CareDB: A Context and Preference-Aware Location-Based Database

Justin J. Levandoski
Mohamed F. Mokbel
Mohamed E. Khalefa
Department of Computer Science and Engineering - University of Minnesota

Overview

Need For Context and Preference in Databases

Consider a mobile location-based business finding application:
- Existing applications return the K closest restaurants
- Consider five closest restaurants for dinner
  - Restaurant 1: Hour and a half wait
  - Restaurant 2: Does not meet my diet
  - Restaurant 3: Too expensive
  - Restaurant 4: Closed for remodeling
  - Restaurant 5: 30 minute drive due to heavy traffic

The five restaurants are NOT useful as database systems are detached from:
1. Personal preferences (dietary restrictions, budget)
2. Extra contextual data (time of day, traffic, waiting times)

CareDB Architecture

Efficient Query Processing

CareDB Functionality

Extensible Preference Query Processor

Extensible preference query processing support provided through FlexPref extensible preference query processing framework:
1. Define two macros and three functions in separate file outside CareDB/FlexPref
   - $\text{IsPreferredObject}(O, \text{PreferenceSet})$
     - False if $O$ can never be a preferred object
     - True otherwise
   - $\text{DefinePreference}(O)$
     - Default score assigned to object
   - $\text{PairwiseCompare}(O_1, O_2)$
2. Compile into CareDB/FlexPref using command:
   - $\text{DefinePreference} \text{MyPref}.$ with $\text{MyPref}$

Efficient Generic Preference Join

- Naive join approach
- Join all data
- Then perform preference evaluation
- CareDB Join
- Prune tuples from base relation that can never be preference answers
- Refine join result to form correct preference answer
- Generic to many preference methods
- More efficient than naive approach

CareDB in Action

User Preference Profiles

- Users can log in and edit their preference profiles explicitly using attributes for restaurants and hotels
- Some contextual attributes are uncertain (reported as range), while others require expensive third-party access

Querying CareDB

- Users can view SQL query extension with preference syntax that was built and executed in CareDB
- Users can also issue ad-hoc queries in CareDB, and view query plans, using graphical client connected to CareDB

Extensible Preference Definition

- Using the CareDB extensible preference query processing framework, attendees can add preference methods to CareDB and instantly use the method in preference queries

Efficient Query Processing For Expensive Attributes

- CareDB optimizations for expensive attributes
- Perform preference evaluation over local (cheap) attributes
- Prune objects guaranteed to never be preference answers
- Minimize subsequent expensive attribute requests for correct answer

Preference Query Processing for Uncertain Data

- Uncertainty inherent in many contextual data sources
- Restaurant prices reported as a range
- Travel time reported with error
- CareDB supports preference query processing over uncertain data reported as range
- Efficient two-phase filter/refine algorithm
- Objects reported with probability of being preference answer

Efficient Generic Preference Join

- PruneJoin
- Refinement
- Join
- Local Pruning
- Local Pruning

CareDB

Writing Queries

- $\text{SELECT} *$
- $\text{FROM} \text{Restaurants} \text{ R}$
- $\text{WHERE} \text{[Where clause]}$
- $\text{PREFERENCES} \text{OF} \text{[Attribute/Objectives]}$
- $\text{WITH} \text{PREF METHOD} \text{MyPref}$

CareDB in Action

Environment

- Users press “find restaurant/hotel” button.
- Answers displayed on Google Maps.
- Answers generated by CareDB executing in PostgreSQL

*This work is supported in part by the National Science Foundation under Grant 15-861998, 13-301935, CNS-0708604, IIS-0952977 and by a Microsoft Research Gift.