DATA REFINEMENT APPROACH FOR ANSWERING WHY-NOT PROBLEM OVER K-MOST PROMISING PRODUCT ($K - MPP$) QUERIES

Vynska Amalia Permadi[1)], Tohari Ahmad[2)], and Bagus Jati Santoso[3)]
[1,2,3)] Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
e-mail: vynska16@mhs.if.its.ac.id[1)], tohari@if.its.ac.id[2)], bagus@if.its.ac.id[3)]

ABSTRAK

K-Most Promising Product ($K - MPP$) adalah strategi product selection yang digunakan pada proses pencarian k-produk yang paling banyak diminati oleh customer. Dasar komputasi yang digunakan untuk melakukan perhitungan $K - MPP$ adalah dua tipe skyline query, yaitu: dynamic skyline dan reverse skyline. Penentuan $K - MPP$ dilakukan pada layer aplikasi, yang merupakan layer paling atas pada model OSI. Salah satu fungsi layer aplikasi adalah untuk menyediakan layanan terbaik sesuai dengan keinginan user.

Dalam implementasi $K - MPP$, akan muncul suatu keadaan dimana produsen mungkin kurang puas dengan query result yang dihasilkan pada proses pencarian di sistem database (why-not question), sehingga mereka juga ingin mengetahui mengapa sistem database memberikan hasil pencarian query yang tidak sesuai dengan harapannya. Sebagai contoh, produsen ingin mengetahui mengapa suatu data point tertentu yang tidak diharapkan (unexpected data) muncul di query result, dan mengapa produk yang diharapkan (expected data) tidak muncul sebagai query result. Permasalahan yang muncul selanjutnya adalah, sistem database tradisional tidak dapat memberikan fasilitas analisis data dan solusi untuk menjawab why-not question yang diajukan oleh user.

Untuk meningkatkan usability pada sistem database, penelitian ini dilakukan dengan tujuan menjawab why-not $K - MPP$ dan memberikan solusi berupa data refinement dengan mempertimbangkan user feedback sehingga user dapat mengetahui mengapa himpunan hasil yang muncul tidak sesuai dengan harapan, dan dapat membantu user untuk memahami serta mengubah query agar menghasilkan query result sesuai keinginan user namun dengan cost perubahan seminimal mungkin.

Kata Kunci: *Dynamic Skyline, Data Refinement, $K - MPP$, Reverse Skyline, Sistem Database, Why-not $K - MPP$.*

ABSTRACT

K-Most Promising (K-MPP) product is a strategy for selecting a product that used in the process of determining the most demanded products by consumers. The basic computations used to perform K-MPP are two types of skyline queries: dynamic skyline and reverse skyline. K-MPP selection is done on the application layer, which is the last layer of the OSI model. One of the application layer functions is providing services according to the user's preferences.

In the K-MPP implementation, there exists the situation in which the manufacturer may be less satisfied with the query results generated by the database search process (why-not question), so they want to know why the database gives query results that do not match their expectations. For example, manufacturers want to know why a particular data point (unexpected data) appears in the query result set, and why the expected product does not appear as a query result. The next problem is that traditional database systems will not be able to provide data analysis and solution to answer why-not questions preferred by users.

To improve the usability of the database system, this study is aiming to answer why-not K-MPP and providing data refinement solutions by considering user feedback, so users can also find out why the result set does not meet their expectations. Moreover, it may help users to understand the result by performing analysis information and data refinement suggestion.

Keywords: *Dynamic Skyline, Data Refinement, $K - MPP$, Reverse Skyline, Sistem Database, Why-not $K - MPP$*

I. INTRODUCTION

THE development of information and communication technology has resulted in the emergence of various information digitization processes in recent decades. By processing the data on a computer system using a certain algorithm, new knowledge and information that previously have been unrealized can be emerged. For example, the sales data of a company can be evaluated by performing computation and analysis processes automatically over database system. In that way, the company may have insight into marketing strategies that can be used to sell its products.

Based on the example above, the database system is having the important role on data processing of information and evaluation recently. With the continuous development of architecture and evaluation techniques in the database systems, the execution and processing of queries can now be provided in real time without being constrained by

the amount of data that needs to be evaluated. Generally, the query is a question or information needed by the user.

Research on the database system mainly discusses the efficiency of query execution and resource sharing in order to provide the best system capabilities. However, most end users do not understand the knowledge of the database system. Therefore, there will be a problem when a query result evaluation is required on the database system but only a certain user can perform those task [1].

To improve the usability of the database system, it is important to understand the user expectation of an interactive and informative database system. If the query result is not desired, it is expected that the user may perform the further evaluation without any knowledge of the database system. Moreover, it is expected that the database system can provide brief and informative explanations so that users are able to understand and evaluate the problems that cause the query result. By giving a brief and informative explanation of the query result, the user may easily evaluate and determine the refinement of its query so that the search results provided by the database system in accordance with the expected result. It will also provide an alternative search efficiency for the user as well as resource database savings since the user does not have to perform multiple searches until the desired result appears as a query result.

In [2], Islam & Liu formulate the K-Most Promising Product ($K-MPP$) framework as a product selection strategy used in the product search process that demanded most by the customer. The basic computations used to perform K-MPP calculations are two types of skyline queries, namely: dynamic skyline and reverse skyline. The skyline operator was first proposed in [3]. By doing the query processing using skyline operator, the data of unique value or not dominated value are collected using three types of function that can be selected, there are MIN or minimum, MAX or maximal, and DIFF or different.

In the implementation of $K-MPP$, there exists a situation in which manufacturers may be less satisfied with the query result generated in the search process, so they also want to know why the database system provides query results that do not match their expectations. For example, the producer wants to know why an unexpected data point appears in the query result set hereinafter called why point, and why the expected product does not appear as a query result, hereinafter called the why-not point. Another problem is that traditional database systems do not provide data analysis and solution facilities to answer why-not question submitted by the user as illustrated in the above problem.

Based on above problems, Liu et al [4] identify the causality which is the cause of the expected data and unexpected data on the query result and the responsibility which is the value of the influence of the expected or unexpected data on the probability of reverse skyline query. As a further development, the research also implemented the identification process of causality and responsibility on reverse skyline query. Evaluation is done by testing the effectiveness and efficiency of the identification process of causality and responsibility. However, to solve the why-not question, further steps such as data or query modification are required so that expected data can appear in the query result, as discussed in research [5], [6] and [7].

This research analyzes why the why-not point does not appear as K-MPP and answer the why-not question which appears on the query result of K-MPP by modifying the data value in query or data refinement. It is also expected that data modification has a minimal cost of change possible.

II. Related Work

There are several methods that have been proposed to answer why-not question in query results. [8] Finds the method which can identify the responsible data point that eliminates users' desired tuples on Select-Project-Join (SPJ) queries, while [9] resolve the why-not question on Select-Project-Join-Union-Aggregation (SPJUA) queries. In [10], [11], [12], and [13] data modifications is provided, so that the missing tuples can appear in the query result. [10] and [11] answer the why-not problem on SPJ queries, in the other hand [12] and [13] focused on SPJUA queries. However, query refinement method can also be applied to revise query results in top-k queries as in [14] and [15], and reverse skyline queries [6].

Islam [5] proposed a framework named FlexIQ to answer the why-not and why question on the SPJ query result. By invoking user input as feedback, a new query is specified which can include the why-not point and eliminate the unexpected why-point that appear in the query result. As the efficiency evaluation, this paper proposed two different query determination methods, namely: the baseline algorithm (TBA) and the trade-off algorithm (TOA).

Solutions that can be used to answer the why-not question on the reverse skyline query are discussed in [6]. The proposed solution consists of three parts, which are: identification of data points that cause expected data does not appear as a reverse query result, data point modification or query modification to make the expected data appear as a reverse query result, and modification of data point and modification of query. In this research, the evaluation was performed on the dataset which has two attribute values, or 2D-dimensional data, and the purpose of the

evaluation was to compare the effectiveness and performance of the three proposed modifications to the data cardinality.

Liu et al on [7] discuss the solutions to answer the why-not question on reverse top-k queries. The proposed solution to answer the why-not question is similar to the research that has been done on [6], i.e. by performing combination task of three different modifications: query modification, point weight, and k value modification. This study used five different dimensional settings as in its evaluation to evaluate its performance on various dimensions of the data. The effectiveness and performance of proposed methods also evaluate in the various data cardinality.

III. RESEARCH PROBLEM

$K-MPP$ requires two evaluation models on the customer and product dataset. The first model is the product selection in which searching the skyline query result of a product using reverse skyline query. The result of this product selection evaluation process is the customer data that is interested in each product. In the second model, which is product adoption, this task performs dynamic skyline computation to the dataset, so that the set of products preferred by each customer can be obtained. In the end, determining the most promising product is done by determining the best k-ranking of the overall market contribution of the product. Market contribution (MC) is the amount of probability value (obtained from the evaluation of product adoption model) of all customers who are members of a reverse skyline of a product. If the query search results of $K-MPP$ are not the expected one, no solution can be given to answer the why-not question in $K-MPP$, while it is especially needed by manufacturers and users of database systems to evaluate why this problem occurs. Assuming the area of expertise of the user is different with the database system designer, then it needs an additional informative solution to give usability for database system user.

Based on a brief explanation of $K-MPP$ and the why-not question problem in the database system, this study answers the why-not question that appears in $K-MPP$ that has not been discussed in the previous research. To answer the why-not question on $K-MPP$ query result, as the second contribution, we proposed a data refinement approach by listing all the best query modification solution which have minimum modification cost. From several lists of proposed data refinement, then the validation task is performed to check whether the data refinement is able to answer why-not question at $K-MPP$ or not

With the contribution proposed in this research, it is expected that the refinement data approach can provide an alternative solution to answer the why-not question so that the expected data may be joined as a member of $K-MPP$.

IV. PROPOSED METHOD

Answering why-not $K-MPP$ consist of several stages which can be seen in Figure 1. The main stages in the process of answering the $K-MPP$'s why-not question are: identifying the k rank of the products, increasing the market contribution value by modifying the query value which is the why-not point (data refinement) and validation. Before modifying the query, it is necessary to identify why the why-not point does not appear as a member of $K-MPP$. After the cause is found, then the process of modification of query or data refinement can be done by evaluating the list of data points that appear as members of $K-MPP$. The query modification process will then generate some possible combinations of data refinements on any one dimension of why-not point data.

Having obtained a list of possible combinations of data refinement, then the validation process is performed to ensure the correctness of the provided data refinement. The validation process is done by checking whether the data refinement solution provided can make the why-not point to become one of the $K-MPP$ members.

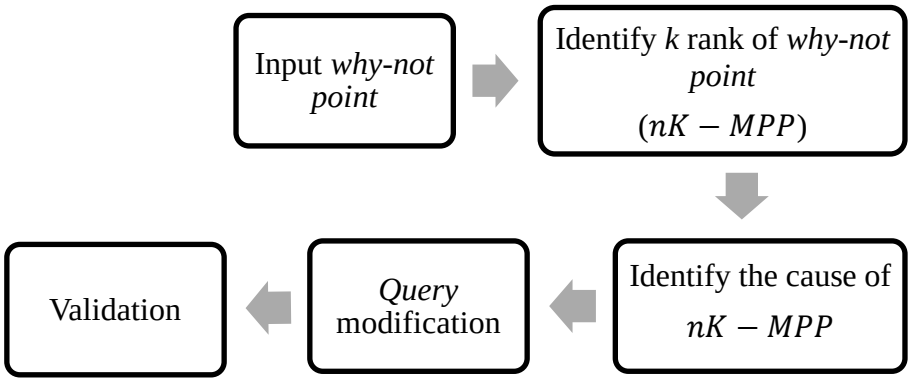

Fig. 1. Design and Implementation of Algorithm to Resolve the Why-Not Question Problem on $K-MPP$

TABLE I
THE DATASET OF PRODUCTS (A) AND CUSTOMER PREFERENCES (B)

ID	d_1	d_2
p_1	12	8
p_2	6	6
p_3	16	14
p_4	20	20
p_5	16	6
p_6	20	8
p_7	4	18
p_8	12	6
p_9	9	15
p_{10}	6	20
q_1	12	12
q_2	7	15
q_3	14	11
q_4	19	11

(A)

ID	d_1	d_2
c_1	10	10
c_2	4	10
c_3	20	13
c_4	12	2
c_5	18	18
c_6	2	8
c_7	8	18
c_8	6	16
c_9	16	14
c_{10}	18	6

(B)

A. Determining The Why-not Point

In this research, the why-not question is illustrated as a situation where the user is less satisfied with the results of the $K-MPP$ query because the preferred product is not a top k-promising product. The product to be evaluated for not appearing in a $K-MPP$ result is then referred to a why-not point. Therefore, the why-not point is the user feedback which will be evaluated at the next stage.

B. Identifying k Rank of Why-not Point (nK-MPP)

The identification of the k value of the why-not point is the first step to identify the rank of a $nK-MPP$ product from the perspective of the entire product contained in the dataset. Since $K-MPP$ only displays k-products with the best market contribution value, this stage is done to evaluate the market contribution value of the why-not point to the whole product. By this step, the cause of a why-not question can be answered by providing the first informative solution in the form of ranking information of the why-not point.

The k rank of the market contribution value of the why-not point from the whole product $MC\ (C, q \mid P)$, denoted by k', can be determined by changing the value of k so that the value is equal to the market contribution value of the query point q'. Therefore, we defined $k'= rank\ (MC\ (C, q \mid P))$. k' and query results $k'-MPP$ that will be used in the next evaluation stage.

Example 1. Based on the dataset in Table I, the market contribution values in Table II is obtained after computing DSL, RSL, and probability. The three best-rated products, defined as $2-MPP$s are product p_1, p_{11}, and p_2. Only these four products will be shown in the query result of $K-MPP$. If the product manufacturer p_5 get this the result, then the question arises, why their product does not appear as a $2-MPP$ result. Therefore, as the first informative solution product p_5 rank will be checked and made as the value of k'. Based on Table II, k' = 3. After the value of k' is identified, the query result of $3-MPP$ and its MC value will also be informed.

C. Identifying the Cause

The higher the value of market contribution, the greater the chance of a product to emerge as a result of $K-MPP$. Since the market contribution value is derived from the sum of the probability values of each RSL member, the identification task of why-not question cause can be done by evaluating the RSL members of $K-MPP$ and $nK-MPP$.

Definition 1. (Cause of why-not question) For product set P, given the reverse skyline of product RSL_{p_i}, the amount of reverse skyline member $NRSL_{p_i}$, k-most promising product $K-MPP$ and non k-most promising product $nK-MPP$ sets. The cause of $p_i \in nK-MPP$ can be identified using RSL_{p_i}.
(i) if $NRSL_{p_i} < \min NRSL_{pp_j}$ where $pp_j \in K-MPP$, then the cause is the lack of $NRSL_{p_i}$.
(ii) if $NRSL_{p_i} \geq \min NRSL_{pp_j}$, then the cause is $MVC \not\ni RSL_{p_i}$or the customer's probability rank of RSL members is equal or less than the MVC of $K-MPP$ members.

TABLE II
MARKET CONTRIBUTIONS OF EACH PRODUCT IN DATASET

Rank	Product	$RSL_{(p)}$	MC
1	p_1	c_4, c_6, c_1	1,5833
	p_{11}	c_3, c_4, c_1	1,5833
2	p_2	c_8, c_{10}, c_2, c_6	1,533
3	p_5	c_{10}, c_9	1,5
	p_8	c_4, c_{10}	1,5
4	p_{13}	c_1, c_9	1,25
5	p_7	c_7, c_2, c_8	1,2
6	p_6	c_3, c_6, c_1	1,166
7	p_9	c_7, c_8	1
	p_3	c_9	1
8	p_4	c_5, c_3	0,666
9	p_{14}	c_3, c_1	0,583
10	p_{12}	c_8	0,5
	p_{10}	c_7	0,5

Based on the identification process of why-not question above, to increase the market contribution value, there are two possible solutions: (a) increase the number of RSL members, and (b) add RSL members with c which has the k-largest probability value, hereinafter referred to as the most valuable customer (MVC).

Definition 2. (Most Valuable Customer) For customer C set, given the dynamic skyline of customer DSL_{c_i} and the probability of customer c, choose product p, defined as $Pr(c, p \mid P)$. Most valuable customer consists of customer c whose probability is k-largest; MVC = k- $Pr(c, p \mid P)$, where the k value for MVC identification process is equal to the k value defined in $K - MPP$.

Example 2. Based on Table II, it can be seen that p_5 do not appear as $2 - MPP$ because $NRSL_{p_5}$ are less than $NRSL_{p_1}$, $NRSL_{p_{11}}$ and $NRSL_{p_2}$ which have a minimum number of RSLs among all $2 - MPP$ members.

Example 3. p_{13} has the number of RSL members equal to p_5 and p_8 but does not appear as $3 - MPP$ because c_{10} which have the best probability value at $3 - MVP$ is not a member of $RSL_{p_{13}}$.

D. Query Modification

The query point modification process is a data refinement approach that proposed in this research. By modifying the attribute value of record q to q', the expected output is the emergence of q' as a member of $K - MPP$.

Definition 3. Query point q modification is performed by considering the RSL members of promising product $pp_i \in K - MPP$. For the set of data dimension D of each customer $c_i \in RSL_{pp_i}$ and c_i is not the member RSL_q, the minimum data value of customer c_i in its d dimension which has the closest difference with the value of q is defined as V_{dmin}; Then, the value of q' can be determined by changing the value in the d dimension of the query point q as the V_{dmin} value.

TABLE III
MARKET CONTRIBUTIONS OF EACH PRODUCT IN DATASET

Rank	Customer	$DSL_{(c)}$	Probability
1	c_9	p_3	1
	c_4	p_8	1
2	c_7	p_{10}, p_7	0,5
	c_8	p_{12}, p_2	0,5
	c_{10}	p_5, p_{14}	0,5
3	c_5	p_{14}, p_7, p_4	0,333
	c_6	p_1, p_7, p_2	0,333
	c_3	p_3, p_6, p_{14}	0,333
4	c_1	p_{13}, p_9, p_1, p_{11}	0,25
5	c_2	p_{13}, p_9, p_1, p_{11}	0,2

Fig. 2. The Query Modification Process

Based on Definition 3, we can conclude that the data refinement process consists of three stages as shown in Figure 2. The first step that needs to be performed before the data modification task is the RSL identification of the query q and each promising $pp_i \in K - MPP$. The pre-processing task begins with the collection of a set of $C \in RSL_{pp_i}$ in Lc. Then we calculate the value difference of each customer $c_i \in Lc$ in its each dimension d_i with q. The computational results are then stored on $SdataLCq$. Because the query modification process needs to consider the least cost change possible, the next pre-processing step is sorting the $SdataLCq$ based on its minimal value of the overall data dimension d.

Example 4. Based on the dataset in Table I, the preference value of p_5 (which is the why-not $2 - MPP$) is (16.6), while the RSL of each $2 - MPP$ member and its preferred value are c_1 (10.10), c_2 (4.10), c_3 (20.13), c_4 (12.2), c_6 (2.8), c_8 (6.16) and c_{10} (18.6). The value difference of p_5 to all members of RSL 2-MPP is depicted in Table III.

After obtaining the $SdataLCq$ table in the pre-processing, the query modification is done by changing the query value q on one of its dimensions by considering the $RSL(pp_i)$ member's value which has a minimal difference in $SdataLCq$. By modifying the query based on the customer preference c_i which appears as a member of $RSL(pp_i)$, then c_i is also expected to appear as a member of $RSL\ (q')$. Modification of the query value leads to a change of probability score. Changes over probability score will also result in changes over MC score and $K - MPP$ ranking.

Example 5. Based on the results of the $SdataLCq$ in Table IV, the data point c_6 and c_{10} has the closest value with why-not point q in its d_2 and d_1 dimension. But, since c_{10} has become a member of $RSLp_5$, the V_{dmin} value chosen is 8 which is the preference value of c_6 in its d_2 dimension. The q value can then be determined by changing the value of d_2 from q as the value of V_{dmin}, so that the new query q' = (16,8) is obtained. This process will continue to be done on the entire data in the $SdataLCq$ table, so $list\_new\_data$ have obtained as in Table V. $list\_new\_data$ represents the data refinement result that may resolve the why-not point as $K - MPP$ result.

Not all members of $list\_new\_data$ can resolve the why-not question problem so that as the next step, we require a validation process. The purpose of the validation process is to compute $K - MPP$ with a new value of why-not point that has been modified as in $list\_new\_data$. Since the $list\_new\_data$ table already contains the query modification value based on the smallest data change on its one dimension, the validation process will be stopped if one valid data refinement has obtained q' result from $list\_new\_data$ which is a member of $K - MPP$. Based on

TABLE IV
$SdataLCq$ RESULT OF WHY-NOT POINT TO RSL_{pp_i} MEMBERS

Customer	Value Difference	
	d_1	d_2
c_1	6	4
c_2	12	4
c_3	4	7
c_4	4	4
c_6	14	2
c_8	10	10
c_{10}	2	4

TABLE V
DATA REFINEMENT RESULTS

# of answers	q'
1	16,8
2	16,10
3	20,8
4	16,2
5	12,8
6	16,16
7	6,6

the three stages of query modification that have been done, data refinement will be obtained with the smallest change of value, so that the cost needed is also the least cost.

V. EXPERIMENTAL RESULT

A. Experimental Data

As an experiment, this research uses three data types to be evaluated, i.e. Forest Cover type (FC) dataset, independent dataset (IND), and anti-correlated dataset (ANT). Each type of data has its variations on the amount of data n and number of data dimensions d.

Independent data (IND) is a synthetic dataset that has the distribution of random attribute values and its values are not mutually affected. The use of this data aims to test the performance of algorithms when dealing with data whose attribute values are not related to each other. The range of values used for each attribute is between 1 to 100.

Anti-correlated (ANT) data is a synthetic data set that has the opposite distribution of its attribute values, which means that the data has a high value on one of its attributes but is very low for the other attributes, along with its vice versa. The use of this data aims to test the performance of the algorithm when dealing with data whose attribute values are contradictory and has the least dominant relationship between the lowest data compared with other data types. The range of values used for each attribute is between 1 to 100.

The Forest Cover type (FC) data is a real dataset derived from the actual source. The use of this data aims to measure the performance of algorithms in the data with the distribution and range of attribute values that are interrelated.

B. Experimental Scenario

The experiment was performed on each dataset type (independent, anti-correlated, and forest cover type) with various variations for each of the available independent variables (cardinality, d, and ΔK) as follows:

a. Variations of cardinality of data: 5,000, 10,000, 20,000, 30,000, and 50,000.
b. Variations in the number of dimensions (d): 2, 3, 5, 7, and 10.
c. Variation of ΔK or the difference of why-not point ranking with K on $K - MPP$: 1, 3, 5, 7, and 10.

In addition to the above variation of variable values, there is also a fixed value which is the default value and not varied in any experiment against certain independent variables, which are:

a. The cardinality of data: 20,000.
b. The number of dimensions: 3.
c. ΔK: 3.

For example, when an experiment is performed in the first scenario, where the data cardinality variable will be an independent variable whose value varies according to predetermined scenario (5,000, 10,000, 20,000, 30,000, and 50,000), the other variable is set as its fixed value d = 3, and ΔK= 3.

There are three metrics to be analyzed. The first metric is the number of $list\_new\_data$ proposed as the data refinement suggestion, and the second metric is the successes of the data refinement approach evaluated in the query modification validation stage. The last metric is the average time needed to find the proposed data refinement with minimal cost that is able to resolve the why-not point as a member of $K - MPP$.

C. Experimental Result: Data Cardinality Variation

As depicted in Table VI, Table VII, and Table VIII, the number of variations of data refinement produced on each amount of cardinality will increase. Similarly, the validation time required to perform a final check whether the formed data refinement has been able to resolve the why-not K-MPP issue will also increase.

In addition, in Table V and Table VII, it can be concluded that the number of variations of data refinement generated on the IND and FC data types tends to be not much different, whereas a much different amount of variation resulted in the ANT data as in Table VI. The number of produced data refinement is also more constant than the other two data types. This is because of the characteristics of the distribution of data owned by each type of data. In IND and FC data, data tend to be more spread over each dimension of data, compared with ANT data.

In the overall test results that have been done, it can also be seen that the time required to determine the data refinement is relatively constant and will also experience a slight increase in the larger data amount. In the three tables below, average execution obtained a minimum value of 1.22 s and the maximum value of 1.9 s.

D. Experimental Result: Data Dimension Variation

In Table IX, Table X and Table XI the number of data refinement generated will be reduced in a larger number of data dimensions. Similarly, the validation time required to perform a final check whether the formed data refinement has been able to resolve the why-not K-MPP issue will also reduce.

In contrast to the previous scenario, in this scenario, the number of generated data refinement overall data types tends to have the same trend; as the number of data dimensions higher, the less data refinement is generated. This is due to the amount of skyline computing results that will have fewer results in a higher number of data dimensions. In addition, the time needed to determine the variation of data refinement will also decrease due to a higher number of data dimension.

TABLE VI
EXPERIMENT RESULTS OF DATA CARDINALITY VARIATIONS ON INDEPENDENT DATASET

Data Type	*d*	Data Cardinality	Δ*K*	Amount of Data Refinement	Δt (s)	Validation (s)
IND	3	5000	3	17	1.56	340
		10000		39	1.61	675
		20000		115	1.77	2100
		30000		139	1.84	2798
		50000		153	1.88	3230

TABLE VII
EXPERIMENT RESULTS OF DATA CARDINALITY VARIATIONS ON ANTI-CORRELATED DATASET

Data Type	*d*	Data Cardinality	Δ*K*	Amount of Data Refine-ment	Δt (s)	Validation (s)
ANT	3	5000	3	5	1.13	25
		10000		5	1.17	47
		20000		7	1.28	210
		30000		7	1.24	176
		50000		9	1.33	328

TABLE VIII
EXPERIMENT RESULTS OF DATA CARDINALITY VARIATIONS ON FOREST COVER TYPE DATASET

Data Type	*d*	Data Cardinality	Δ*K*	Amount of Data Refinement	Δt (s)	Validation (s)
FC	3	5000	3	15	1.56	322
		10000		39	1.78	710
		20000		118	1.64	1988
		30000		134	1.69	2679
		50000		148	1.9	3123

TABLE IX
EXPERIMENT RESULTS OF DATA DIMENSION VARIATIONS ON INDEPENDENT DATASET

Data Type	*d*	Data Cardinality	Δ*K*	Amount of Data Refinement	Δt (s)	Validation (s)
IND	2	20000	3	172	2.94	3782
	3			112	1.56	1986
	5			98	1.32	1876
	7			87	1.32	1479
	10			71	1.33	1283

TABLE X
EXPERIMENT RESULTS OF DATA DIMENSION VARIATIONS ON ANTI-CORRELATED DATASET

Data Type	*d*	Data Cardinality	Δ*K*	Amount of Data Refine-ment	Δt (s)	Validation (s)
ANT	2	20000	3	166	1.87	3457
	3			104	1.28	1894
	5			97	1.27	998
	7			89	1.22	831
	10			78	1.2	819

TABLE XI
EXPERIMENT RESULTS OF DATA DIMENSION VARIATIONS ON FOREST COVER TYPE DATASET

Data Type	*d*	Data Cardinality	Δ*K*	Amount of Data Refine-ment	Δt (s)	Validation (s)
FC	2	20000	3	169	1.54	3566
	3			103	1.52	1992
	5			95	1.44	1772
	7			83	1.42	1362
	10			68	1.37	1003

E. Experimental Result: ΔK Variation

Similar to the outcome of the first scenario, this scenario results are depicted in Table XII, Table XIII, and Table XIV. The amount of generated data refinement and validation time will increase in the higher number of ΔK. The number of variations of data refinement generated on IND and FC data types also tends to be similar, whereas in the ANT data as in Table XII have a relatively constant amount compared to the other two data types. This condition happens because of the distribution of data characteristics for each type of data. In IND and FC, data distribution tends to be more spread over each dimension.

In the overall results of the evaluation, it can also be seen that the time needed to determine the data refinement is relatively constant and will also experience a slight increase in the greater ΔK.

VI. CONCLUSION

Why-not question that appears in the given $K - MPP$ query result, can be identified and resolved by evaluating the RSL member of the why-not point or product that is not a member of K-MPP and RSL of K- MPP members. If the number of RSLs from the why-not point is less than the minimum number of RSLs from K-MPP members, the cause of the product is not a member of K-MPP is the lack of RSL members. Conversely, if the number of RSLs of the why-not point equals or exceeds the minimum RSL number of K-MPP members, then the cause is the absence of Most Valuable Customer (MVC) on RSL members.

The query point modification process is a data refinement approach which proposed in this research. By modifying the q value to q', the expected output is the appearance of q' as a member of K-MPP. Query point modification q can be done by considering RSL members of K-MPP. The minimum data value of customer c_i in its d dimension which has the closest difference with the value of q is defined as V_{dmin}; Then, the value of q' can be determined by changing the value in the d dimension of the query point q as the V_{dmin} value.

Data refinement approach is evaluated under three different scenarios (variation of data cardinality, variation of data dimension, and ΔK variation), the results obtained that: (a) The time required to find variation of data

TABLE XII
EXPERIMENT RESULTS OF ΔK VARIATIONS ON INDEPENDENT DATASET

Data Type	*d*	Data Cardinality	ΔK	Amount of Data Refine-ment	Δt (s)	Validation (s)
IND	3	20000	1	57	2.31	892
			3	113	2.99	1562
			5	268	3.11	3862
			7	563	3.43	6610
			10	665	3.48	9234

TABLE XIII
EXPERIMENT RESULTS OF ΔK VARIATIONS ON ANTI-CORRELATED DATASET

Data Type	*d*	Data Cardinality	ΔK	Amount of Data Refine-ment	Δt (s)	Validation (s)
ANT	3	20000	1	5	2.31	15
			3	7	2.99	11
			5	11	3.11	25
			7	21	3.43	38
			10	33	3.48	49

TABLE XIV
EXPERIMENT RESULTS OF ΔK VARIATIONS ON FOREST COVER TYPE DATASET

Data Type	*d*	Data Cardinality	ΔK	Amount of Data Refine-ment	Δt (s)	Validation (s)
FC	3	20000	1	68	2.31	1325
			3	100	2.99	1897
			5	246	3.11	3348
			7	448	3.43	4779
			10	579	3.48	5691

refinement tends to be constant and will increase in data cardinality and (b) In the variation of data dimensions, the time required to find data refinement will be faster in higher number of d because the number of generated data refinement is fewer than the less one. (c) The validation process still requires a long time and its value will be higher in the large number of data cardinality and a large number of ΔK but will be decreased in the higher number of data dimensions.

REFERENCES AND FOOTNOTES

[1] Jagadish, H. V., "Making database systems usable", in SIGMOD, 2007
[2] Islam, M. S. and Liu, C. (2016). Know your customer: computing k-most promising products for targeted marketing. *The VLDB Journal.*
[3] Bartolini, I., Ciaccia, P. and Patella, M. (2006). Salsa: computing the skyline without scanning the whole sky. *CIKM*, pp. 405-414.
[4] Liu, Q. et al. (2016). Answering why-not and why questions on reverse top-k queries. *VLDB Journal*. 25(Research Collection School Of Information Systems), pp. 867-892.
[5] Islam, M. S., "On Answering Why and Why-not Questions in Databases", in: ICDE, 2013, pp. 973-984.
[6] Islam, M. S., Zhou, R. and Liu, C, "On Answering Why-not Questions in Reverse Skyline Queries", in IEEE International Conference on Data Engineering (ICDE), 2013.
[7] Liu, Q. et al. (2016). Answering why-not and why questions on reverse top-k queries. *VLDB Journal*. 25(Research Collection School Of Information Systems), pp. 867-892.
[8] Chapman, A. and Jagadish, H.V., "Why not?", in SIGMOD, 2009, pp. 523– 534.
[9] Bidoit, N., Herschel, M. and Tzompanaki, K., "Query-based why-not provenance with nedexplain", in EDBT, 2014, pp. 145–156.
[10] Huang, J., Chen, T., Doan, A.H. and Naughton, J.F. (2008). On the provenance of non-answers to queries over extracted data. *VLDB Journal*, pp. 736–747.
[11] Zong, C., Yang, X.,Wang, B. and Zhang, J., "Minimizing explanations for missing answers to queries on databases", in DASFAA, 2013, pp. 254–268.
[12] Herschel, M. and Hernandez, M. (2010). Explaining missing answers to spjua queries. *VLDB Journal*, pp. 185–196.
[13] Herschel, M., Hernandez, M.A. and Tan,W.C. (2009). Artemis: A system for analyzing missing answers. *VLDB Journal*, pp. 1550–1553.
[14] He, Z. and Lo, E. "Answering why-not questions on top-k queries", in ICDE, 2012, pp. 750–761.
[15] He, Z. and Lo, E. "Answering why-not questions on top-k queries", in IEEE Trans. Knowl. Data Eng. 26(6), 2014, pp. 1300–1315.