

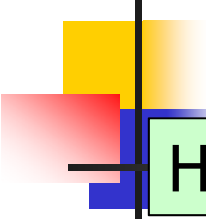


Understanding Users' Preference on Big Data

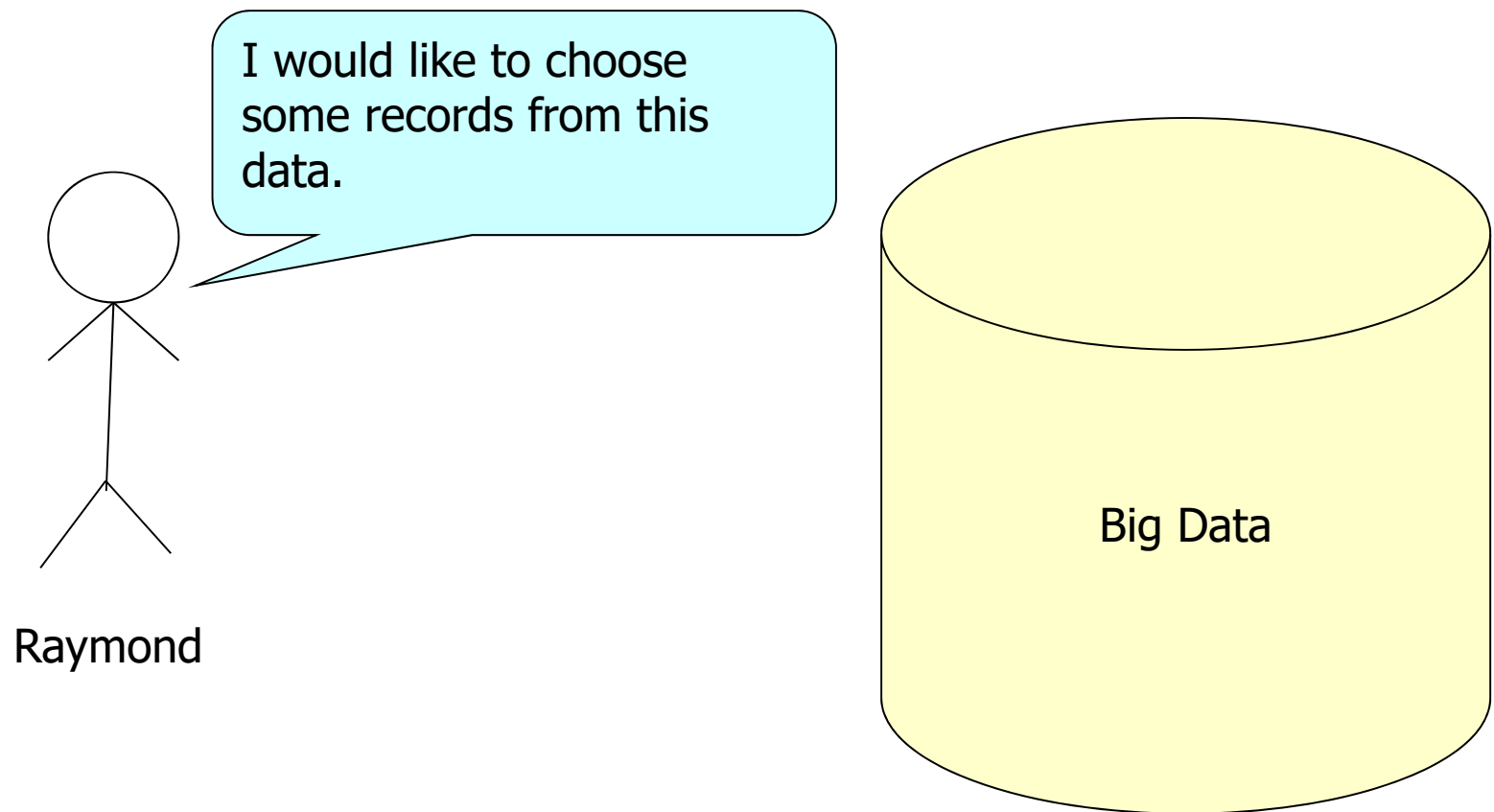
Raymond Chi-Wing Wong

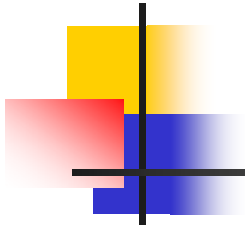
The Hong Kong University of Science and
Technology

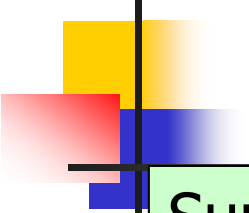
Prepared by Raymond WONG
Presented by Raymond WONG



How can we understand the preference from Raymond?







Which apartment should Raymond buy?

Suppose that user Raymond wants to buy an apartment

One example is the apartment size.

There are 2 popular queries for this problem.

→ Top-k queries

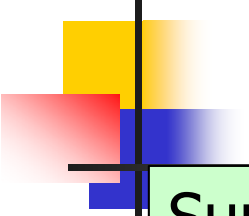
Skyline queries

Recently, I focus on a new type of queries.

k-regret queries

D

Apartment	x_1	x_2
p_1	0	1
p_2	0.2	1
p_3	0.6	0.9
p_4	0.9	0.6
p_5	1	0.2
p_6	1	0
...



Suppose that user Raymond wants to buy an apartment

Top-k queries

D	Apartment	x_1	x_2
	p_1	0	1
	p_2	0.2	1
	p_3	0.6	0.9
	p_4	0.9	0.6
	p_5	1	0.2
	p_6	1	0

Top-k queries

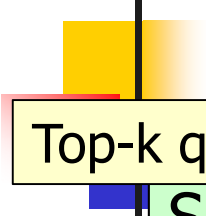
Suppose that user Raymond wants to buy an apartment

- Assume that Raymond has a “known” preference function.
- We could find the “best” apartment for Raymond immediately

Advantage: The output size is “fixed”

Disadvantage: We need to know the “exact” preference function of Raymond

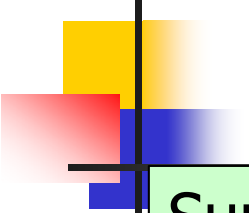
D	Apartment	x_1	x_2
	p_1	0	1
	p_2	0.2	1
	p_3	0.6	0.9
	p_4	0.9	0.6
	p_5	1	0.2
	p_6	1	0



Top-k queries

Suppose that user Raymond wants to buy an apartment

- My previous work
 - k-Hit Query: Top-k Query with Probabilistic Utility Function (SIGMOD 2015)



Which apartment should Raymond buy?

Suppose that user Raymond wants to buy an apartment

One example is the apartment size.

There are 2 popular queries for this problem.

Top-k queries

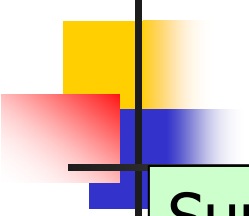
→ Skyline queries

Recently, I focus on a new type of queries.

k-regret queries

D

Apartment	x_1	x_2
p_1	0	1
p_2	0.2	1
p_3	0.6	0.9
p_4	0.9	0.6
p_5	1	0.2
p_6	1	0
...



Suppose that user Raymond wants to buy an apartment

Skyline queries

D	Apartment	x_1	x_2
	p_1	0	1
	p_2	0.2	1
	p_3	0.6	0.9
	p_4	0.9	0.6
	p_5	1	0.2
	p_6	1	0

Skyline queries

Suppose that user Raymond wants to buy an apartment

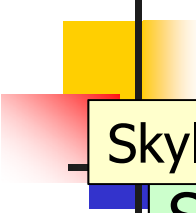
- There is no assumption that we know the “exact” preference function of Raymond
- The major idea is to remove all apartments that Raymond must not consider
- The remaining apartments are all possible candidates that Raymond will consider
- Candidate apartments = $\{p_2, p_3, p_4, p_5\}$

Advantage: There is no need to specify the preference function of Raymond

Disadvantage: The output size is uncontrollable.

D

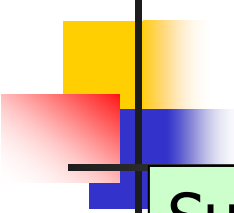
Apartment	X_1	X_2
p_1	0.1	0.1
p_2	0.2	1
p_3	0.6	0.9
p_4	0.9	0.6
p_5	1	0.2
p_6	1	1
...



Skyline queries

Suppose that user Raymond wants to buy an apartment

- My previous work
 - Skyline Queries and Pareto Optimality (Encyclopedia of Database Systems, 2016)
 - Finding Competitive Price (SIGSPATIAL GIS 2013)
 - Finding Top-k Preferable Products (TKDE 2012)
 - Finding Top-k Profitable Products (ICDE 2011)
 - Creating Competitive Products (VLDB 2009)
 - Online Skyline Analysis with Dynamic Preferences on Nominal Attributes (TKDE 2009)
 - Finding the Influence Set through Skylines (EDBT 2009)
 - Efficient Skyline Querying with Variable User Preferences on Nominal Attributes (VLDB 2008)
 - Mining Favorable Facets (SIGKDD 2007)



Which apartment should Raymond buy?

Suppose that user Raymond wants to buy an apartment


One example is the apartment size.

There are 2 popular queries for this problem.

Top-k queries

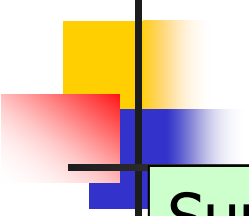
Skyline queries

Recently, I focus on a new type of queries.

 k-regret queries

D

Apartment	x_1	x_2
p_1	0	1
p_2	0.2	1
p_3	0.6	0.9
p_4	0.9	0.6
p_5	1	0.2
p_6	1	0
...



Suppose that user Raymond wants to buy an apartment

k-regret queries

D	Apartment	x_1	x_2
	p_1	0	1
	p_2	0.2	1
	p_3	0.6	0.9
	p_4	0.9	0.6
	p_5	1	0.2
	p_6	1	0

k-regret queries

Suppose that user Raymond wants to buy an apartment

- It has **both** the advantage of the top-k queries and the advantage of the skyline queries.

The output size is specified by parameter k (e.g., 2)

Advantage: The output size is "fixed"

Advantage: There is no need to specify the preference function of Raymond

D


Apartment	x_1	x_2
p_1	0	1
p_2	0.2	1
p_3	0.6	0.9
p_4	0.9	0.6
p_5	1	0.2
p_6	1	0
...



k-regret queries

Suppose that user Raymond wants to buy an apartment

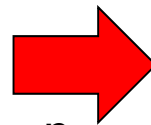
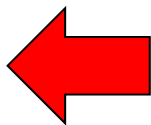
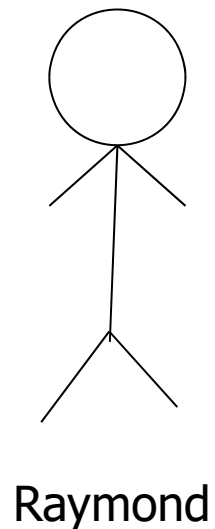
■ My previous work

- 
- An Experimental Survey of Regret Minimization Query and Variants: Bridging the Best Worlds between Top-k Query and Skyline Query (VLDBJ 2020)
 - Strongly Truthful Interactive Regret Minimization (SIGMOD 2019)
 - FindYourFavorite: An Interactive System for Finding the User's Favorite Tuple in the Database (SIGMOD 2019 (demo paper))
 - Finding Average Regret Ratio Minimizing Set in Database (ICDE 2019)
 - Efficient k-Regret Query Algorithm with Restriction-free Bound for any Dimensionality (SIGMOD 2018)
 - k-Regret Minimizing Set: Efficient Algorithms and Hardness (ICDT 2017)
 - Minimizing Average Regret Ratio in Database (SIGMOD 2016 (Undergraduate Research Competition))
 - Geometry Approach for k-Regret Query (ICDE 2014)

k-regret queries

Suppose that user Raymond wants to buy an apartment

- Suppose that we do not know Raymond's preference at the beginning.



p_3

Which one is better?

1. p_2
2. p_3

After this round,
we understand Raymond's
preference better.

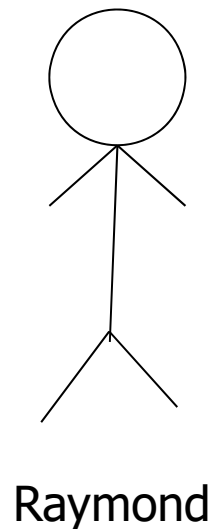
D

Apartment	x_1	x_2
p_1	0	1
p_2	0.2	1
p_3	0.6	0.9
p_4	0.9	0.6
p_5	1	0.2
p_6	1	0
...

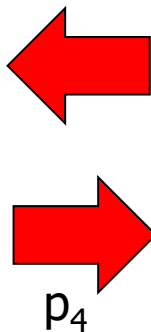
k-regret queries

Suppose that user Raymond wants to buy an apartment

- Suppose that we do not know Raymond's preference at the beginning.



HKUST



Which one is better?

1. p_4
2. p_5

After this round,
we understand Raymond's
preference much better.

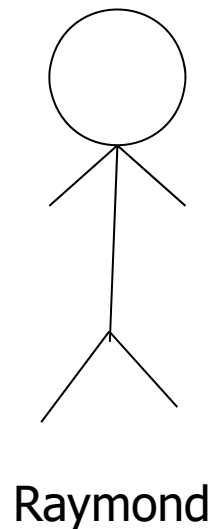
D	Apartment	x_1	x_2
	p_1	0	1
	p_2	0.2	1
	p_3	0.6	0.9
	p_4	0.9	0.6

With more
rounds/questions, we
could know Raymond's
preference more.

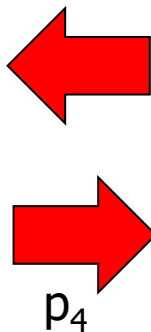
k-regret queries

Suppose that user Raymond wants to buy an apartment

- Suppose that we do not know Raymond's preference at the beginning.



HKUST



Which one is better?

1. p_4
2. p_5

After this round,
we understand Raymond's
preference much better.

D	Apartment		
	x_1	x_2	
	p_1	0	1
	p_2	0.2	1
	p_3	0.6	0.9
	p_4	0.9	0.6

Finally, we know
Raymond's preference and
recommend the best
apartment to him.



k-regret queries

Suppose that user Raymond wants to buy an apartment

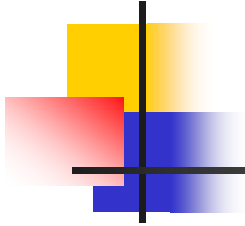
- My previous work

- An Experimental Survey of Regret Minimization Query and Variants: Bridging the Best Worlds between Top-k Query and Skyline Query (VLDBJ 2020)
- Strongly Truthful Interactive Regret Minimization (SIGMOD 2019)
- ➔ ■ FindYourFavorite: An Interactive System for Finding the User's Favorite Tuple in the Database (SIGMOD 2019 (demo paper))
- Finding Average Regret Ratio Minimizing Set in Database (ICDE 2019)
- Efficient k-Regret Query Algorithm with Restriction-free Bound for any Dimensionality (SIGMOD 2018)
- k-Regret Minimizing Set: Efficient Algorithms and Hardness (ICDT 2017)
- Minimizing Average Regret Ratio in Database (SIGMOD 2016 (Undergraduate Research Competition))
- Geometry Approach for k-Regret Query (ICDE 2014)



Demo System

- We developed a demo system on a car database with the following attributes
 - Price
 - Year
 - Power
 - Used km

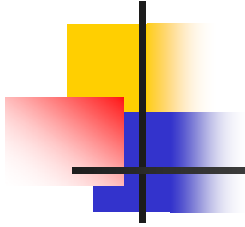


Your Choice

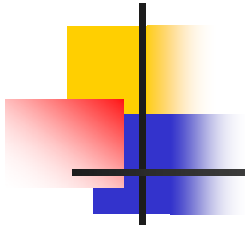
Q1: Choose the Car You Favor More among the Following Options

Option	Price (USD)	Year	Power (PS)	Used KM	
1	10500	2015	110	10000	<input type="button" value="Choose"/>
2	8000	2011	156	30000	<input type="button" value="Choose"/>

Raymond chooses this option.
Then, he is asked for several questions and keep choosing options.



- Finally, he obtains the following answer.



Total No. of Questions Asked is: 9.

Your Favourite Car is:

Price (USD)	Year	Power (PS)	Used KM
4200	2003	276	150000



Conclusion

- Queries
 - Top-k queries
 - Skyline queries
 - K-regret queries
- We just illustrated to select “apartments” and “cars”
- In fact, all applications which requires users to select some records could be applied.
- Thus, there are a lot of applications.



- If you are interested in some of my projects, please feel free to contact me.

Raymond WONG

raywong@cse.ust.hk