OLAP4R: A Top-K Recommendation System for OLAP Sessions

Youwei Yuan[1, 2], Weixin Chen[1], Guangjie Han[3*], Gangyong Jia[1]
[1] School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou, 310018, China
[2] Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education
Hangzhou, 310018, China
[3] Department of Information and Communication Systems, Hohai University
Changzhou 213022, China,
[e-mail: hanguangjie@gmail.com]
*Corresponding author: Guangjie Han

Received February 22, 2017; revised March 24, 2017; accepted April 2, 2017; published June 30, 2017

Abstract

The Top-K query is currently played a key role in a wide range of road network, decision making and quantitative financial research. In this paper, a Top-K recommendation algorithm is proposed to solve the cold-start problem and a tag generating method is put forward to enhance the semantic understanding of the OLAP session. In addition, a recommendation system for OLAP sessions called “OLAP4R” is designed using collaborative filtering technique aiming at guiding the user to find the ultimate goals by interactive queries. OLAP4R utilizes a mixed system architecture consisting of multiple functional modules, which have a high extension capability to support additional functions. This system structure allows the user to configure multi-dimensional hierarchies and desirable measures to analyze the specific requirement and gives recommendations with forthright responses. Experimental results show that our method has raised 20% recall of the recommendations comparing the traditional collaborative filtering and a visualization tag of the recommended sessions will be provided with modified changes for the user to understand.

Keywords: On-Line Analytical Processing (OLAP), recommend system, date mining, big data, label generate

The work was supported by Natural Science Foundation of Zhejiang Province (No. Y17E050119) and also supported by a grant from the Research Center of Information Technology & Economic and Social Development in Zhejiang Province (No.15XXHJD04).

https://doi.org/10.3837/tiis.2017.06.009 **ISSN : 1976-7277**

1. Introduction

OLAP (On-Line Analytical Processing), which is used for analysis of multi-dimensional models in data warehouses provides a better support for decision making in the business information sector. Usually users conduct a sequence of OLAP queries called OLAP session to analyze the results for the specific goals. Nevertheless, numerous OLAP operations (e.g. drill down, slice, roll up and pivot) may confuse the user so it was necessary to design a recommendation system for OLAP sessions that cater to the amateur in order to ease the difficulty in data analysis [1-4].

Previous works have been conducted on similarity calculation among OLAP queries to recommend a single query or session using collaborative filtering techniques. J. Wei proposed a collaborative filtering and deep learning based recommendation system for cold start items [5]. L.Zhu et.al put forward a semantical pattern and preference-aware service mining method for personalized point of interest recommendation [6]. To acquire a better recommendation, a collaborative filtering approach is put forward to recommend the next query [7-8]. S. Rizzi et al. designed a realistic OLAP workloads tool for similarity calculation which can be used as a benchmark for OLAP recommendation. Moreover, OLAP sessions are treated as the first-class citizens in recommendations by J. Aligon et al. based on the benchmark tool mentioned above [9-10].

However, it is widely accepted that the collaborative filtering approach suffers from sparsity and cold start problems, the recent work can not well solve this problem in the OLAP session recommendation, which will affect the final effectiveness of recommendation. Morever, because of the speciality in OLAP field, the recommended OLAP session without a clear insight is also an obstruction to the newcomer [11-12].

Based on the problems above, it is quite natural that the Top-K recommendation can be introduced in OLAP sessions since this approach can find the latent valuable target which can well raise the effectiveness of recommendations [13-14]. In addition, a kind of database-based systems is introduced to recommend more useful information to the user in social tagging system. This system allows users to label items with specific meanings, called user-defined tags, which can reflect the preferences of users and evaluations on items.

In this paper, we investigate cold-start problem of recommending an OLAP session to the newcomer for the purpose of predicting his latent preference, and realize a Top-K recommendation algorithm to cope with this problem. Moreover, the speciality issue in OLAP field is studied in order to the better comprehension of the recommended OLAP session, a tag-aware approach is presented to help beginner to understand the meaning of the recommended session.

To sum up, the main contributions of this paper are listed as follows: (i). It is the first time we introduce Top-K recommendation for OLAP sessions to improve the diversity and recall of the result. (ii). A tag-aware approach with modified changes was proposed to label the OLAP session to help acquire a better understanding for the user. (iii). A prototype system was designed for OLAP recommendation with a user-friendly interface integrated with a big data analysis framework.

The remainder of this paper is organized as follows: Section 2 gives the design of the proposed system architecture along with the definition and modeling of OLAP recommendation. Section 3 presents the implementation of the Top-K recommendation and

the tag generation algorithm. Section 4 illustrates the prototype system and experimental evaluation for OLAP recommendation while Section 5 makes a summary of this paper and outlines the future.

2. Designing and Modeling

2.1 The proposed system architecture

Fig. 1 shows the models of the proposed system architecture, which are User Interface, OLAP4R core architecture, OLAP Workload generator, Data ETL tool, Big Cube System and Parameterized Data Schema, respectively. The function of each models are:

- The User Interface provides a friendly view for OLAP recommendation which can help the user to gain a correct insight from a previously-known tag.
- The OLAP4R core architecture is the key component of the whole system, which has four features: the vectorial coordinate is constructed by parameterized queries according to the multi-dimensional expressions (MDX) [9]. The similarity calculation method is implemented for OLAP sessions using the Smith-Waterman algorithm. The Top-K recommendation part is designed for OLAP sessions based on the fitting algorithm. The label generator is used for semantic understanding with modified changes.
- The OLAP Workload generator is an open source tool that is used for OLAP sessions generation based on a series of data parameters provided by the user.
- The Data ETL tool is implemented by us which is designed for massive data extraction to the big data platform in order to have an effective query analysis.
- The Big Cube System is built using big data techniques such as Hadoop, HBase, Hive and Kylin, which are both used for multi-dimensional analysis by building data cubes
- The Parameterized Data Schema contains user-defined configuration files which mainly include table schema, dimension value list, measure information and so on.

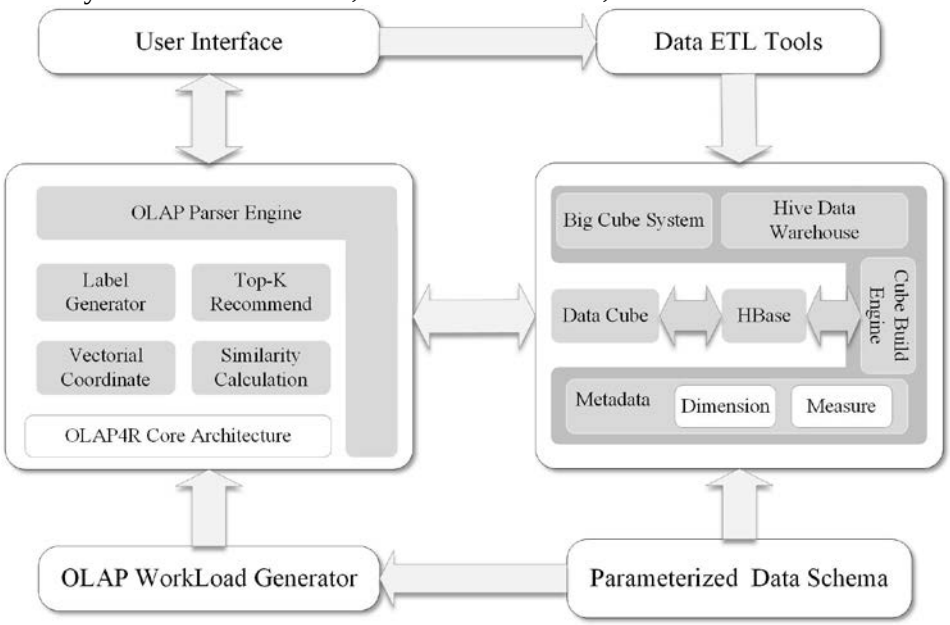


Fig. 1. System architecture of OLAP4R

The goal of the system is to recommend Top-K OLAP sessions with a well-known understanding, which is mainly implemented in the OLAP4R core architecture whose model is depicted in **Fig. 2** in the form of a UML class diagram.

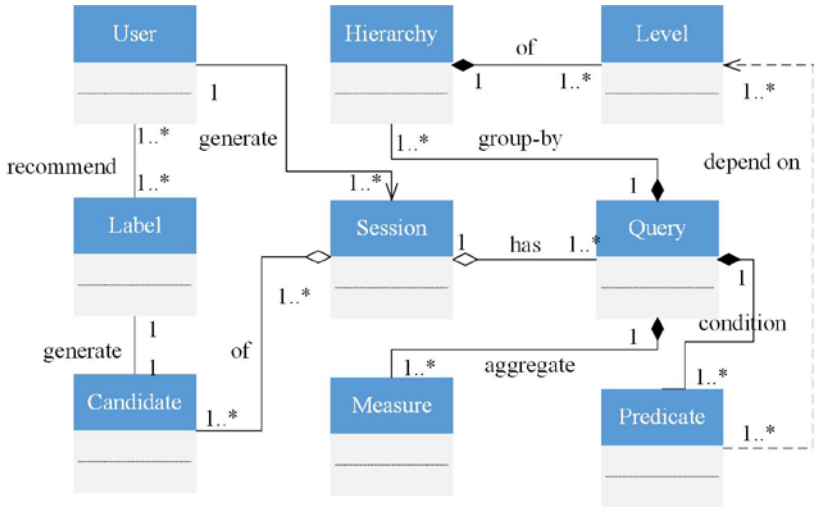


Fig. 2. The UML model of OLAP4R

The whole process of OLAP session recommendation is depicted in **Fig. 3**, we should first input the basic user information in order to generate the user preference model and the user log, and then the user preference model will be built. Besides, the user log contains three kinds of files, which are used for generating the OLAP sessions. Moreover, the similarity calculation method among sessions is implemented by our improved Smith-Waterman algorithm, which is the basis of the Top-K recommend algorithm and tag-aware method is used for semantic understanding in the final recommendation.

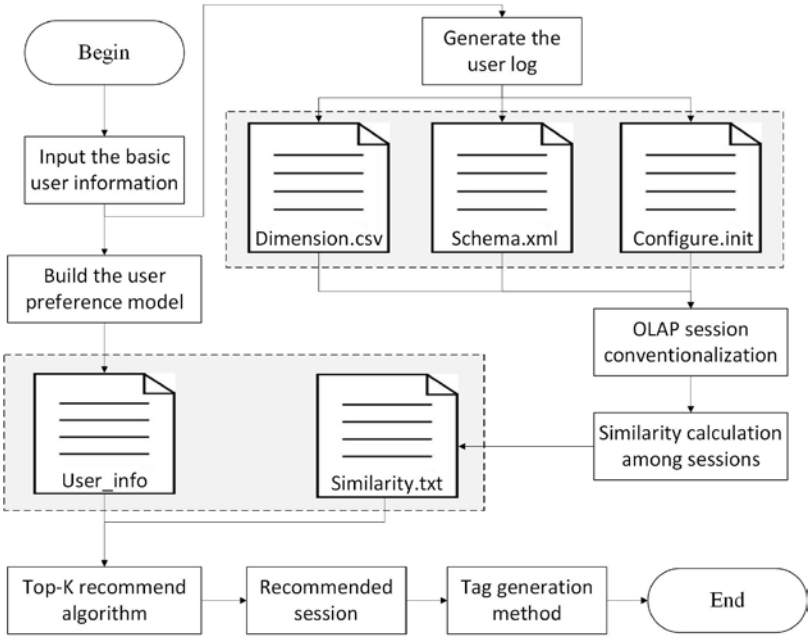


Fig. 3. Workflow of the whole process of OLAP session recommendattion

2.2 Definitions and examples about OLAP

In this section, preliminary knowledge is discussed, to introduce the OLAP related concept, and some technologies applied in semantic understanding.

Definition 1 (OLAP query). For the sake of simplicity, an OLAP query is denoted as q = <*gbc*, *pre*, *meas*> which consists of three parts: i). group-by clause (denoted as *gbc*). ii). Select predicate clause (denoted as *pre*). iii). Measure clause (denoted as *meas*).

Example 1: Think about the IPUMS, which is a census micro database that contains five dimensional levels with full roll-up orders (e.g. from City to All Cities): RESIDENCE, RACE, TIME, SEX and OCCUPATION, and six dimensions that would be measured with aggregate functions such as sum, average, max and min.

Definition 2 (OLAP session). An OLAP session S_k is a combination of a series of related queries q_i ($S_k = \{q_i \mid q_i \in S_k\}$), which aim at exploring the desired target by a succession of OLAP operations. The length of an OLAP session can be denoted as S_{len}.

Example 2: An OLAP session consists of a series of queries, each of which contains three parts: group-by, select predicate and measure clauses, where there will be some relations between the adjacent queries.

Definition 3 (OLAP similarity metric). We define the similarity calculation method by three clauses mentioned in Definition 1 [15], the structure of which is depicted in **Fig. 4** (For simplicity, "ST" is short for "STATE", "CI" is "CITY", "SE" is "SEX", "YE" is "YEAR", "MR" is "MRN", "RA" is "RACE", "RG" is "REGION").

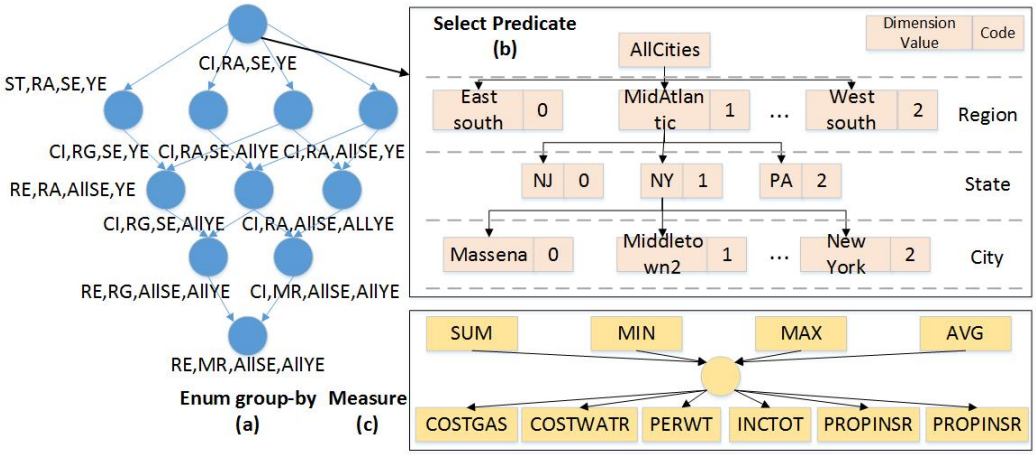


Fig. 4. Example of similarity calculation among queries

Based on the definitions above, the assessment of similarity between two queries consists of three parts: the group-by clause, the select predicates and the measures, the calculation of which can be computed in Formula (1).

$$\sigma_{q_i,q_j} = 0.35 * \sigma_{gbc}(q_i,q_j) + 0.5 * \sigma_{pre}(q_i,q_j) + 0.15 * \sigma_{meas}(q_i,q_j) \in [0,1] \quad (1)$$

Where:

1). $\sigma_{gbc}(q_i,q_j)$ represents the similarity between group-by clauses in two queries, which is calculated by Formula (2):

$$\sigma_{gbc}(q_i, q_j) = 1 - \frac{\sum_{k=1}^{n} \frac{dist_{lev}(h.l_i(q_i), h'.l_i(q_j))}{len(l_i) - 1}}{n} \tag{2}$$

$dist_{lev}(h.l_i(q_i), h.l_i'(q_j))$ denotes the distance between two queries (q_i and q_j) on the level of the *i*-th dimension hierarchy while $len(l_i)$ represents the quantity of levels in the *i*-th hierarchy and *n* is the number of dimensions.

2). $\sigma_{pre}(q_i, q_j)$ shows the similarity of select predicates between two queries, which can be computed using Formula (3):

$$\sigma_{pre}(q_i, q_j) = 1 - \frac{\sum_{k=1}^{n} \frac{dist_{pre}(p_k(q_i), p_k'(q_j))}{len(l_k)}}{n} \tag{3}$$

$$dist_{pre}(p_i(q_i), p_i'(q_j)) = (\left|code_{hier}(p_i(q_i)) - code_{hier}(p_i'(q_j))\right| + 1) * \left|code_{val}(p_i(q_i)) - code_{val}(p_i'(q_j))\right| \tag{4}$$

Formula (4) represents the distance between predicates on the dimension hierarchy when the value is 0 if they have the same level and dimension value, if they have different dimension values but are still on the same level, it will be $\left|code_{val}(p_i(q_i)) - code_{val}(p_i'(q_j))\right|$, greater than $\left|code_{val}(p_i(q_i)) - code_{val}(p_i'(q_j))\right|$ if they are on the different levels [16].

3). $\sigma_{meas}(q_i, q_j)$ denotes the similarity between measures ($Meas_{q_i}$ and $Meas_{q_j}$), which is calculated by Formula (5) using the Jaccard coefficient:

$$\sigma_{meas}(q_i, q_j) = \frac{\left|Meas_{q_i} \cap Meas_{q_j}\right|}{\left|Meas_{q_i} \cup Meas_{q_j}\right|} \tag{5}$$

3. Our Proposed Methods for OLAP Recommendation

In this section, two novel algorithms will be proposed for OLAP recommendation based on collaborative filtering. More specifically, the Top-K recommendation algorithm is implemented by calculating the most similar OLAP sessions tailed with some personalized sessions (similarity-first) such that the higher frequency and the more similar partner for the middle or end-session which has already executed some queries as well as calculating the most similar users to acquire the sessions (user-first) for the head-session which has few queries executed, which is used to solve the cold-start problem [17]. The tag-aware algorithm is designed for labeling the recommended OLAP sessions based on the existing user-defined tag lib with modified changes [18, 21-23].

3.1 Top-K recommendation algorithm for OLAP sessions

Before the introduction of our proposed recommend method, similar findings will be listed for OLAP recommendation. Edit-based similarity of OLAP sessions is proposed by J. Aligon, the calculation of this method is shown in formula (6):

$$D[i][j]=\begin{cases} 0, & i=0 \ \ or \ \ j=0 \\ D[i-1][j-1], & i>0, j>0 \ \ and \ \ \sigma_{i,j}>\delta \\ \min\begin{cases} D[i-1][j]+1 \\ D[i][j-1]+1 \\ D[i-1][j-1]+1 \end{cases}, & i>0, j>0 \ \ and \ \ \sigma_{i_i,j}\le\delta \end{cases} \tag{6}$$

Where:

δ is a given threshold and $\sigma_{i_i,j}$ is the similarity between two queries.

Algorithm 1: Session similarity calculation based on Smith-Waterman

Input: current session *S*, candidate sessions *LS*, number of the recommendations *K*
Output: the most similar OLAP recommendation sessions

```
1   m= LS.getSessions.length
2   for each i in [1,m] do
3     currentSize = S.queryList.length
4     candidateSize = LS[i].queryList.length
5     create query similarity matrix qs
6     for each j in [1,currentSize] do
7       for each k in [1,candidateSize] do
8         qs[i][j] = querySimilarity(S.queryList[j], LS[i].queryList[k])
9       end for
10    end for
11    scoreMatrix = smithWaterman(S, LS[i], qs)
12    subQuerySequence = traceBack(scoreMatrix, qs)
13    sim = proportion(subQuerySequence, scoreMatrix)
14  end for
```

Illustrated in algorithm 1 is the similarity calculation between two sessions, which is calculated based on the Smith-Waterman algorithm. At line 11 in algorithm 1, the score matrix is calculated using Formula (7):

$$S[i][j]=\begin{cases} 0, & i=0 \ \ or \ \ j=0 \\ max\begin{cases} 0 \\ S[i-1][j-1]+(\sigma_{q_i,q_j}-\delta)*(\rho(l-i,l'-j)) \\ max\{S[k][j]-\delta*\dfrac{1-\theta}{e^{\sqrt{i-k}}-1}\}, & 1\le k<i; \\ max\{S[i][k]-\delta*\dfrac{1-\theta}{e^{\sqrt{j-k}}-1}\}, & 1\le k<j; \end{cases}, & \text{else} \end{cases} \tag{7}$$

Where δ represents the average similarity value of the whole couple of queries between two sessions, ρ is a time-discounting function that is improved in [10], θ is a given threshold that is limited in the interval [0, 1]. Finally, the similarity between two sessions which shown in line 13 can be calculated using Formula (8):

$$\sigma_{s_i,s_j} = \frac{\sum_{i=1}^{k} l_i * l'_i}{l * l'} \tag{8}$$

Where l and l' represent the length of sessions S_i and S_j, respectively while l and l' are the matched subsequences that belong to the corresponding sessions.

Algorithm 2: User similarity calculation based on profiling attribute

Input: user profile U_{pro}, current user U_{cur}, number of the recommends K, candidate sessions LS

Output: the most similar users with their corresponding sessions R_s

```
1   extract occupation information U_occ from U_pro
2   translate U_occ to dimension level value D_occ, U_cur to D_cur
3   m = D_occ.length
4   max ← ϕ
5   R_s ← ϕ
6   for each i in [1,m] do
7      sim = userSimilarityCalculation(D_cur, D_occ[i])
8      max.add(sim, D_occ[i].sessionId)
9   end for
10  for each j in [1,K] do
11     S_id = max.getTopValue(j).getSessionId
12     S_recommend = LS.getSession(S_id)
13     R_s.add(S_recommend)
14  end for
```

Illustrated in algorithm 2 is the similarity calculation between two users to get the Top-K similarity users to recommend the corresponding session. The similarity calculation formula of two users is shown in Formula (9):

$$\sigma_{U_i,U_j} = e^{-\eta * dist^{\alpha}(U_i,U_j)} \tag{9}$$

Where $dist(U_i,U_j)$ is the distance between the occupations of the two users on the level of the dimension hierarchy while $\eta = 3.8$ and $\alpha = 2$ [19-20]. Finally, the recommended sessions will be generated by the similar users.

Based on the formulas and algorithms above, the Top-K candidate recommended sessions can be acquired, before the final recommendation, a fitting algorithm will be applied [10, 24-26] after which the recommendation of OLAP session S_{cur} for user U_i can be shown by Formula (10):

$$R(U_i,S_{cur}) = \begin{cases} \left\{ S_l \mid S_l \mid max_{1\le j\le n}\{\sigma_{S_{cur},S_j}\}, 1\le l\le K \right\}, & len(S_{cur}) \le \lambda \\ \left\{ S_j \mid max\{U_{max}(S_1,S_2,...S_x)\}, 1\le j\le K \right\}, & \text{else} \end{cases} \tag{10}$$

Where U_{max} get the max value in $\{\sigma_{U_i,U_l}\}_{1\le l\le m}$, λ is a threshold value that distinguishes the head-session or not.

3.2 Tag-aware algorithm for labeling the recommended OLAP sessions

This subsection introduces a tag-aware algorithm that can label the OLAP sessions by user-defined tags, the main idea of which is the similarity calculation of OLAP sessions so as to find the most similar session to apply its tag by changing the difference between the two sessions in order to solve the sparseness of tags.

3.2.1 Building approximate tags based on the OLAP session similarity

The OLAP session is labeled by user for specific meaning and constitutes a triadic relation (user, session and tag). However, only a few tags will be allocated to the OLAP sessions that cause the sparseness. To solve this problem, we get the similar tag from the most similar OLAP session by translating the triadic relation to the tag-session matrix, which we demonstrate that the similar session has a similar semantic of tags.

In addition, we also apply an adaptable method to change the semantics by highlighting the differences between two sessions, which is illustrated in **Fig. 5**. The three kinds of modified changes are the basic inconformity between two sessions which when combined can describe the situations between the two similar sessions.

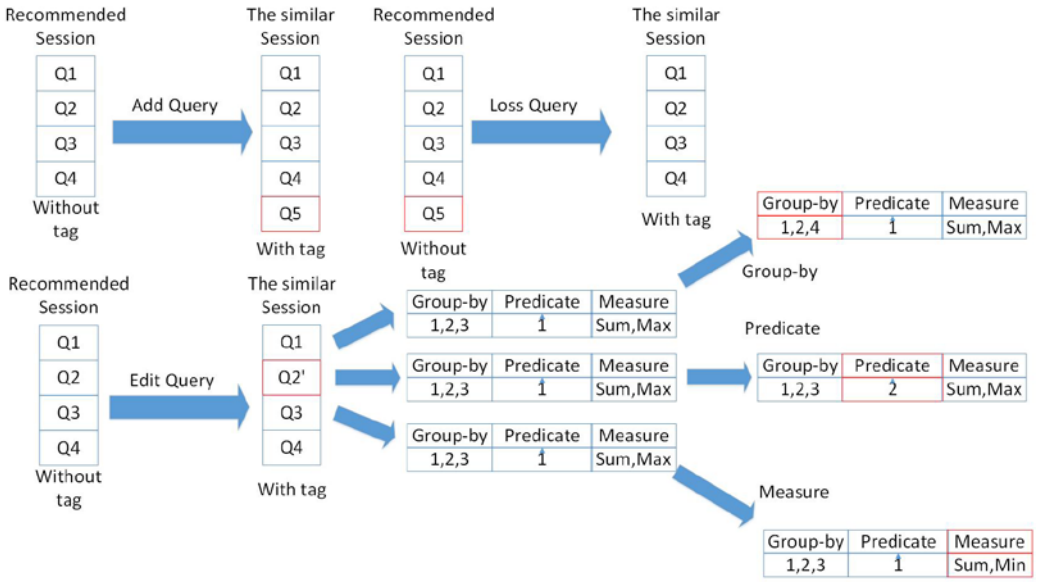


Fig. 5. Three kinds of modified changes between two sessions

To describe our proposal in detail, we use the following Formula (11) to (13) to express the three changes.

$$T_{add} = \{q_1, q_2 ... q_s\}_{recommend} \sim \{q_1, q_2 ... q_t\}_{similar}, \quad s < t \tag{11}$$

Where:

T_{add} denotes that the recommended session has less queries than the similar session

$$T_{edit} = \{q_1 ..., q_i ..., q_s\}_{recommend} \sim \{q_1 ..., q_i' ..., q_s\}_{similar}$$
$$\sigma_{q_i, q_i'} = 0.35 * \sigma_{gbc}(q_i, q_j) + 0.5 * \sigma_{pre}(q_i, q_j) + 0.15 * \sigma_{meas}(q_i, q_j) < 1 \tag{12}$$

Where:

T_{edit} represents that the queries of the recommended session have some different changes with the similar session which shows that the similarity between the different queries is below 1.

$$T_{loss} = \{q_1, q_2 ... q_s\}_{recommend} \sim \{q_1, q_2 ... q_t\}_{similar}, \quad s > t \tag{13}$$

Where:

The T_{loss} denotes that the recommended session has more queries than the similar session

3.2.2 Tag generation algorithm based on the OLAP session similarity

Algorithm 3: Tag generation algorithm based on the OLAP session similarity

Input: recommended session *RS*, candidate session *LS*, tag-session matrix *TS*

Output: the corresponding tags for recommended sessions

```
1  n = RS.length
2  Tag ← ϕ
3  for each i in [1,n] do
4     SS = sessionSimilarity(RS[i], LS)
5     m = SS.length
6     for each j in [1,m] do
7        //the tag of the similar session is not null
8           if(TS[i][j]!=null){
9              //get the final tag through the modified changes
10             //between two sessions
11             ts = modifiedChanges(RS[i], SS[j], TS[i][j])
12             Tag.add(ts)
13          }
14    end for
15 end for
```

Illustrated in algorithm 3 is the tag generation approach which is used for labeling the recommended sessions based on the session similarity, which combines three basic kinds of modified changes to describe the meaning of the recommended sessions accurately.

4. System Implementation and Experimental Evaluation

To testify the validity of our proposal, a prototype system is implemented which consists of two parts shown in **Fig. 1**: the left part is an OLAP recommendation system which has been deployed on the Jetty web container that invokes the API of the Big Cube System to execute OLAP queries while the right part is used for providing the OLAP engine to return the result of an OLAP query in a very short time. All the experiments are conducted on Windows 7 pro SP1 over a 64-bits Intel Core i5 quad-core 3.3 GHz, with 8 GB RAM.

4.1 User interface of the prototype system for OLAP session recommendation

Fig. 6 shows the user interface of our OLAP recommendation system. The left part of the system is used for group-by and measuring clause selection, the content of which will be presented on the middle part including a column and a row in which the result will be listed in the form of a table or other charts after executing the running button. After the result is shown,

the recommended sessions are listed on the right, which can be executed one by one, and the tags are generated followed by the recommendation, which is signed with modified changes if there are no tags defined on the recommendation sessions, and a direct recommended session if it has.

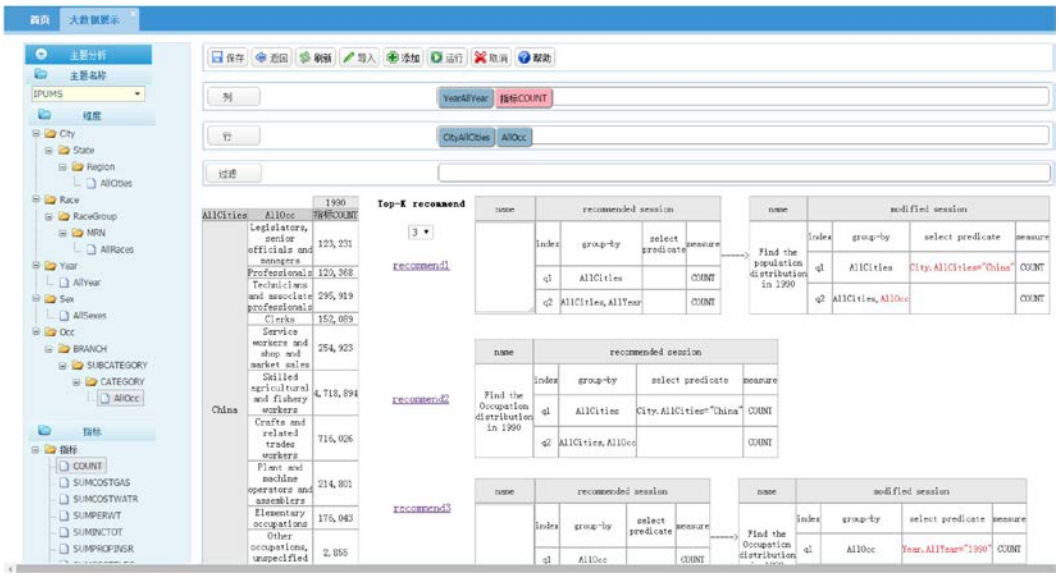


Fig. 6. Three kinds of modified changes between two sessions

4.2 The parameterized settings and assessment criterions

Table 1 shows the abbreviation of symbols of **Table 2**, which has set a lot of parameters (e.g. Length of sessions, Number of sessions per seed query) to generate different kinds of query logs.

Table 1. Abbreviation of symbol

symbol	abbreviation	symbol	abbreviation
Max number of measures	Max_{meas}	Length of sessions	S_{len}
Size of seed query reports	Num_{rep}	Number of sessions per seed query	S_{num}
Number of surprising queries	Num_{sup}	Year prompt fraction	$Year_{frt}$
Number of seed queries	Num_{seed}	Segregation predicate	$Pred_{seg}$

Table 2. Parameterized setting for OLAP log generation.

Max_{meas}	K=1				K=3				K=5			
	$Year_{frt}$ =0.25		$Pred_{seg}$ =No		$Year_{frt}$ =0.25		$Pred_{seg}$ = No		$Year_{frt}$ =0.25		$Pred_{seg}$ =No	
	S_{len}	S_{num}	S_{len}	S_{num}	S_{len}	S_{num}	S_{len}	S_{num}	S_{len}	S_{num}	S_{len}	S_{num}
3	3-6	20	3-6	120	3-6	20	3-6	120	3-6	20	3-6	120
3	6-9	40	6-9	140	6-9	40	6-9	140	6-9	40	6-9	140
5	9-12	60	9-12	160	9-12	60	9-12	160	9-12	60	9-12	160
5	7-12	80	7-12	180	7-12	80	7-12	180	7-12	80	7-12	180
5	3-9	100	3-9	200	3-9	100	3-9	200	3-9	100	3-9	200

The assessment criterion for the experiments usually consists of three parts: precision, recall and F-measure. In the OLAP recommendation area, the precision is measured by considering whether the recommended session is the actual applied session, the recall is used to measure

that whether the recommended sessions contain the actual session and to assess the aggregative indicator, we use the F-measure to balance the precision and recall so that the result will have a better assessment. The formula of the three parts can be calculated in formula (14), (15), (16).

$$precision = \frac{|TP|}{|TP| + |FP|} \tag{14}$$

$$recall = \frac{|TP|}{|TP| + |FN|} \tag{15}$$

$$F - measure = \frac{2 * precision * recall}{(precision + recall)} \tag{16}$$

Where $|TP|$ represents the actual session in the following step is similar to the recommended ones, $|FP|$ is the false recommended set that does not match the actual counterparts and $|FN|$ represents the missing recommended sets that appear in the actual counterparts.

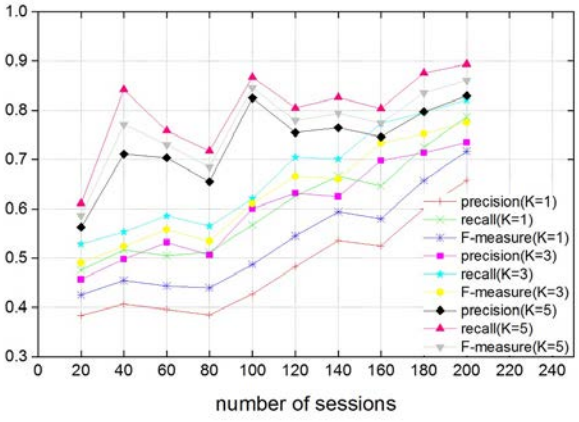
4.3 Experimental results and evaluations

Fig. 7 shows the evaluation of our recommended method which consists of four parts: the number of sessions, the number of recommended sessions, the length of a single session and the length of the current session.

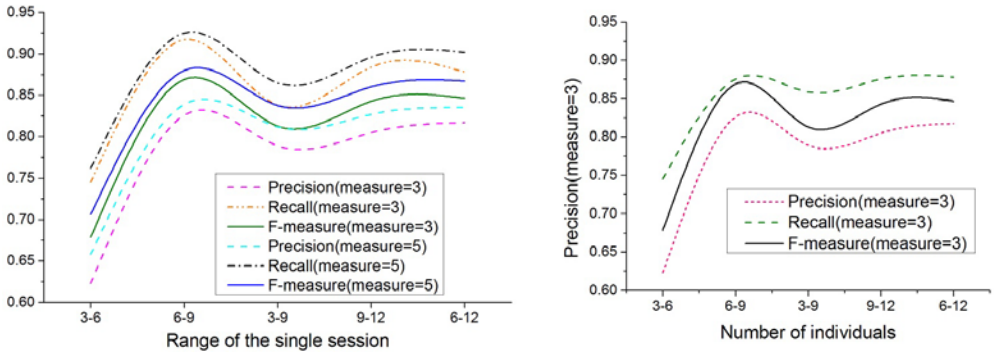
Fig. 7 (a) shows that with the increasing number of sessions, the three measures are growing generally since the more the sessions the less the probability of contingency with misjudgment. In addition, the more recommended sessions, the better effectiveness of the three measures, which is easy to understand that the more recommended sessions, the more choices that can be adopted to maximize the precision, recall levels can also increase simultaneously because of the Top-K recommendation.

Fig. 7 (b) illustrates the variation of the three measures with the change of length range of a single session, which can see that the 6-9 of the length range get the highest measure value, the result of which shows that the length of a single session may not be recommended accurately.

Fig. 7 (c) shows the influence of the length of the current session on the measure value, which demonstrates that with the increasing length of the current session, the three measures grow and have a relative high value because we adopt the user-first and similarity-first strategy to cope with the different situations to enhance the actual effectiveness.



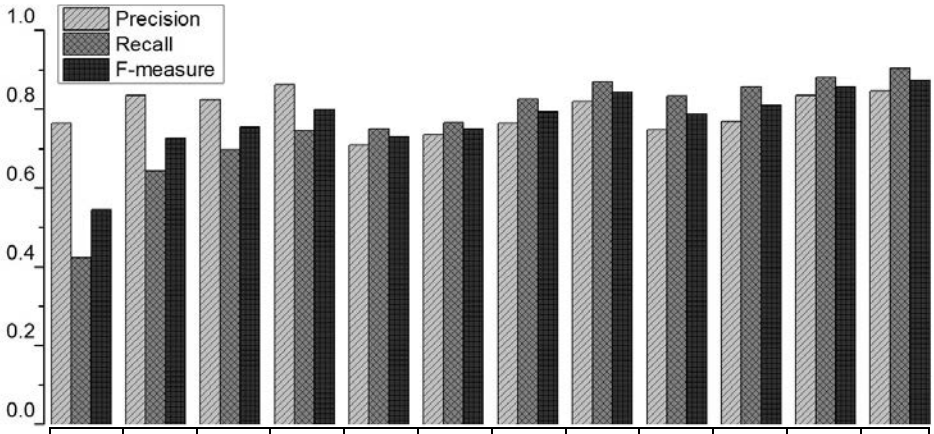
(a)



(b) (c)

Fig. 7. Evaluation of our recommend method with different parameters

we choose four scales of session sets S1, S2, S3, S4 (the number of which are 50, 100, 150, 200, respectively), which is depicted in **Fig. 8**, the result between collaborative filtering and our Top-K method with different K values shows that although the precision of our method is a little lower than collaborative filtering, ours has a striking recall and thus a higher F-measure, for our approach can provide more choices to find accurate results.



	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
	Collaborative filtering				Top-3 recommend				Top-5 recommend			
precision	0.765	0.835	0.824	0.863	0.709	0.735	0.764	0.819	0.748	0.769	0.835	0.846
recall	0.423	0.643	0.696	0.745	0.75	0.765	0.824	0.869	0.833	0.857	0.88	0.904
F-measure	0.544	0.726	0.754	0.799	0.729	0.75	0.793	0.843	0.788	0.81	0.857	0.874

Fig. 8. Comparison of effectiveness between Top-K recommendation and Collaborative filtering algorithm on a variety number of sessions

5. Conclusion

Superior OLAP recommendation system is a novel topic in the decision making field. In this paper, a Top-K OLAP recommendation system for OLAP sessions is designed to fill up this blank using collaborative filtering technique, the kernel module of which consists of four parts: the vectorial coordinate is used for query parameterization of the input logs as the similarity calculation is adopted using Smith-Waterman algorithm among sessions, the Top-K recommendation applied to OLAP sessions is a combination of similarity-first and user-first

strategy and the label generator is designed for semantic understanding based on the OLAP session similarity. The theory together with our prototype system have well solved the low recall of the recommendation system and provided a user-friendly interface for the user to understand the actual meaning of our recommendations by the generated tags. The key to the recall of our recommendation session is the Top-K recommendation method of OLAP sessions and a series of experiments are conducted to show that although our method has a little lower precision than the customized collaborative filtering, our method has a remarkable higher recall and thus finer F-measure to recommend a variety of sessions to find an accurate result.

Our future work will continue to refine our prototype and optimize the algorithm we propose, mainly on how to raise the precision for a better and accurate recommendation.

References

[1] S. Laraichi, A. Hammani, A. Bouignane, "Data integration as the key to building a decision support system for groundwater management: Case of Saiss aquifers, Morocco," *Groundwater for Sustainable Development*, vol. 2-3, pp. 7–15, August–September, 2016. Article (CrossRef Link)

[2] W. Jakkhupan, S. Kajkamhaeng, "Movie Recommendation Using OLAP and Multidimensional Data Model," in *Proc .of International Conference on Computer Information Systems and Industrial Management*, pp. 209-218, November, 2014. Article (CrossRef Link)

[3] P. Marcel, "Log-driven user-centric OLAP," *International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp. 1446-1451, May, 2014. Article (CrossRef Link)

[4] L. Sautot, B. Faivre, L. Journaux, P. Molin, "The hierarchical agglomerative clustering with Gower index: A methodology for automatic design of OLAP cube in ecological data processing context," *Ecological Informatics*, vol. 26, pp. 217-230, March, 2015. Article (CrossRef Link)

[5] J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29-39, March, 2017. Article (CrossRef Link)

[6] L. Zhu, C. Xu, J. Guan, H. Zhang, "SEM-PPA: A semantical pattern and preference-aware service mining method for personalized point of interest recommendation," *Journal of Network and Computer Applications*, vol. 82, pp. 35–46, March, 2017. Article (CrossRef Link)

[7] A. Singh, N. Parimala, "Recommending Next Query in an OLAP Session," *International Symposium on Computational & Business Intelligence*, pp. 73-80, December, 2014. Article (CrossRef Link)

[8] S. Ali, O. Boussaid and F. Bentayeb, "Towards Collaborative Multidimensional Query Recommendation with Triadic Association Rules," *Decision Support Systems*, vol. 7, no. 3, pp. 17-35, July-September, 2015. Article (CrossRef Link)

[9] S. Rizzi, E. Gallinucci, "CubeLoad: A Parametric Generator of Realistic OLAP Workloads," in *Proc .of Conference on Advanced Information Systems Engineering*, pp. 610-624, June. 16-20, 2014. Article (CrossRef Link)

[10] J. Aligon, E. Gallinucci, M. Golfarelli, P. Marcel, S. Rizzi, "A collaborative filtering approach for recommending OLAP sessions," *Decision Support Systems*, vol. 69, pp. 20-30, January, 2015. Article (CrossRef Link)

[11] D. Vandic, JWV. Dam, F. Frasincar, "A semantic-based approach for searching and browsing tag spaces," *Decision Support Systems*, vol. 54, pp. 644-654, December, 2012. Article (CrossRef Link)

[12] Y. Zuo, J. Zeng, M. Gong, L. Jiao, "Tag-aware recommender systems based on deep neural networks," *Neurocomputing*, vol. 204, pp. 51-60, September, 2016. Article (CrossRef Link)

[13] Y. Li, G. Li, L. Shu, Q. Huang, and H. Jiang, “Continuous Monitoring of Top-k Spatial Keyword Queries in Road Networks,” *Journal of Information Science and Engineering*, vol. 31, no. 6, pp. 1831-1848, November, 2015. Article (CrossRef Link)
[14] D. Liu, “Novel Semantics of the Top-k Queries on Uncertainly Fused Multi-Sensory Data,” *Journ al of Information Science and Engineering*, vol. 31, no 1, pp. 179-205, January, 2015. Article (CrossRef Link)
[15] J. Aligon, M. Golfarelli, P. Marcel, S. Rizzi, E. Turricchia, “Similarity measures for OLAP sessions,” *Knowledge and Information Systems*, vol. 39, no. 2, pp. 463-489, May, 2014. Article (CrossRef Link)
[16] J. Song, C. Guo, Z. Wang, Y. Zhang, G. Yu and J. M. Pierson, “HaoLap: A Hadoop based OLAP s ystem for big data,” *The Journal of Systems and Software*, vol. 102, pp. 167-181, April, 2015. Article (CrossRef Link)
[17] L. Safoury, A. Salah, “Exploiting user demographic attributes for solving cold-start problem in re commender system,” *Lecture Notes on Software Engineering*, vol. 1, pp. 303-307, January, 2013. Article (CrossRef Link)
[18] D. Gkesoulis, P. Vassiliadis, P. Manousis, “CineCubes: Aiding data workers gain insights from OLAP queries,” *Information Systems*, vol. 53, pp. 60-86, October–November, 2015. Article (CrossRef Link)
[19] M.Y.H. Al-Shamri, “User profiling approaches for demographic recommender systems,” *Knowledge-Based Systems*, vol. 100, pp. 175–187, March, 2016. Article (CrossRef Link)
[20] M.Y.H. Al-Shamri, “Power coefficent as a similarity measure for collaborative recommender system,” *Expert Systems with Applications*, vol. 41, no. 13, pp. 5680–5688, October, 2014. Article (CrossRef Link)
[21] S. Xie, Y. Wang, "Construction of Tree Network with Limited Delivery Latency in Homogeneous Wireless Sensor Networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 231-246, April, 2014. Article (CrossRef Link)
[22] J. Shen, H. Tan, J. Wang, J. Wang, S. Lee, "A Novel Routing Protocol Providing Good Transmission Reliability in Underwater Sensor Networks,” *Journal of Internet Technology*, vol. 16, no. 1, pp. 171-178, January, 2015. Article (CrossRef Link)
[23] Y. Zhang, X. Sun, B. Wang, "Efficient Algorithm for K-Barrier Coverage Based on Integer Linear Programming,” *China Communications*, vol. 13, no. 7, pp. 16-23, July, 2016. Article (CrossRef Link)
[24] P. Guo, J. Wang, XH. Geng, CS. Kim, JU. Kim, "A Variable Threshold-value Authentication Architecture for Wireless Mesh Networks," *Journal of Internet Technology*, vol. 15, no. 6, pp. 929-936, November, 2014. Article (CrossRef Link)
[25] G. Jia, G. Han, J. Jiang, L. Liu, "Dynamic Adaptive Replacement Policy in Shared Last-Level Cache of DRAM/PCM Hybrid Memory for Big Data Storage,” *IEEE Transactions on Industrial Informatics*, vol. 99, pp. 1-1, December, 2016. Article (CrossRef Link)
[26] G. Han, L. Liu, S. Chan, R. Yu, Y. Yang, “HySense: A Hybrid Mobile CrowdSensing Framework for Sensing Opportunities Compensation under Dynamic Coverage Constraint,” *IEEE C ommunications Magazine*, vol. 55, pp. 93-99, March, 2017. Article (CrossRef Link)

Youwei Yuan was born in 1966, received his doctor degree in computer science from Wuhan University of Technology, China, in 2007. He is currently a professor of computer science, Hangzhou Dianzi University (Hangzhou, China). His research interests include artificial intelligence, data mining, Big data analytics and distributed parallel processing, He has published over 50 technical papers in prestigious journals and conferences, 30 papers been indexed by SCI and EI.

Weixin Chen was born in 1992, is a postgraduate student of computer science, Hangzhou Dianzi University (Hangzhou, China). His research interests include Big data analytics and Data Mining.

Guangjie Han is currently a Professor with the Department of Information and Communication System, Hohai University, Changzhou, China. He received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004. From 2004 to 2006, he was a Product Manager for the ZTE Company. In February 2008, he finished his work as a Postdoctoral Researcher with the Department of Computer Science, Chonnam National University, Gwangju, Korea. From October 2010 to 2011, he was a Visit-ing Research Scholar with Osaka University, Suita, Japan. He is the author of over 220 papers published in related international conference proceedings and journals, and is the holder of 90 patents. His current research interests include sensor networks, computer communications, mobile cloud computing, and multimedia communication and security.

Gangyong Jia is currently an Assistant Professor of Department of Computer Science at Hangzhou Dianzi University, China. He received his Ph.D. degree in Department of Computer Science from University of Science and Technology of China, Hefei, China, in 2013. His current research interests are power management, operating system, cache optimization, memory management.