【VLDB 2013勉強会】

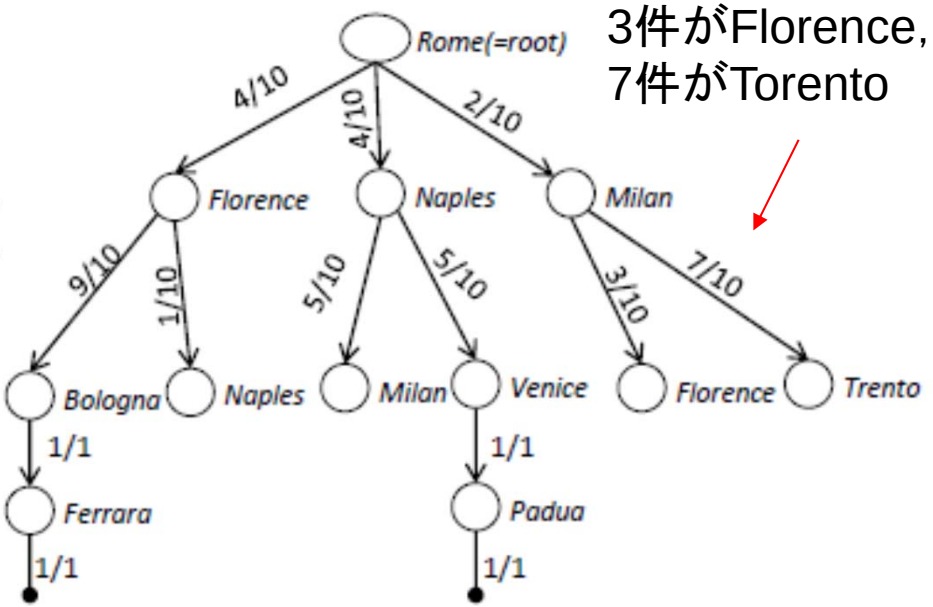
Session 5: Social and Crowd

担当：石川佳治，簗井美咲（名大）

Some figures are copied from VLDB 2013 proceedings.

Answering Planning Queries with the Crowd

- Haim Kaplan, Iila Lotosh, Tova Milo, Slava Novgorodov (Tel-Aviv U.)
- プラン（plan）：順序付けされた選択肢のシーケンス
- 研究の目的：クラウドソーシングを用いてプラン問合せに応える
- CrowdPlanr：旅行プランを求めるシステム（応用事例）
- 例：Rome→Naples→Milan というプランのスコアは 4/10 * 5 / 10 = 0.2

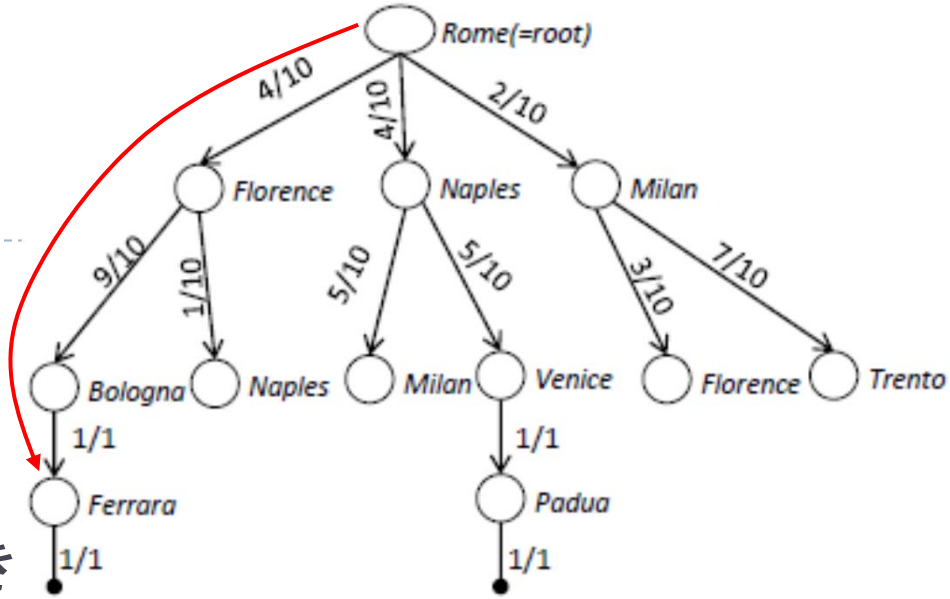


問題設定

$k = 1$が
メイン

- トップk問合せ：スコア順にk個のプランを返す
  - 例：ローマからの旅行プランのうちベストなk件を求めよ
- 基本的な考え方
  - 先に示した木構造を，クラウドソーシングをもとに構築・展開
  - パラメータN: 各ノードで最大何回まで人間に問合せ可能か
- 問題点：完全な木構造の構築には膨大な問合せ必要
- アイデア1：近似的なトップk（パラメータεで条件緩和）
  - トップkに含まれるプランpは，トップkに含まれない任意のパスp'に対し，$\text{score}(p) \geq \text{score}(p') - \varepsilon$を満たす
- アイデア2：今後 人がどう回答してもトップkが変わらないなら，処理を途中で終了可

手法の概要



- 例（$N = 10$, $k = 1$, $\varepsilon = 0.01$）
  - 図のパスが最大スコアだがまだ確定しない
  - Bolognaであと9回問合せできその結果がすべてTorentoなら，Florence→Bologna→Torentoが逆転可能：まだ停止できない
- 本研究の貢献
  - 木構造の展開処理の停止条件を示す
  - 「次にどのノードについて尋ねるか」のヒューリスティクスを示す：最もスコアが高くなりそうな（トップ1）のプランで，これまでの質問が回数が最小のもの
  - アルゴリズムの最適性：理論的分析（研究のポイント）
  - 実験による評価

Query Optimization over Crowdsourced Data

- Hyunjung Park, Jennifer Widom (Stanford U.)
- スタンフォード大で開発しているDecoの最適化について
- 例：国と都市に関するDB

Country(country, [language], [capital])
City(city, country, [population])

・アンカー属性：
　エンティティを識別
・依存属性

- Decoの特徴
  - 過去に取得したデータをストア：問合せ処理では，新たにデータをどの程度取得するかがポイント ⇒ 費用の問題
  - Fetchルール：属性値をクラウドソーシングで取得
    - [Country] ∅ ⇒ country　　// 国名を質問
    - [Country] country ⇒ capital　// 与えられた国名に対し首都を質問

Decoのフレームワーク

- Decoの特徴(続き)
  - Resolutionルール:値の不整合を解消
    - [Country] ∅ → country : dupElim // 同じ国名は削除
    - [Country] country → language : majority-of-3
      // 3件以上は多数決をとるユーザ定義関数
    - Fetch-Resolve-Joinシーケンスによる問合せ処理
  - 結果タプルの最小数(MINTUPLES)を指定

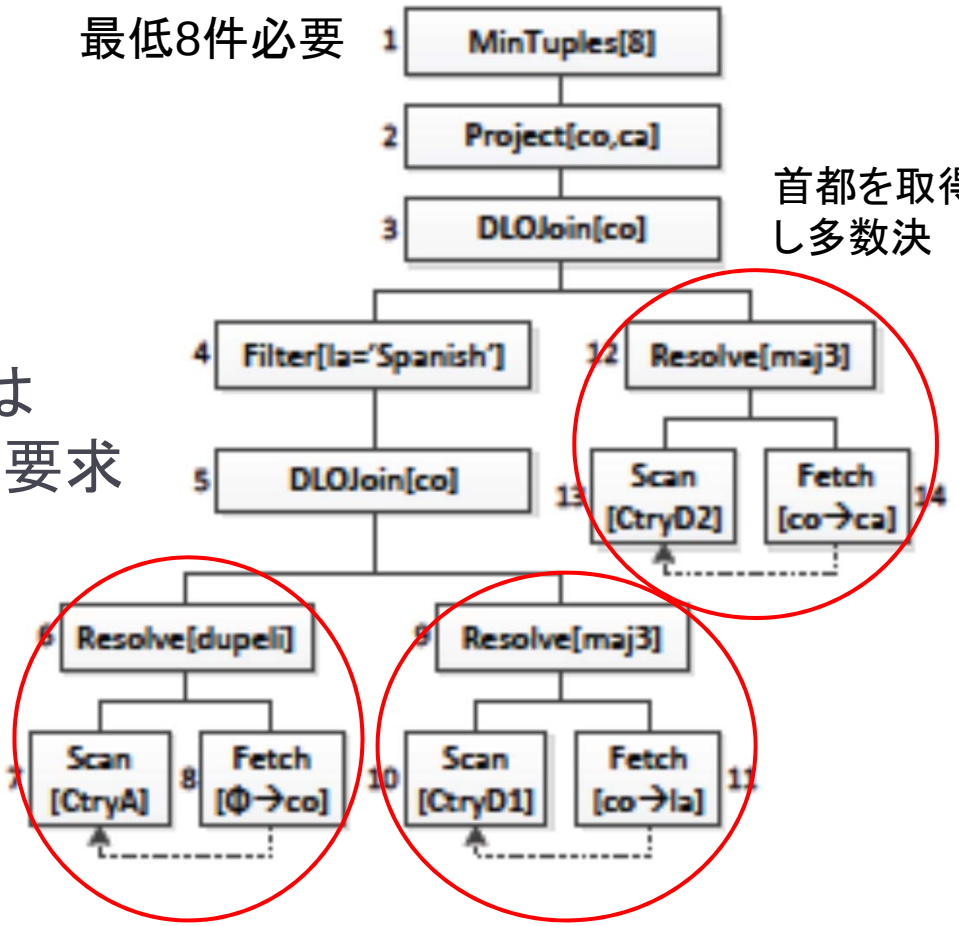
```
SELECT country, capital FROM Country
WHERE language = 'Spanish' MINTUPLES 8
```

スペイン語を話す国の情報を8件以上求めよ

  - 2フェーズの問合せ処理
    - 実体化(materialization):既存データによる問合せ処理
    - 増加(accretion):フェーズ1で不十分なら新たにデータ取得

最適化処理

- 問合せプランの例
  - トップダウンに実行：ルートは8件になるまで次のタプルを要求
- コストファクター：費用
  - 各Fetch処理の費用×呼び出し回数
- 最適化のポイント
  - 選択率：各Resolve処理でタプルがどの程度減るか
  - カーディナリティ（結果のタプル数）の推定
    - 研究としてはここがポイントの一つ
  - Reverse fetchルール（例：language ⇒ country）の活用



Counting with the Crowd

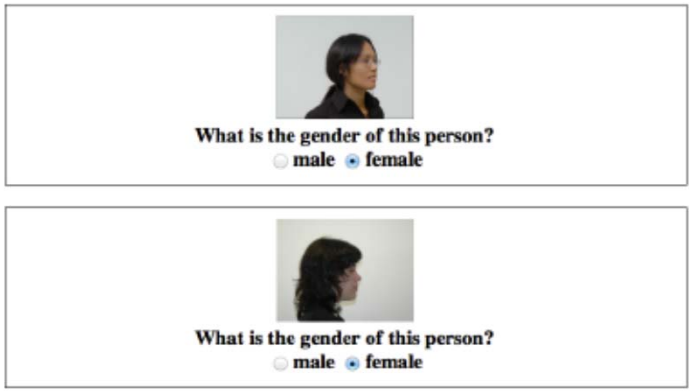
- Adam Marcus, David Karger, Samuel Madden, Robert Miller, Sewoong Oh (MIT)
- 目的:クラウドソーシングを用いた選択率(selectivity)の推定
- 例:画像データベースへの問合せ

```
SELECT id, name
FROM photos
WHERE gender(picture) = 'male'
   AND hairColor(picture) = 'red'
```

それぞれの選択率は?

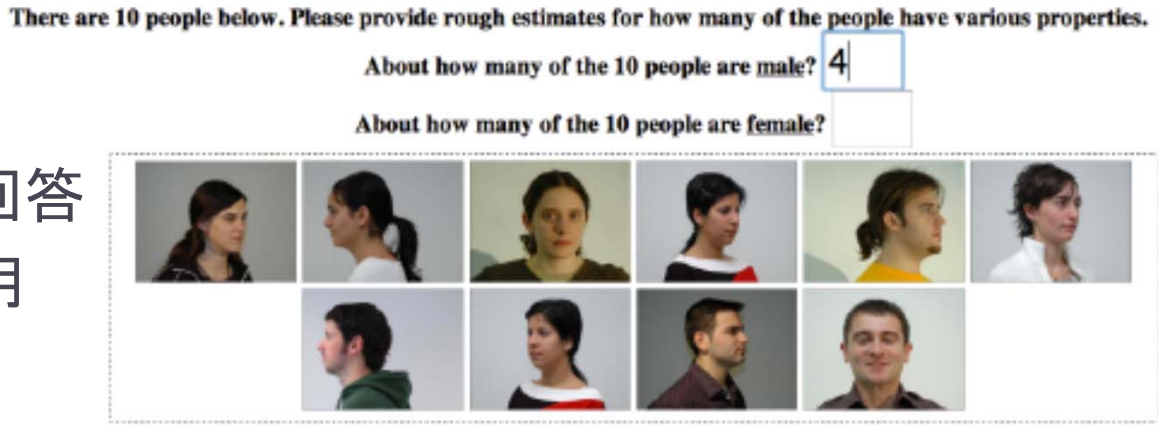
  - 対象の画像データベースに男性,赤い髪の人がどの程度いるかは人による判断が必要
    - クラウドソーシングを活用

アイデア



ラベルベース

- 通常はラベルベースの処理
  - それぞれの画像について ワーカがラベルを付与
- 本研究：カウントベースの処理
  - 何件が該当するか回答
  - 人の把握能力を利用
- 処理のアプローチ
  - 基本的にはワーカのスコアを平均
  - スパマーの検出：かけ離れたスコアを与えたワーカを排除
  - “gold standard”（値が分かっている基準データ）の利用：サンプルに混在

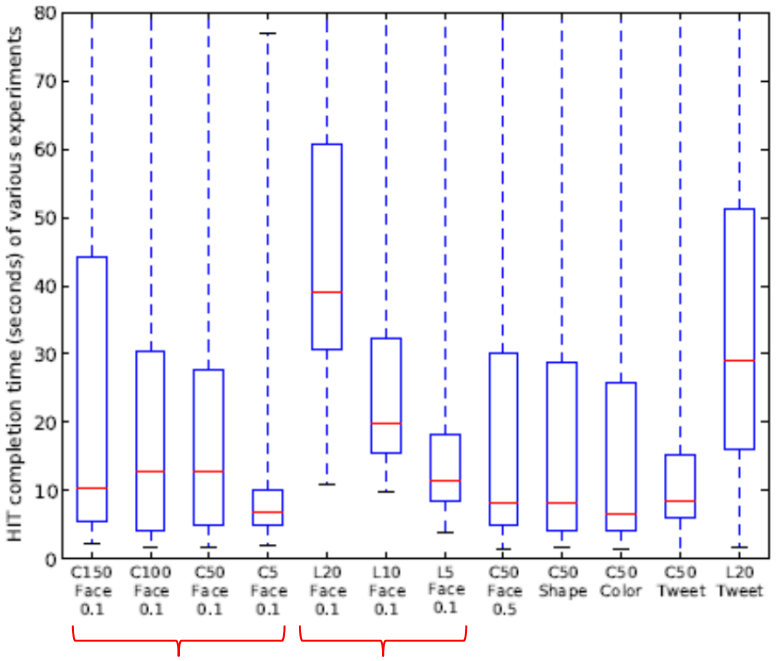


カウントベース

実験

- Amazon Mechanical Turkを利用
- データセット：顔画像の性別認識（メイン），Twitterの分類など
- 実験のサマリ
  - ラベルベース：より正確だが，コスト大
  - カウントベース：スパマー検出すれば，ラベルベースよりやや悪い程度．提示数大の場合，収束早い
  - カウントベースにはタスクの向き不向きあり
  - ワーカの回答時間について興味深い結果（右図）

150個の画像を提示したときの平均回答時間は，50または100個の場合より早い



顔画像（カウント）
150, 100, 50, 5

顔画像（ラベル）