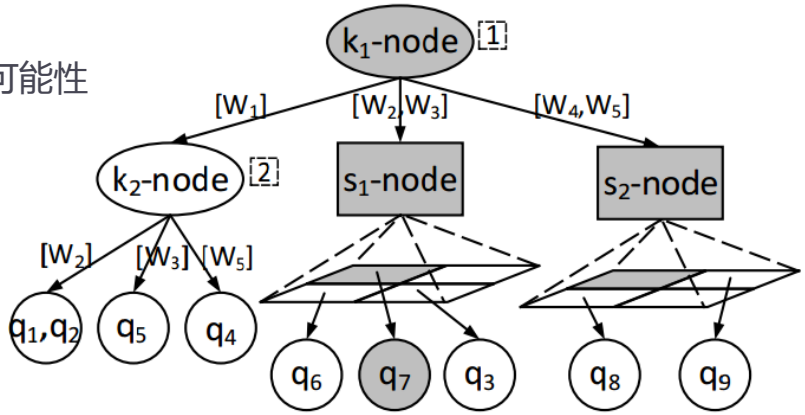
- 組合せといっても，どちらはkeyword-nodeどちらはspatial-nodeを挿入すべきか効率的に最適化になれるか
- ここで挿入nodeを決めるCost modelを導入
- Keyword-nodeの場合を想定し，全部のTrie(1層だけ)での構築案を探し尽くして，1枝のcostは，
  - 当枝以下のデータ保有数 * その枝を辿り着く可能性
- Spatial-nodeの場合のcostは，
  - 当グリッド内のデータ保有数 * そのグリッドに入る可能性
- Cost 比較

  if(Keyword.cost < Spatial.cost)

  Keyword-nodeを挿入

  else

  Spatial-nodeを挿入



TrieとQuad-treeの組合せ

【ICDE 2015勉強会】

Session 15-4: Spatial Query Processing

担当：胡（名大）

Some figures are copied from ICDE 2015 proceedings.

A Location-Aware Publish/Subscribe Framework for Parameterized Spatio-Textual Subscriptions

Huiqi Hu, Yiqun Liu, Guoliang Li, Jianhua feng(Tsinghua U) Kian-Lee Tan(National University of Singapore)

なぜかこの論文をここに置くか

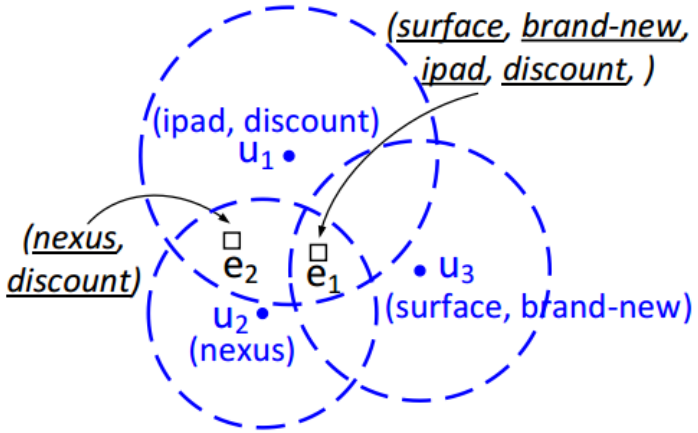
ストリームの話は毛頭ないが, 実はpublish/Subscribe問題はストリームと同じ

典型的なstreamカッコカリ問題: 店の電子招待券を特定興味の客への配り

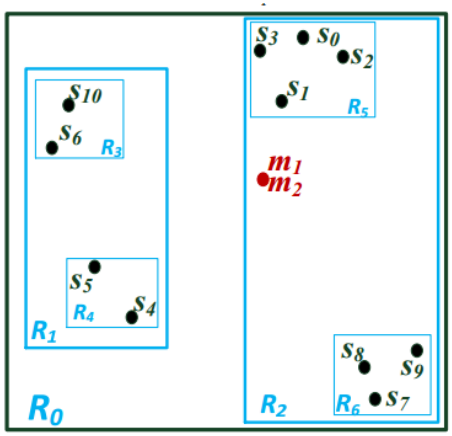
Publish側　　Subscribe側

もちろんSpatial-Keyword問題です(ここでSpatio-Textualと表記)

すこし変わったのは:



AP-tree:Boolean Region Query



本論文 :Top-k Knn Query

Core Technique

- Top-k 問題だから, 類似度からのランキングは鉄板
- 前にも話したが, spatialとtextual両種類のデータがあるため, 類似度の計算も二股
- 計算した二つの類似度を重みつけてランキング (重み: δ)
  - 総合類似度 = δ * Textual類似度 + (1- δ) * Spatial類似度

Spatial-oriented prefix-filter(実はtextualのprefix)

- 上の式により, 総合類似度の閾値を確定した上で, textualとSpatial類似度の関係式は確定できる
- これを踏まえて, Spatial類似度は小なり1のは固有性質なので, textualのある閾値も確定できる(不等式変換により)
- 分かりやすく言うと, textual部分は大間違いがあるから, いずれSpatialの部分はどんなに近くなっても(1に至るまで), そのデータと類似するのは絶対無理だという
- Textualはどの程度に間違ってる？ このprefixは測定できる

例: Original data <ipad, discount, iphone, mac> ----> Prefix <ipad, discount>

Query1: <iphone, mac> はprefixと共通部分はない, 類似が絶対不可能

Query2: <ipad, discount, iphone> 共通部分あり, でも必ず類似成立でもない

ちなみに, そのprefixを提案したのはvldb2012のSEAL(Ju Fan.etc), ここのはオリジナルではない

Core Technique(2)

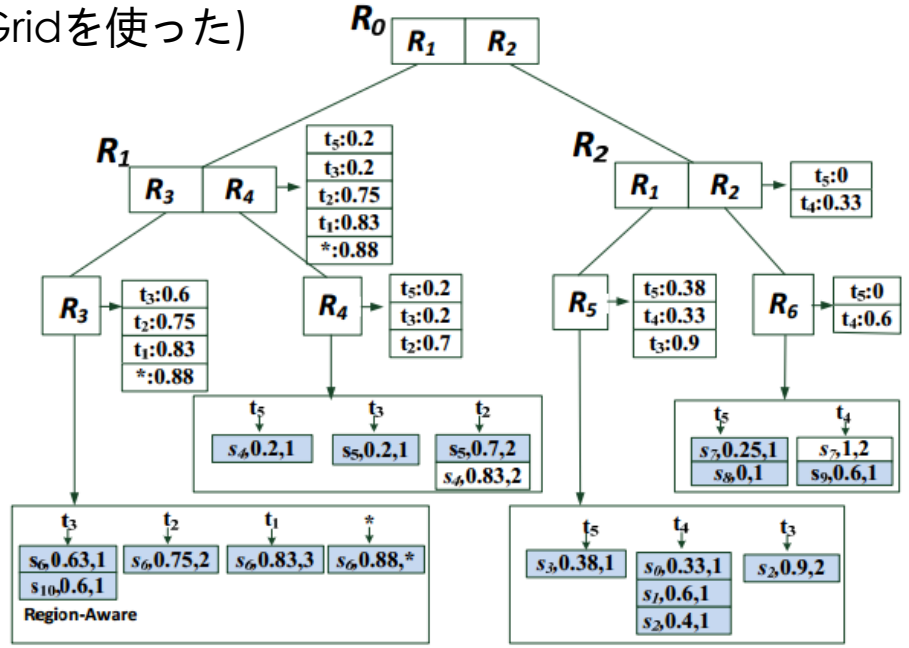
- 前にも話したが, spatialとtextual両種類のデータがあるから, 索引の手法も別々

本論文ではR-treeとInverted List(変種)の組合せ

R-treeを用いると, regionごとにInverted Listを作ることが出来る

クエリとRegionの類似度を計算すると, よりtightな閾値は得られる

(VLDB2012のSEALはR-treeの代わりにGridを使った)



R-TreeとInverted listの組合せ