Multi-dimensional Skyline to find shopping malls

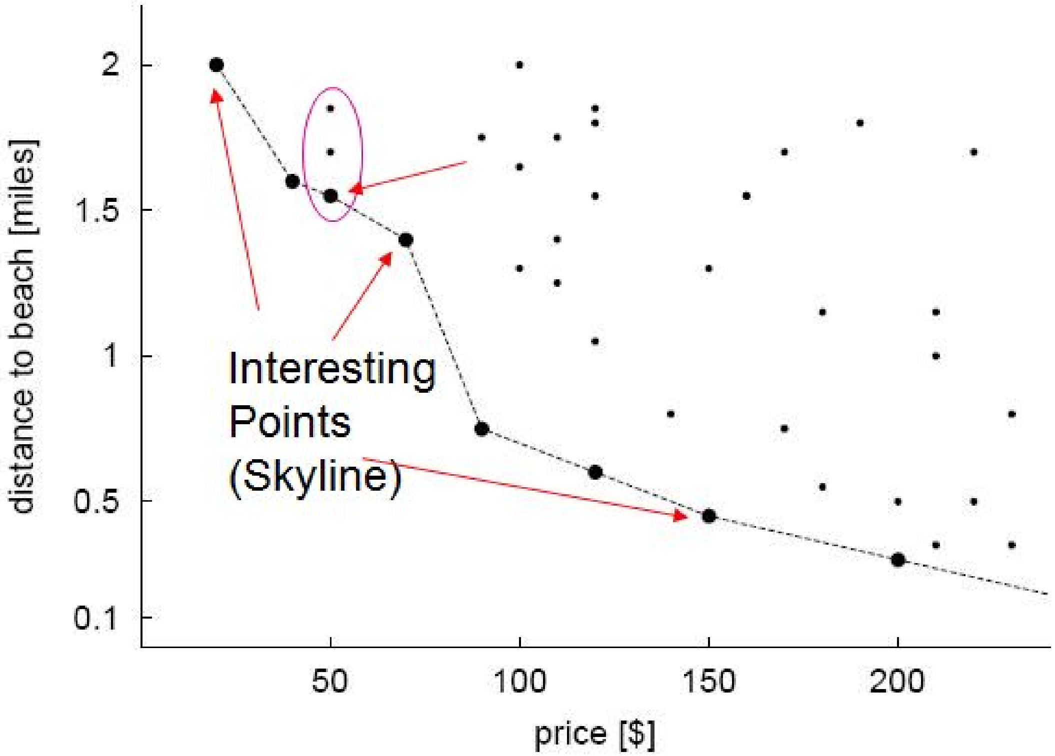
Md “Amir” Amiruzzaman
Suphanut “Parn” Jamonnak
Zhengyong Ren

Introduction

In market research predicting customer movement is very important. While customers to decide which shopping mall to go to depends on many uncertain or probabilistic factors, so, it is not easy to compute their movement ahead of time. However, with the help of uncertain or probabilistic data management, it is possible to compute customer choices with some certainty.

Skyline query?

Skyline



- Minimize price (x-axis)
- Minimize distance to beach (y-axis)
- Points not dominated by other points
- Skyline contains everyone's favorite hotel regardless of preferences

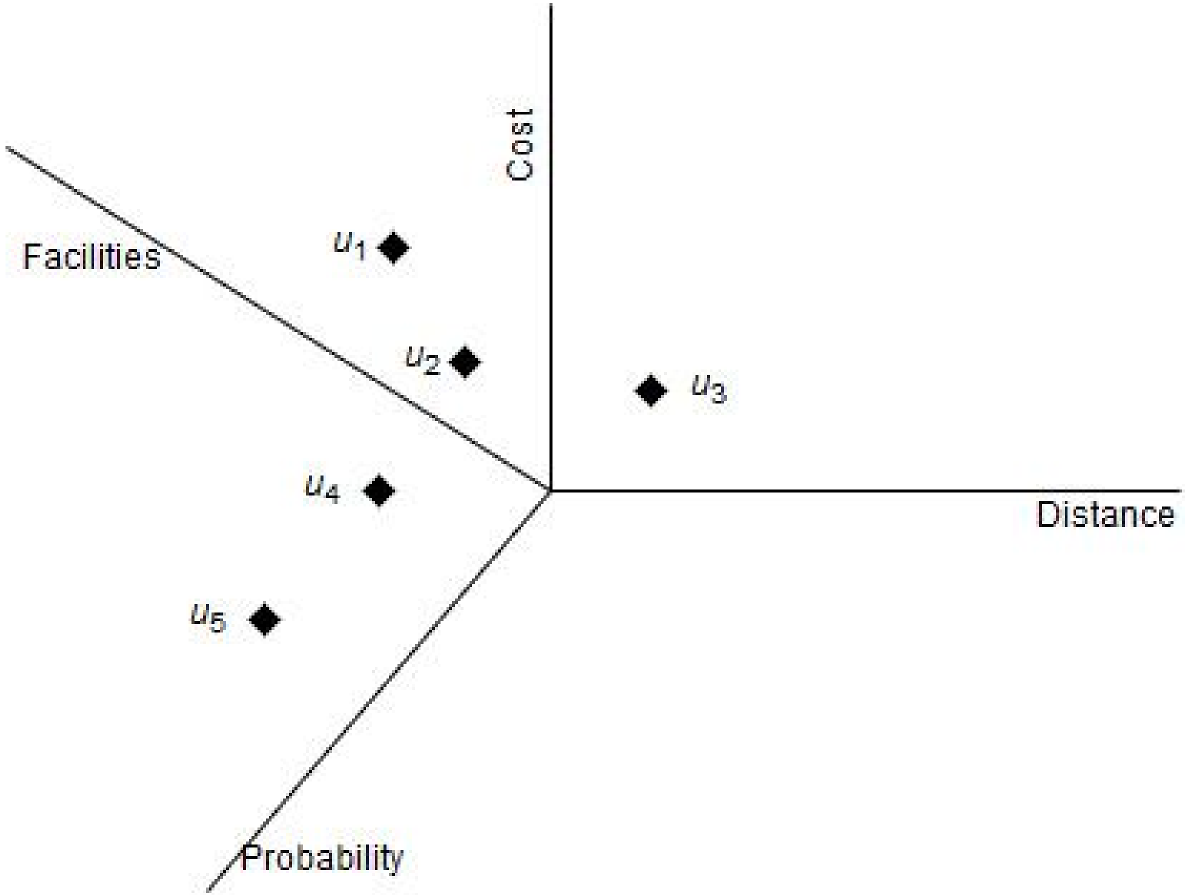
Problem statement

Formal (mathematical) definitions of problems

Let, $U = \{u_1, u_2, u_3, \ldots, u_n\}$ are users/customer who shops in different shopping malls, $S = \{s_1, s_2, s_3, \ldots, s_n\}$. However, users have preference which can be represented as keyword $K = \{k_1, k_2, k_3, \ldots, k_n\}$.

The distance of each shopping malls are $D = \{d_1, d_2, d_3, \ldots, d_n\}$, and price of products $C = \{c_1, c_2, c_3, \ldots, c_n\}$, however, we may consider total cost or sum of products, i.e., . Based on previous visits of each shop, probability to pick a shop can be denoted as, $P = \{p_1, p_2, p_3, \ldots, p_n\}$.

Note that, different shops may sale different types of goods, $G = \{g_1, g_2, g_3, \ldots, g_n\}$, and facilities (e.g., restaurant, kids zone, bar, etc.) in each shopping mall may vary as well, $F = \{f_1, f_2, f_3, \ldots, f_n\}$



Cost
Facilities
Distance
Probability
u_1
u_2
u_3
u_4
u_5

If we are interested to know about a particular user (i.e., query user, u_q), then this problem can be represented as a multi-dimensional skyline problem. As such, shorter distance, lower cost, more facilities, higher variety of goods are desirable. Also, for the simplicity of the problem, we will consider higher probability or mostly visited shopping malls first (see Figure 1).

Objective

We would like to let the query user (u_q) to use a tool to find the shopping mall based on all the parameters we mentioned earlier. The tool will find a shopping mall using the multidimensional skyline query.

We also want to perform a user study to find how to improve user experience and usability of this proposed tool.

Literature Review

1. Probabilistic Skylines on Uncertain Data

   Jian Pei, Bin Jiang, Xuemin Lin, Yidong Yuan

2. Computing All Skyline Probabilities for Uncertain Data

   Mikhail J. Atallah, Yinian Qi

3. Skyline Query Processing for Uncertain Data

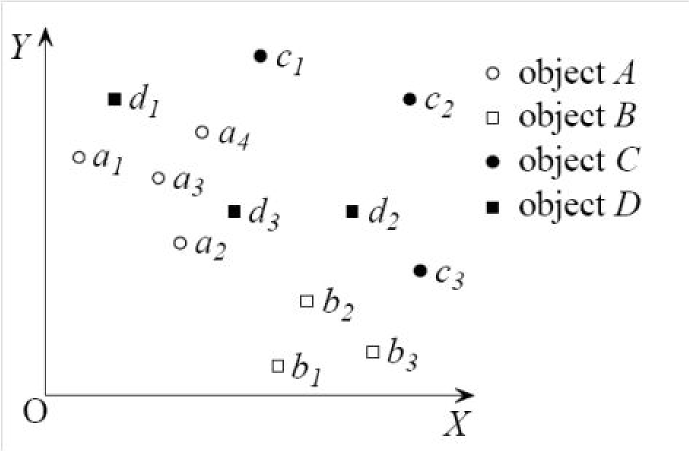
   Mohamed E. Khalefa, Mohamed F. Mokbel, Justin J. Levandoski

Probabilistic Skyline on Uncertain Data

Very Large Data Bases (VLDB), 2007

Example of Calculating Skyline Probability

4 instances of *A*
3 instances of *B*
3 instances of *C*
3 instances of *D*



- The probability $Pr(D)$ that D is not dominated by other objects is given by:

$$
\begin{aligned}
&\tfrac{1}{3} \times (&& D \text{ has three instances} \\
&\quad (1 - \tfrac{1}{4}) + && \text{case of } d_1 \\
&\quad (1 - \tfrac{1}{4}) \times (1 - \tfrac{2}{3}) + && \text{case of } d_2 \\
&\quad (1 - \tfrac{1}{4})\ ) && \text{case of } d_3 \\
&= \tfrac{7}{12}
\end{aligned}
$$

Computing All Skyline Probabilities for Uncertain Data

Mikhail J. Atallah
CS, Purdue University
West Lafayette, IN 47907-2107, USA
mja@cs.purdue.edu

Yinian Qi
CS, Purdue University
West Lafayette, IN 47907-2107, USA
yqi@cs.purdue.edu

Main Contributions：

- The First Sub-Quadratic Algorithm for Computing All Skyline Probabilities
- New Probabilistic Skyline Analysis
- More General Uncertain Data Model

Probabilistic Skyline

- The probability for an instance to be a skyline point is called the instance’s skyline probability.
- The object’s skyline probability is the sum of the skyline probabilities over all its instances.

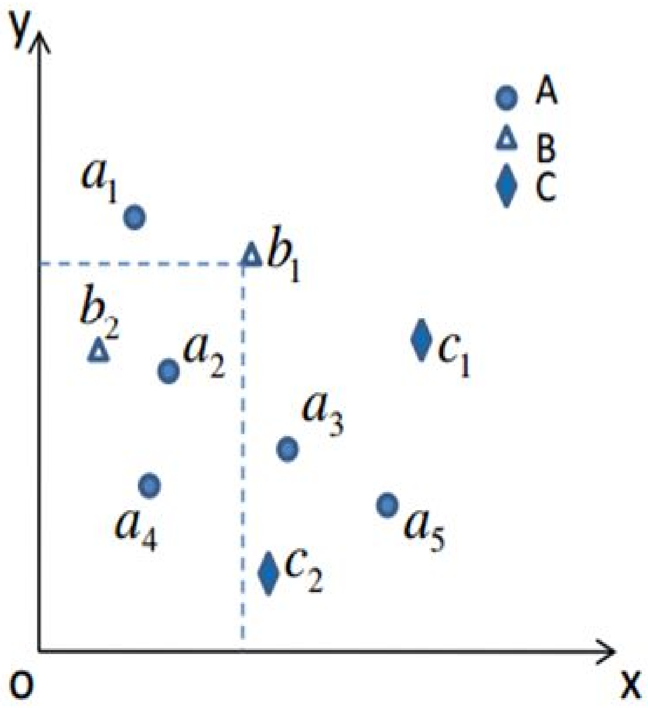


Figure 3: Skyline computation with uncertainty

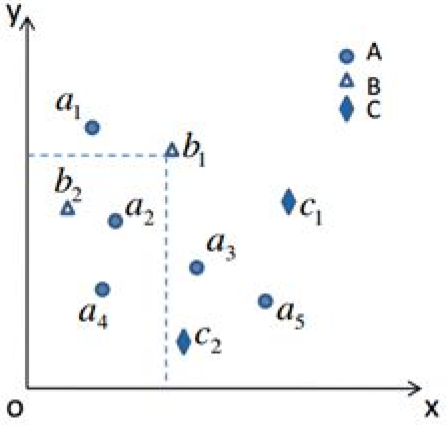
A					B		C	
a_1	a_2	a_3	a_4	a_5	b_1	b_2	c_1	c_2
0.3	0.2	0.1	0.1	0.2	0.2	0.4	0.5	0.5

Table 1: Instance probabilities in Figure 3

For example, B has two instances b1 and b2. b2 is not dominated by any point, so its skyline probability is simply its own probability 0.4. For b1 to be a skyline point, none of the points that dominate b1 (i.e., a2, a4, b2, points in the rectangle) should exist. Hence its skyline probability is 0.2 * (1 - 0.2 - 0.1) = 0.14. The skyline probability of B is 0.54.

The Grid Method

Notation	Meaning
m	number of all uncertain objects
n	number of all instances
d	number of dimensions
O_i	the ith uncertain object
n_i	number of instances of O_i
S	the set of all instances ($n = \|S\|$)
S_i	the set of instances of O_i ($n_i = \|S_i\|$)
p	point/instance in S
$Pr_{sky}(\cdot)$	skyline probability
$D_{S,i}(p)$	instances of O_i in S that dominate p
$\sigma_i(p)$	sum of probabilities of O_i's instances that dominate p
$\beta(p)$	the probability that p is not dominated by any instance of other object



$$\beta(p) = \prod_{i=1, i \neq j}^{m} (1 - \sigma_i(p))$$

$$Pr_{sky}(p) = Pr(p) \cdot \beta(p)$$

In Figure 3, instance b1 is dominated by instances a2, a4 and b2. Therefore, $\beta(b1) = 1 - (Pr(a2) + Pr(a4)) = 0.7$. $Pr_{sky}(b1) = Pr(b1) \cdot \beta(b1) = 0.14$

1.Process the horizontal grid lines:

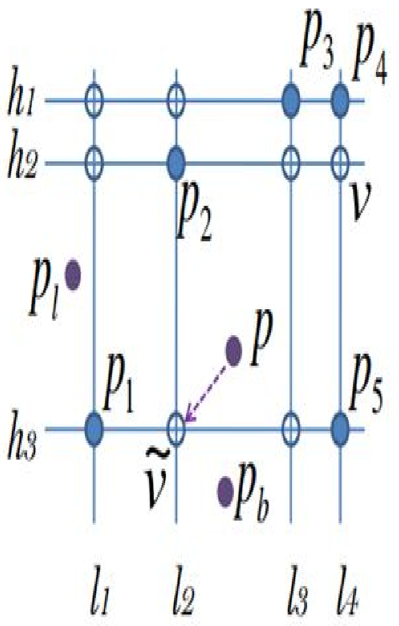


Figure 4: Space partitioning using a grid

$$\sigma_i^*(p) = \sum_{p' \in S_i, p' <_h p} Pr(p')$$

Example 7: *In Figure 4, p_1 is an instance of O_1 with probability 0.8, p_2 and p_4 are instances of O_2 with probability 0.5 each, p_3 and p_5 are instances of O_3 with respective probabilities 0.6 and 0.1. Then for p_4 on the horizontal line h_1, $\sigma_1^*(p_4) = \sigma_2^*(p_4) = 0$ while $\sigma_3^*(p_4) =$* 0.6.

2.Process the vertical grid lines

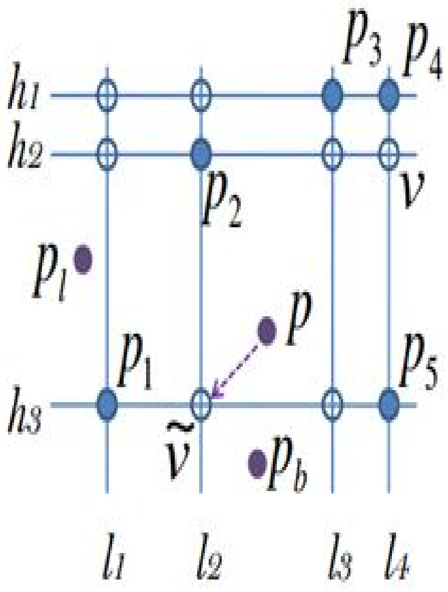


Figure 4: Space partitioning using a grid

$$\sigma_i(p) = \sigma_i^*(p) + \sigma_i(p') + \begin{cases} Pr(p') & \text{if } p' \in S_i \\ 0 & \text{otherwise} \end{cases}$$

Example 8: *To compute $\sigma_i(p_4)$'s from $\sigma_i^*(p_4)$'s computed in Example 7, we follow Equation 5 (take $i = 3$ for example):*

$$\begin{aligned} \sigma_3(p_4) &= \sigma_3^*(p_4) + \sigma_3(v) + 0 \\ &= Pr(p_3) + \sigma_3^*(v) + \sigma_3(p_5) + Pr(p_5) \\ &= 0.6 + 0 + \sigma_3^*(p_5) + 0.1 = 0.7 \end{aligned}$$

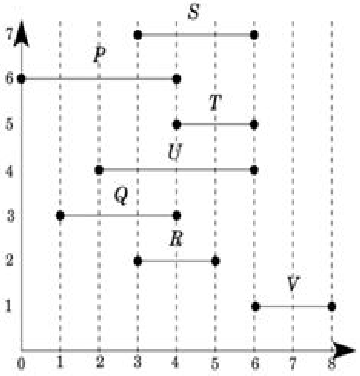
Similarly, we compute $\sigma_1(p_4) = 0.8$, $\sigma_2(p_4) = 0.5$.

Skyline Query Processing for Uncertain Data *

Mohamed E. Khalefa
University of Minnesota
Minneapolis, MN, USA
khalefa@cs.umn.edu

Mohamed F. Mokbel
University of Minnesota
Minneapolis, MN, USA
mokbel@cs.umn.edu

Justin J. Levandoski
University of Minnesota
Minneapolis, MN, USA
justin@cs.umn.edu



(a) Running Example

Point	Skyline
P	60%
Q	91%
R	100%
S	0.5%
T	0%
U	16%
V	100%

(b) Probabilities

Figure 1: Skyline example over data with uncertain ranges

Table 1: Related Work

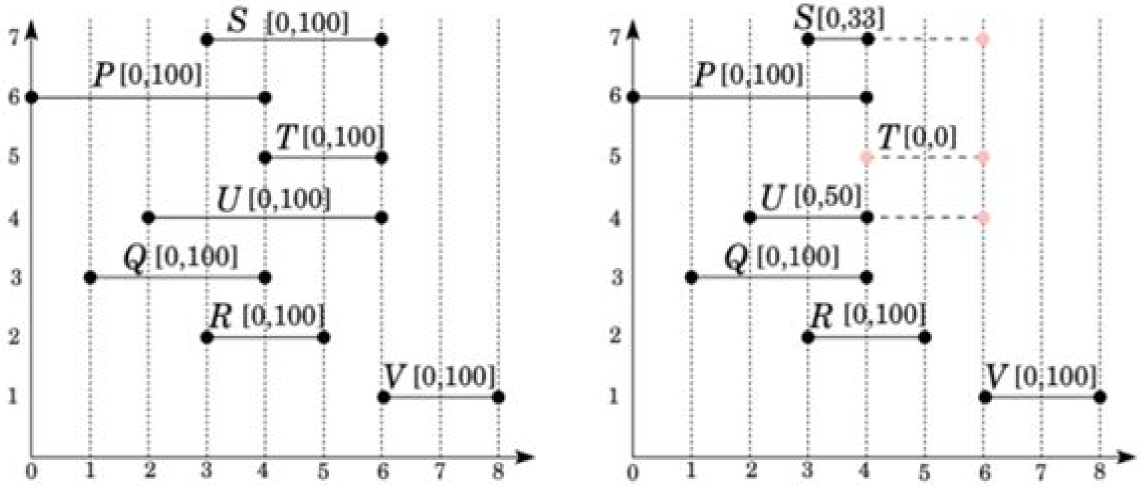
Related work	Query	Continuous	Threshold	Tolerance
[5]	Range & 1-NN	√	—	—
[1]	1-NN	√	√	—
[3]	1-NN	√	√	√
[4]	k-NN	√	√	—
[14, 18]	Rank	√	—	—
[13]	Reserve Skyline	√	√	—
[19]	Top-k	—	—	—
[8]	Top-k	—	√	—
[17]	Rank	—	—	—
[16]	Skyline	—	√	—
Our work	Skyline	√	√	√

This paper propose an efficient framework that supports skyline queries for uncertain data represented as a continuous range.

Uncertainty reduction

- An ordered pair of objects (Q,P) qualifies to uncertainty reduction only if the endpoint of Q dominates the endpoint of P.
- Reduce the upper bound probability for an object P by removing a portion of its uncertainty range which would have a zero probability being a skyline object.

Example



(a) Sample dataset **(b) After uncertainty reduction**

Example:For object V , as the endpoint of the uncertainty range eV does not dominate any other endpoint, V doesn't have any uncertainly reduction.

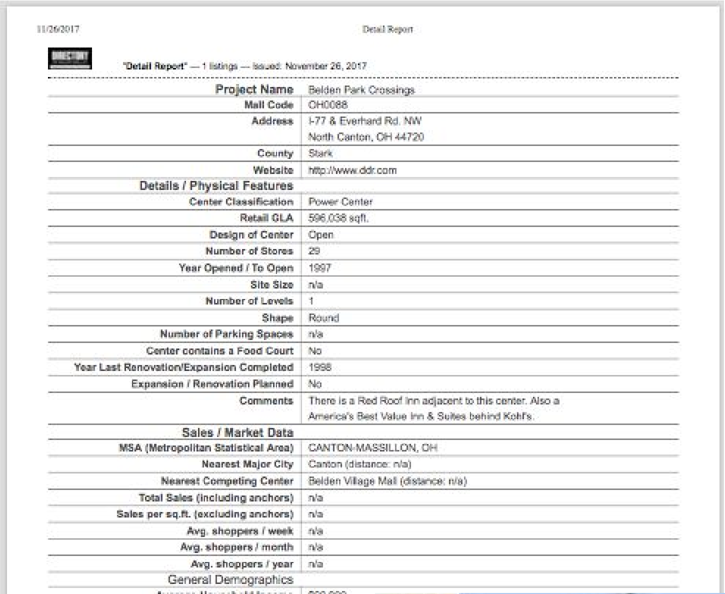
For object R, eR dominates eU ,eT , and eS, so, the pairs (R,U), (R,T), and (R,S) qualify for uncertainty reduction. This results in reducing the uncertainty range of U to be [2-5] instead of [2-6]. Since the reduced range is one quarter of the original range, the upper bound probability of U is set to 75%.

Respectively. Figure 2b gives the result of all points after the uncertainty reduction with their upper probability bounds, pruning objects S, T and U.

Implementation

Data collection

1. Collect data from http://www.shoppingcenters.com with detail report for each shopping mall (focused in Cleveland/Akron areas)
2. Manually fill in the spreadsheet
3. Selecting attributes:
   a. Shopping Mall Name (Text)
   b. Shopping Mall Code (ID)
   c. Stores (Number)
   d. Parking Space (Number)
   e. Household Income (Number)
   f. Population (Number)
   g. Food Court (Yes/No)
   h. Facilities and Categories (Total Sum)
4. Total: 90 Shopping Malls (Remove missing data)


11/26/2017 Detail Report

"Detail Report" — 1 listings — Issued: November 26, 2017

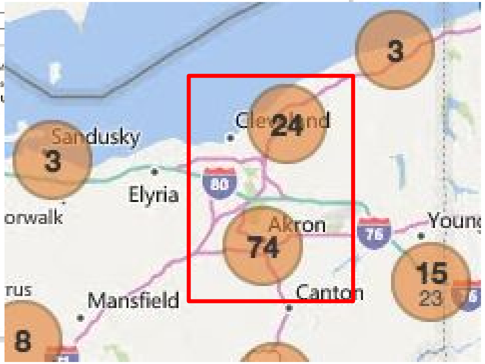
Project Name	Belden Park Crossings
Mall Code	OH0088
Address	I-77 & Everhard Rd. NW North Canton, OH 44720
County	Stark
Website	http://www.ddr.com
Details / Physical Features	
Center Classification	Power Center
Retail GLA	596,038 sqft.
Design of Center	Open
Number of Stores	29
Year Opened / To Open	1997
Site Size	n/a
Number of Levels	1
Shape	Round
Number of Parking Spaces	n/a
Center contains a Food Court	No
Year Last Renovation/Expansion Completed	1998
Expansion / Renovation Planned	No
Comments	There is a Red Roof Inn adjacent to this center. Also a America's Best Value Inn & Suites behind Kohl's.
Sales / Market Data	
MSA (Metropolitan Statistical Area)	CANTON-MASSILLON, OH
Nearest Major City	Canton (distance: n/a)
Nearest Competing Center	Belden Village Mall (distance: n/a)
Total Sales (including anchors)	n/a
Sales per sq.ft. (excluding anchors)	n/a
Avg. shoppers / week	n/a
Avg. shoppers / month	n/a
Avg. shoppers / year	n/a
General Demographics	
Average Household Income	$62,800
Population of Primary Market	359,300
Distance of Primary Market	n/a

Page 1 — © 1999-2017 Directory of Major M

Trend demographic data s

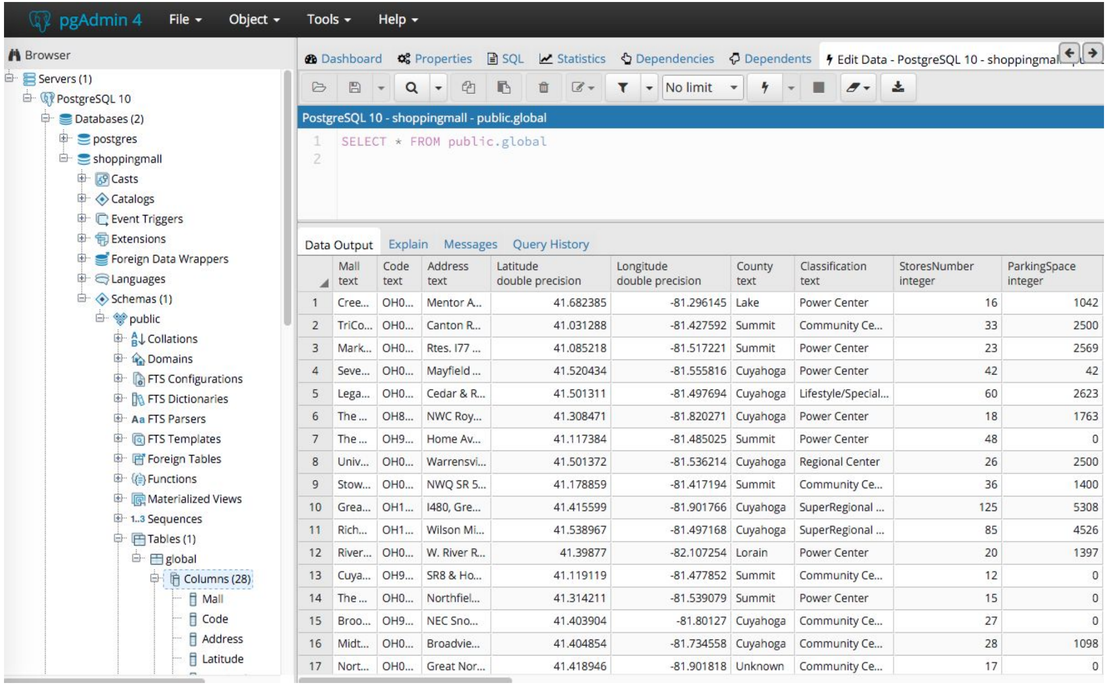
Printed exclusively for Sophanut Jamonnak, L

https://shoppingcenters.com/reports/details/OH0088



Data Preparation and Preprocessing

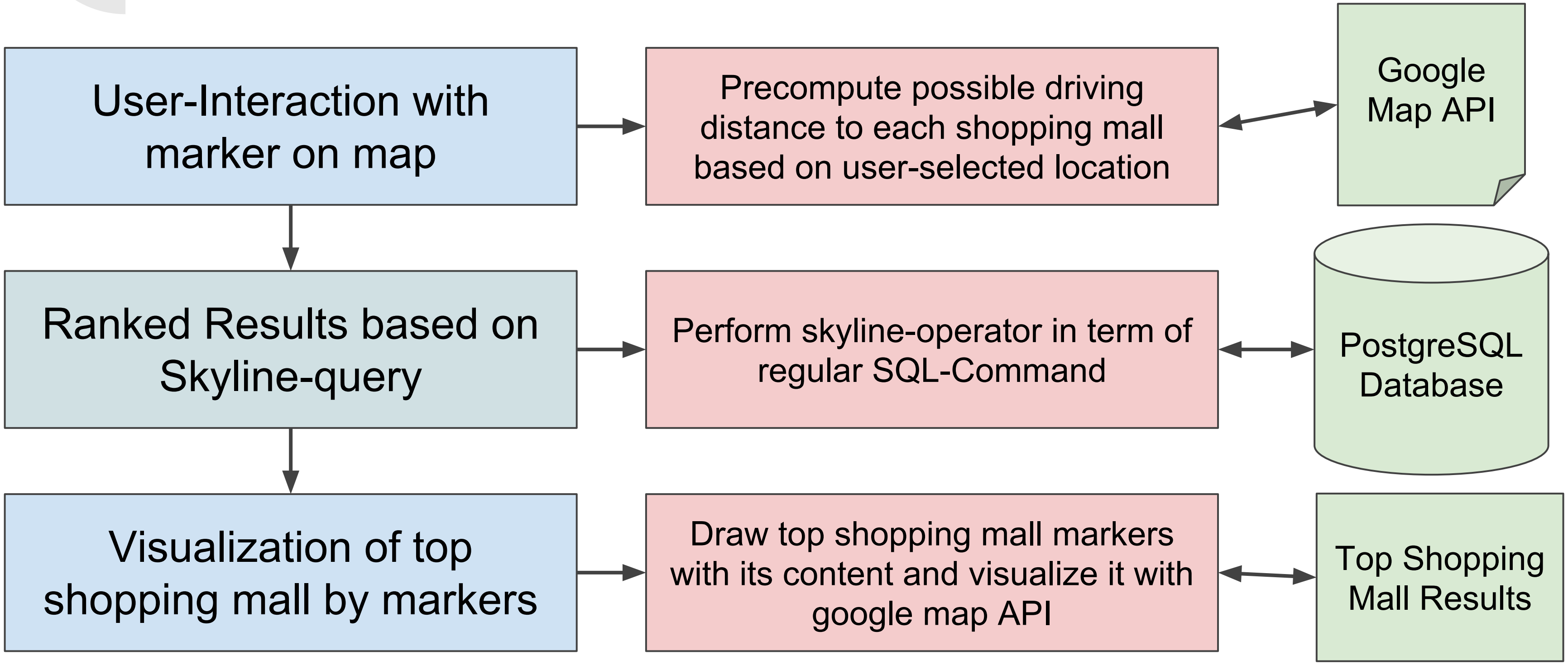
1. Generate geolocation (Latitude and Longitude) for each shopping mall
2. Apply indexes with 2-dimensional points for each shopping mall
3. Import spreadsheet to PostgreSQL Database using PgAdmin4
4. DB name = “Shopping Mall” with 1 table as global view



Methodologies and Design

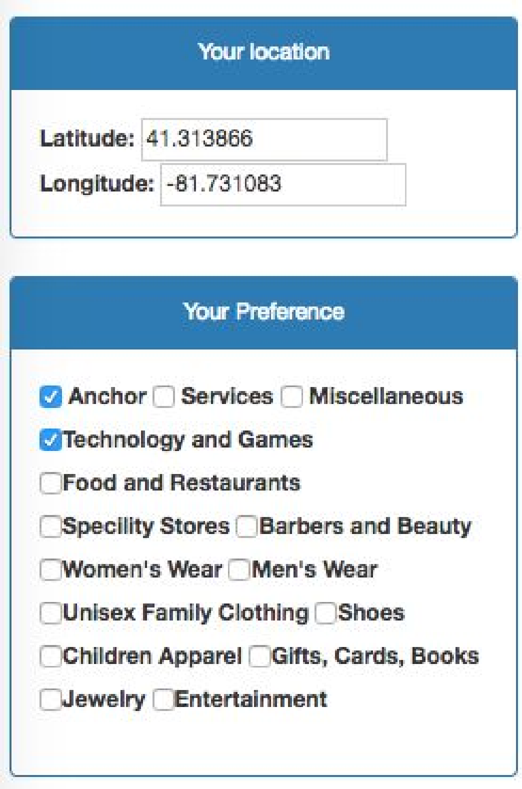
1. Web-Interfaces
2. Front-End: HTML and JavaScript
3. Styles: Bootstrap
4. Back-End: PHP
5. External Library:
    a. Google Maps API
    b. Google Direction Services API
    c. Some pre computation libraries
6. Database: PostgreSQL

System Workflow



Dynamic Location and Preference Inputs

1. User Location
   a. Interaction: Dragging marker over map
2. User Preferences
   a. Anchor: Walmart, Giant Eagles, Target
   b. Services: Chase Bank, NTB Car Repair
   c. Miscellaneous: Yankee Candle, Toy R Us
   d. Hi-Tech: At&t, Time Warner, Gamestop
   e. Foods and Restaurants
   f. …
3. User input will **dynamically** change the SQL command of Sky-line Query



Skyline-Query Methods

Shopping Mall	Stores Number	Parking Space	Household Income	Population
s1	50	20	$20,000	10000
s2	20	0	$50,000	20000
s3	40	100	$30,000	7000
s4	60	40	$40,000	8000
s5	30	50	$45,000	5000

No shopping mall better than another on every criteria.

While no one best shopping mall, we want to **eliminate shopping mall** which are worse on all criteria. In this case is “s2”

Skyline-operator

- **Skyline Operator**
  - **SELECT * FROM global**
    SKYLINE OF Distance MIN,
    Stores Number MAX,
    Parking Space MAX,
    Household Income MIN,
    Population MIN, ...

Can we write SQL query without using Skyline operator?

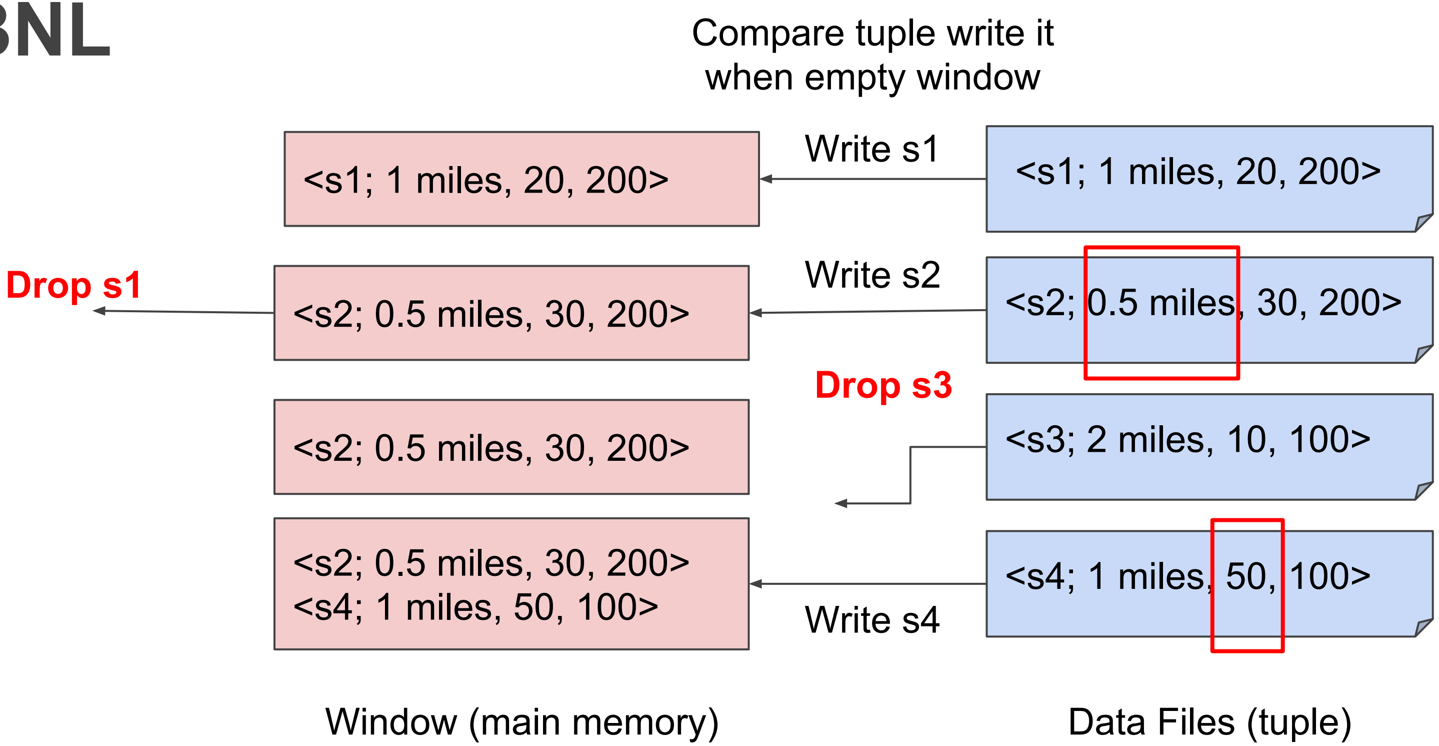
Skyline Implementation in N-dimension

- There are several Skyline-Algorithms presented
- In our project, we Implement a regular SQL query with Skyline Operator:
  - ```
    SELECT *
    FROM ShoppingMall S
    WHERE NOT EXISTS ( SELECT * FROM ShoppingMall S1
                          AND S1.Distance <= S.Distance
                          AND S1.StoresNum >= S.StoresNum
                          AND S1.ParkingSpace >= S.ParkingSpace
                          AND ( S1.Distance < S.Distance OR
                                S1.StoresNum > S.StoresNum OR
                                S1.ParkingSpace > S.ParkingSpace ));
    ```
- This SQL query is equivalent to previous example but without skyline operator
- After generate the result, we sort it by distance in descending order
    ```

Block-nested Loop (BNL) Algorithms

- Block Nested Loop
- Compare each tuple with one another
- Window in main memory contain best tuple
- Write to temp file (if window has no space)
- Implement in javascript and compare it with SQL-Skyline result
- pick an optimal shopping malls.

BNL


Compare tuple write it
when empty window
Write s1
<s1; 1 miles, 20, 200>
<s1; 1 miles, 20, 200>
Drop s1
Write s2
<s2; 0.5 miles, 30, 200>
<s2; 0.5 miles, 30, 200>
Drop s3
<s2; 0.5 miles, 30, 200>
<s3; 2 miles, 10, 100>
<s2; 0.5 miles, 30, 200>
<s4; 1 miles, 50, 100>
Write s4
<s4; 1 miles, 50, 100>
Window (main memory)
Data Files (tuple)

Example Result

Dominate Shopping Mall

1. The Shoppes at Parma
2. The Greens of Strongsville
3. Southland Shopping Center
4. Midtown Plaza
5. Brunswick Town Center
6. Ridge Park Square
7. Great Northern Mall
8. Southgate USA
9. Beachcliff Market Square
10. Medwick Marketplace

Demo

Evaluation Result

- Select 10 Participants
- Rating 1 to 5 based on our ranking result
- Collect comment and feedback
- Will include this part in final report

Conclusion

- There are several Skyline-Query algorithms out there, we found that our SQL command and BNL methods is cumbersome, expensive to evaluate, and huge result set
- Both BNL and SQL command need to improve
    - E.g. Create Self-Organizing list for BNL algorithms
- Our system works with dynamic user input
- Future Work
    - Implement more Skyline algorithms (R-tree, Divide and Conquer, K-NN)
    - Evaluate and summarize which algorithm is the best to rank shopping mall from our dataset.
    - Perform user study with the domain experts

Q & A

Md “Amir” Amiruzzaman
Suphanut “Parn” Jamonnak
Zhengyong Ren