A Performance Evaluation of Preference Evaluation Techniques in Real High Dimensional Database

Ali A. Alwan, Hamidah Ibrahim, Tan C. Yip, Nur Izura Udzir, Fatimah Sidi

Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia

Abstract

Preference query has received high interest due to its great benefits over various types of database applications. This type of query provides more flexible query operators that retrieve data items which are not dominated by the other data items in all attributes (dimensions). Many preference techniques for preference queries have been introduced including top-$k$, skyline, multi-objective skyline, top-$k$ dominating, $k$-dominance, ranked skyline, and $k$-frequency. All of these preference techniques aimed at finding the “best” result that meets the user preferences. This paper aims at evaluating the performance of the five well-known preference evaluation techniques, namely: top-$k$, skyline, top-$k$ dominating, $k$-dominance and $k$-frequency; in a real database application when high number of dimensions is the main concern. To achieve this, a recipe searching application with maximum number of 60 dimensions has been developed which assists users to identify the most desired recipes that fulfill their preferences. Several analyses have been carried out, where execution time is the main measurement used to evaluate each preference technique.

Keywords: Preference queries; preference evaluation techniques; skyline; top-$k$; query processing

1. Introduction

Designing and developing database management systems that provide solutions that best meet the user preferences has achieved great attention. The preference queries which incorporate special query operators to limit the set of results which satisfy the preferences of users have been widely introduced in many real life application domain such as multi-criteria decision making applications [1-4], decision support systems, e-commerce, personal preference web services such as hotel recommender [5] and restaurant finder [6]; and peer-to-peer network [7]. Due to this, many preference techniques have been proposed including but not limited to top-$k$ [8], skyline [9], $k$-dominance [3], top-$k$ dominating [2], and $k$-frequency [1]. The main concern of these preference techniques is to minimize the searching space and

* Corresponding author. Tel.: +60-3-89471713; fax: +60-3-89466572.
E-mail address: hamidah@fsktm.upm.edu.my
This paper is a refinement of our previous article [10] which attempts to examine the well-known preference evaluation techniques in the database systems, namely: top-k, skyline, top-k dominating, k-dominance, and k-frequency when a huge number of dimensions is to be considered. The evaluation should be carried out on real database application and hence we have purposely developed a recipe searching application which offers a variety of recipes that meets the ever-changing demands of user. The difference between this article and our previous article [10] is that in this article we further evaluate the performance of the above five preference techniques with respect to execution time when various number of results (with specific number of dimensions) and various number of dimensions and result sizes are considered.

The reasons for choosing the recipe domain to evaluate the performance of the preference techniques are mainly due to: (i) each recipe normally consists of several components like ingredients, course types, cuisine types, cooking method, occasions, diet and others while the requirements of the end user are multi-dimensional and cannot be easily expressed on discrete scales. In this paper 60 dimensions have been identified. (ii) The main critical issue is a recipe component ratio which is defined by what is known, as the “best” recipe for user. To tackle this, the preference technique that considers the ratio and ranks the results according to the user needs is the best technique to be used and evaluated.

This paper is organized as follows. In Section 2, the previous works related to this work is presented and discussed. The recipe searching application is introduced in Section 3. Performance analysis is explained and discussed in Section 4. Conclusions are presented in the final section, 5.

2. Related Works

Various types of preference techniques of preference queries have been introduced in the database literature which include but not limited to: top-k, skyline, k-dominant skyline, skycube, k-frequency, top-k dominating, sort-filter-skyline (SFS), ranked skyline, sort and limit skyline algorithm (SaLSa), linear elimination sort for skyline (LESS), and Z-Sky. In the following, the preference techniques used in our work are elaborated. Further detail explanations on these techniques can be found in [1-3, 8-9].

**Top-K:** Top-k technique retrieves a set of selected data items (k) that dominates the other data items based on the scoring value of the monotonic preference ranking function F. The basic concept of this technique is to give score (weight) to each data item in the database. Hence, in order to compute the scoring results a preference ranking function (monotone function) is used to accumulate the values of dimensions for each data item. Based on the final results of the preference ranking function, the k-data items with the best scores are considered the preferred data items [8, 11, 12]. Several algorithms have been proposed based on the top-k preference technique such as Onion [13], PREFER [14], Top-k Monitoring Algorithm (TMA) [15], SPEERTO [11], and Skyband Top-k Monitoring Algorithm (SMA) [15]. However, these algorithms are being evaluated on small scale of dimensions within the range 2-7.

**Skyline:** The skyline preference technique identifies the set of data items, S, in a way such that they are not dominated by the other data items in the dataset. In other words, a data item p is preferred over another data item q if and only if p is as good as q strictly in at least one dimension and in all other n dimensions [1, 2, 3, 6, 9, 16]. Several algorithms have been proposed based on the skyline preference technique such as Block-Nested-Loop (BNL) [9], Divide-and-Conquer (DC) [9], Linear Elimination Sort for Skyline (LESS) [17], Branch-Bound-Skyline (BBS) [18], SkyCube [9], and Sort and Limit Skyline algorithm (SaLSa) [19] but these algorithms are being evaluated on small scale of dimensions within the range 2-10.

**Top-K Dominating:** Top-k dominating technique identifies the set of data items k which are dominating the largest number of data items in the dataset. That means data item p is preferred over another data item q if and only if the domination power of p is greater than the domination power of q. The value of
domination power of data item \( p \) comes from the total number of data items in the dataset that is dominated by \( p \). Top-\( k \) dominating technique is a very significant tool for multi-criteria application such as decision making system and decision support system, since it identifies the most significant data items in an intuitive way [3].

**K-Dominance:** \( K \)-dominance technique prefers one data item \( p \) over another data item \( q \) in the dataset if and only if \( p \) is as good as \( q \) strictly in at least one \( k \)-dimension and in the subset of \( k \) dimensions where \( k \) is less than the total number of dimensions. Generally, the result size of \( k \)-dominance skyline is less than the result size of conventional skyline, particularly when the considered dimensions are few. However, \( k \)-dominance has some similar characteristics with skyline especially when \( k = d \) where \( d \) is the total number of dimensions in the dataset.

**K-Frequency:** \( K \)-frequency technique retrieves a set of data items \( D' \) from the given dataset \( D \) in a space \( S \), where a data item \( p \) in \( D' \) has the smallest dominating score, denoted as \( S(p) \), which represents the number of available sub-dimensions where \( p \) is not a skyline. \( K \)-frequency has many common characteristics with skyline such as transitivity property is preserved and the \( k \)-frequency queries' answers are obtained by merely comparing the actual values of each identical dimension in two different data items. Furthermore, this technique can be applied in the full space and subspace dataset. However, \( k \)-frequency needs a powerful data structure that saves the dominated sub-dimensions for every data item \( p \) in order to precisely determine the score of every data item \( p \) [1].

### 3. The Recipe Searching Application

The proposed recipe searching application has been successfully developed using PHP web programming language and SQL server. Each preference technique has been developed and tested with respect to different type of recipes. We have identified six elements which are important in searching and later selecting a particular recipe. These elements are type of ingredients, courses, cooking methods, occasions, diet restrictions, and type of cuisines. Each element has its own set of dimensions that can be selected. For instance the main ingredient element consists of 16 dimensions which represent the types of ingredient which include chicken, fish, cheese, beef, etc. Similarly, the diet element consists of 8 dimensions which represent the diet restrictions such as vegetarian, low fat, diabetic, low cholesterol, etc. All together there are 60 dimensions. A range of 0-5 is prepared for each dimension which indicates the degree of interest of a user towards a particular dimension. The smallest scale, 0, denotes no interest at all while the scale 5 denotes the highest preferences. Table 1 summarizes these dimensions. We use the notation \( d_i \) to denote the \( i \)th dimension.

Table 1. Dimensions of the recipe searching application

<table>
<thead>
<tr>
<th>Element</th>
<th>Number of dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Ingredient</td>
<td>16 ((d_1 - d_{16}))</td>
</tr>
<tr>
<td>Course</td>
<td>12 ((d_{17} - d_{28}))</td>
</tr>
<tr>
<td>Cooking Method</td>
<td>8 ((d_{29} - d_{36}))</td>
</tr>
<tr>
<td>Occasion</td>
<td>8 ((d_{37} - d_{44}))</td>
</tr>
<tr>
<td>Diet</td>
<td>8 ((d_{45} - d_{52}))</td>
</tr>
<tr>
<td>Cuisine</td>
<td>8 ((d_{53} - d_{60}))</td>
</tr>
</tbody>
</table>

The application provides several features for the user before a particular recipe is selected. These features include (i) users can select the preference technique they prefer; (ii) users are free to omit any dimensions that are not interest to them. By default all dimensions are assigned the value 0; and (iii) users may rank the dimensions according to their needs by manipulating the scale to be assigned to the needed dimensions. For example, Table 2 represents a sample of query submitted by a user.
Table 2. Example of dimensions selected in a user query

<table>
<thead>
<tr>
<th>Element</th>
<th>Dimensions selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Ingredient</td>
<td>$d_1 = 5$; $d_2 = 3$</td>
</tr>
<tr>
<td>Course</td>
<td>$d_{18} = 4$</td>
</tr>
<tr>
<td>Cooking Method</td>
<td>$d_{29} = 4$</td>
</tr>
<tr>
<td>Occasion</td>
<td>$d_{43} = 5$</td>
</tr>
<tr>
<td>Diet</td>
<td>$d_{46} = 4$</td>
</tr>
<tr>
<td>Cuisine</td>
<td>$d_{54} = 5$</td>
</tr>
</tbody>
</table>

Note: $d_1$ (chicken); $d_2$ (rice); $d_{18}$ (dinner); $d_{29}$ (baking); $d_{43}$ (Christmas); $d_{46}$ (healthy); $d_{54}$ (Italian)

After selecting the appropriate dimensions by giving a suitable preference value, then user is required to specify the type of preference technique before the application retrieves the recipes. For the purpose of this paper, 100 recipes have been collected and saved in a database called the Recipe Database ($RDb$). Several steps are initially achieved before the preference techniques are being applied. These steps mainly aim at removing the irrelevant data items (records) from the Recipe Database from being considered in the searching process as they will not contribute to the final result. The steps are discussed below:

1. Each recipe from the $RDb$ is mapped into a two dimensional array, $RA$, with the following format:

   Structure of $RA$

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>$Id$</td>
<td>$d_1$</td>
<td>$d_2$</td>
<td>$d_3$</td>
<td>...</td>
<td>$d_{60}$</td>
</tr>
</tbody>
</table>

   Where $Id$ is the identifier of the recipe and $d_i$ is a score given to the $i$th dimension. We use the notation $r_kd_i$ to denote the $i$th dimension of the $k$th recipe. An example of a recipe stored in the array is as follow:

   An instance of $RA$

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>101</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>...</td>
<td>5</td>
</tr>
</tbody>
</table>

   The above is an information about the recipe 101 which uses chicken ($d_1$) as the main ingredient, vegetable ($d_3$), ..., and South-western ($d_{60}$) is the main cuisine.

2. Given a query, $Q$, with a set of $n$ selected dimensions, $dq = \{d_{q1}, d_{q2}, ..., d_{qn}\}$ only those recipes in the $RA$ that matched with these dimensions are selected and stored in a temporary array, $TRA$. The following definition defined the match criteria.

   **Definition 1:** A recipe $r_k$ is said to be matched to a query $Q$ if $\exists d_{qi} \in dq$, $\exists d_j \in r_k$ and $r_kd_j > 0$ where $j$ is the equivalent dimension as $i$.

   This gives the following definition which defined the unmatched criteria.

   **Definition 2:** A recipe $r_k$ is said to be unmatched to a query $Q$ if $\forall d_{qi} \in dq$, $\exists d_j \in r_k$ and $r_kd_j = 0$ where $j$ is the equivalent dimension as $i$.

   The following example clarifies this point. Consider the query given in Table 2 and the following instances of $RA$.

   User query

<table>
<thead>
<tr>
<th>Index</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>...</th>
<th>$d_{18}$</th>
<th>...</th>
<th>$d_{29}$</th>
<th>...</th>
<th>$d_{43}$</th>
<th>...</th>
<th>$d_{46}$</th>
<th>...</th>
<th>$d_{54}$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>5</td>
<td>3</td>
<td>...</td>
<td>4</td>
<td>...</td>
<td>4</td>
<td>...</td>
<td>5</td>
<td>...</td>
<td>4</td>
<td>...</td>
<td>5</td>
<td>...</td>
</tr>
</tbody>
</table>

   Note: The other dimensions have the value 0.

   Instances of $RA$

<table>
<thead>
<tr>
<th>Index</th>
<th>$Id$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>...</th>
<th>$d_{18}$</th>
<th>...</th>
<th>$d_{29}$</th>
<th>...</th>
<th>$d_{43}$</th>
<th>...</th>
<th>$d_{46}$</th>
<th>...</th>
<th>$d_{54}$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>102</td>
<td>5</td>
<td>5</td>
<td>...</td>
<td>5</td>
<td>...</td>
<td>5</td>
<td>...</td>
<td>5</td>
<td>...</td>
<td>5</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>103</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>0</td>
<td>5</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Note: The other dimensions that are not listed in the table might have value 0, 1, 2, 3, 4, or 5, while for 103 we assume that all values are zero.

   From the above instances of $RA$, recipe $r_{102}$ and $r_{110}$ satisfied the Definition 1 and are selected while $r_{103}$ is omitted as for all the dimensions requested by the user their values = 0 (satisfied the Definition 2).
3. Those dimensions in the temporary array, TRA, which are not considered in the query, Q, are then eliminated to reduce the size of dimensions to be considered. Based on the example given in Step 2 above, the following is the result of Step 3.

<table>
<thead>
<tr>
<th>Index</th>
<th>1d</th>
<th>d1</th>
<th>d2</th>
<th>d18</th>
<th>d29</th>
<th>d43</th>
<th>d46</th>
<th>d54</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>102</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4. The preference evaluation techniques are then applied towards the recipes that have been saved in the TRA. The algorithms for each of the evaluation technique as used in our application are given in the Fig. 1.

**Skyline Algorithm**

**Input:** A set of recipes, \( R = \{r_1, r_2, ..., r_n\} \)

**Output:** A set of results, \( S \)

BEGIN

FOR each \( r_k \in R \) DO

FOR each \( r_l \in R \) where \( k \neq l \) DO

BEGIN

IF \( r_l \) dominates \( r_k \) THEN

\( R = R \cup \{r_k\} \)

ELSE IF \( r_k \) dominates \( r_l \) THEN

\( R = R \cup \{r_l\} \)

END

END

\( S = S \cup R \)

END

**K-dominance Algorithm**

**Input:** A set of recipes, \( R = \{r_1, r_2, ..., r_n\} \), a set of preferred dimensions \( dq = \{dq_1, dq_2, ..., dq_k\} \)

**Output:** A set of results, \( S \)

BEGIN

FOR each \( r_k \in R \) DO

FOR each \( r_l \in R \) where \( k \neq l \) DO

BEGIN

IF \( r_l \) dominates \( r_k \) w.r.t \( dq \) THEN

\( R = R \cup \{r_k\} \)

ELSE IF \( r_k \) dominates \( r_l \) w.r.t \( dq \) THEN

\( R = R \cup \{r_l\} \)

END

END

\( S = S \cup R \)

END

**Top-k Algorithm**

**Input:** A set of recipes, \( R = \{r_1, r_2, ..., r_n\} \), a ranking function \( F = \sum_{l=1}^{k} d_{l} \)

**Output:** A set of results, \( S \)

BEGIN

FOR each \( r_k \in R \) DO

FOR each \( r_l \in R \) where \( k \neq l \) DO

BEGIN

IF \( F(r_l) > F(r_k) \) THEN

\( R = R \cup \{r_k\} \)

ELSE IF \( F(r_k) > F(r_l) \) THEN

\( R = R \cup \{r_l\} \)

END

END

\( S = S \cup R \)

END

**Top-k Dominating Algorithm**

**Input:** A set of recipes, \( R = \{r_1, r_2, ..., r_n\} \), a dominating power function \( F(r_i) = \sum_{l=1}^{k} d_{l} \) i.e. the number of data items dominated by \( r_i \)

**Output:** A set of results, \( S \)

BEGIN

FOR each \( r_k \in R \) DO

FOR each \( r_l \in R \) where \( k \neq l \) DO

BEGIN

IF \( F(r_l) > F(r_k) \) THEN

\( R = R \cup \{r_k\} \)

ELSE IF \( F(r_k) > F(r_l) \) THEN

\( R = R \cup \{r_l\} \)

END

END

\( S = S \cup R \)

END

**K-frequency Algorithm**

**Input:** A set of recipes, \( R = \{r_1, r_2, ..., r_n\} \), a dominating score, \( F(r_i) \)

**Output:** A set of results, \( S \)

BEGIN

FOR each \( r_k \in R \) DO

FOR each \( r_l \in R \) where \( k \neq l \) DO

BEGIN

IF \( F(r_l) < F(r_k) \) THEN

\( R = R \cup \{r_k\} \)

ELSE IF \( F(r_k) < F(r_l) \) THEN

\( R = R \cup \{r_l\} \)

END

END

\( S = S \cup R \)

END

Fig. 1. The preference evaluation techniques

4. Performance Evaluation

We have carried out three analyses. The first analysis aims at analyzing the performance of the preference techniques with respect to the total number of dimensions that represents the user’s preferences. In this paper we vary the number of dimensions from 10 – 60 dimensions, while the size of the recipe database is fixed. Furthermore, in this analysis we also evaluate the preference techniques with respect to the size of the recipe database while the number of dimensions is fixed during the process of searching the best recipes that meet the user’s request. The results reported for this analysis also appeared in [10]. The second analysis focuses on evaluating the performance of preference techniques with respect to the size of the results. Finally, analysis three aims at comparing the processing time of the preference techniques by varying the result size and fixed the number of dimensions. In this paper we focused
exclusively on the number of dimensions, the size of databases, and the size of results as they significantly impact on the process of finding preference query answer.

4.1 Results of Analysis 1

Fig. 2(a) demonstrates the results when various numbers of dimensions with fixed number of data items (recipes), which is 100, are considered. The initial number of dimensions is 10 and it is incrementally increased by 10, until the number of dimensions reached 60, which is the maximum number of dimensions considered in this paper. All together there are 6 experiments that have been accomplished whereby in each experiment the number of dimensions considered is different. For each experiment 10 queries have been randomly generated where each query selects the appropriate number of dimensions (see Step 2 of Section 3). The execution time of each query is measured when Step 4 as described in Section 3 is executed. Averaging the execution time of these 10 queries gives the final execution time of the experiment. Thus, six different sets of queries have been designed for this analysis.

Similarly, Fig. 2(b) depicts the results of applying different number of recipes which reflects the size of database with fixed number of dimensions, which is 10. The initial number of recipes is 10 and it is incrementally increased by 10, until the number of recipes reached 100, which is the maximum number of recipes considered in this analysis.

From the figures, it is obvious that the top-k technique has the lowest amount of execution time in all cases compared to the other techniques in terms of the number of dimensions and the database size. This is due to the fact that most of the process in finding the best query answer is executed without needing to compare the individual dimensions at the data item level to determine the query results. i.e. it accumulates the values of all dimensions as a single value. However, k-dominance, k-frequency and skyline techniques achieved almost the same amount of increment in the execution time when the number of dimensions and the size of database is increased. However, top-k dominating has the worst performance in all cases with respect to the number of dimensions and the database size compared to the other techniques.

![Fig. 2. The amount of execution time](image)

(a) Number of dimensions  (b) Database size

4.2 Result of Analysis 2

Fig. 3 illustrates the results of each preference technique by varying the result size. The initial size of result is set to 5 and it is incrementally increased by 5, until the size of the result reached up to 25, which is the maximum number of results (recipes) considered in this analysis. All together there are 5 experiments that have been conducted whereby in each experiment the number of recipes considered is fixed to 100 recipes. For each experiment 5 queries have been randomly generated where each query
selects 60 dimensions (see Step 2 of Section 3). The execution time of each query is measured when Step 4 as described in Section 3 is executed. From the figure, it is clear that the top-k technique is the superior technique in most of the cases compared to the other four techniques. This is due to the fact that most of the process in finding the best query answer is performed without needing to compare the individual dimensions at the data item level to determine the query results, i.e. it accumulates the values of all dimensions as a single value. However, k-dominance and skyline techniques achieved almost the same amount of increment in the execution time when the size of results is increased. However, k-frequency has the worst performance compared to the other techniques.

**Fig. 3. The amount of execution time with respect to the result size**

### 4.3 Result of Analysis 3

Fig. 4 depicts the results of varying the result size (number of recipes) and fixed the number of dimensions. The initial size of the result is 5 and it is incrementally increased by 5, until the number of results reached 20. Furthermore, the initial number of dimensions is 20 and it is increased by 20, until the number of dimensions becomes 60, which is the maximum number of dimensions considered in this analysis. All together there are 12 experiments that have been carried out whereby in each experiment the result size considered is different. For each experiment 5 queries have been randomly generated where each query selects 20, 40 or 60 dimensions (see Step 2 of Section 3). The execution time of each query is measured when Step 4 as described in Section 3 is executed.

**Fig. 4. The amount of execution time with respect to the number of dimensions and the result size**
From the figure, it is obvious that the top-k technique outperforms the other four techniques in all cases. This is due to the fact that most of the process in finding the best query answer is performed without needing to compare the individual dimensions at the data item level to determine the query results, i.e. it accumulates the values of all dimensions as a single value. However, k-dominance, k-frequency and skyline techniques achieved almost the same amount of execution time in most cases when the number of dimensions is 20. However, top-k dominating and k-frequency techniques performed worst compared to the other techniques when the numbers of dimensions are 40 and 60.

5. Conclusion

In this paper we have presented and discussed a recipe searching application which has been developed with the aim to evaluate the various types of preference evaluation techniques for preference queries. Three analyses with different aims have been accomplished by considering various numbers of dimensions, database sizes, and result sizes. These are the most significant factors that impact the execution time of the preference evaluation techniques in searching for the “best” query answer that meets the users’ preferences. We have also shown that the best preference technique in term of execution time is top-k, while the worst is top-k dominating through our developed recipe searching application which represents a real high dimensional database.

6. References