Recommending Multidimensional Spatial OLAP Queries

Olfa Layouni, Fahad Alahmari and Jalel Akaichi

Abstract Huge volumes of data are stored in a spatial data warehouse. Stored data is explored and analyzed by different users. The users interrogate spatial data cube through Spatial OLAP queries which are generally misspoke, leading to non-pertinent results. Therefore, we propose in this paper a Spatial OLAP queries recommendation system based on the SOLAP server query log; that helps and guides users in their exploration. Compared to the users needs, the proposed queries return more relevant results.

Keywords Spatial OLAP · Spatial data cube · SOLAP · Query · Recommendation

1 Introduction

Nowadays, the popularity of spatial data, such as names of cities, postal codes, the position of individual object in space, the maps created from satellite images, can be stored in a spatial data warehouse. According to Franklin [13] spatial data representing about 80 % of all data stored in databases has a spatial or location component, therefore, data are stored in a spatial data warehouse. Stefanovic et al. [28] defined a spatial data warehouse as a collection of spatial and thematic, integrated, nonvolatile and historical data to support the spatial decision making process. In addition, the analysis of spatial data warehouse allows the historical data; the integration and the storage of large volumes of spatial and non-spatial data from multiple sources. A spatial data warehouse contains both a spatial and an alphanumeric data type. It

O. Layouni (✉)
BESTMOD Laboratory, Université de Tunis, ISG - Tunis, Tunis, Tunisia
e-mail: layouni.olfa89@gmail.com

F. Alahmari · J. Akaichi
College of Computer Science, King Khalid University, Abha, Saudi Arabia
e-mail: fahad@kku.edu.sa

J. Akaichi
e-mail: jalel.akaichi@kku.edu.sa

G. De Pietro et al. (eds.), *Intelligent Interactive Multimedia Systems and Services 2016*, Smart Innovation, Systems and Technologies 55,
DOI 10.1007/978-3-319-39345-2_35

is made of a multidimensional spatial model which defines the concepts of spatial measures and dimensions in order to take into account the spatial component. In the literature [4, 19, 21], the authors identified three types of spatial dimensions hierarchies based on the spatial references of the hierarchy members: non-geometric spatial dimensions, geometric-to-non-geometric spatial dimensions and fully geometric spatial dimensions. Also, a spatial data warehouse supports two types of spatial measures: the first type of spatial measures is a set of all the geometries representing the spatial objects corresponding to a particular combination of dimension members. The second type of spatial measures results from the computation of spatial metric or topological operators [2, 3]. In order to analyze and explore a spatial data warehouse, we need a SOLAP server to help the user to make the best decisions.The SOLAP has been identified as an effective means to explore the contents of a spatial data warehouse. The SOLAP is the result obtained after the combination of Geographic Information Systems (GIS), with OLAP tools. To navigate in the spatial data cube the user launches a sequence of SOLAP queries over a spatial data warehouse. A spatial data cube can be queried by using the MDX *(Multi-Dimensional eXpressions)* with spatial extensions query language [2].

SOLAP users interactively navigate a spatial data cube by launching sequences of SOLAP queries over a spatial data warehouse. The problem appeared when the user may have no idea of what the forthcoming SOLAP queries should be. As a solution and to help the user in his navigation, we need a recommendation system. This system gives the possibility to recommend SOLAP queries.

This paper is organized as follows: Sect. 2 introduces related works. Section 3 enlighten our approach of recommending SOLAP queries. Section 4 presents our experimentation. We discuss future work in Sect. 5.

2 Related Work

In this section we focus on the related works of the exploration of spatial data warehouse to provide recommendations to the user. For this reason, we begin by presenting a recommendation system concept, in order to help the user in his exploration of data. Then, we introduce the various methods that have been proposed to explore data.

A recommendation system is usually categorized into a content-based method, a collaborative method and a hybrid method [1, 20].

- *Content-based method*: The user is recommended elements similar to the ones the user preferred in the past.
- *Collaborative method*: The current user is recommended elements similar to the preferences of the previous users and the preferences of the current user.
- *Hybrid method*: This method combines both the content-based and the collaborative method.

In various studies [20, 25–27], we find that the authors described the characteristics of the general algorithm of a recommender system for the exploration of data. These characteristics are the inputs, outputs and the recommendation steps. The inputs of the algorithm can be a log of sessions of queries, a schema, an instance of the relational or multidimensional database, a current session and a profile. The outputs of the algorithm can be a query, a set of ordered queries and a set of tuples. An algorithm of recommendation is decomposed into three steps. The first step consists in choosing an approach for evaluating the used scores. In fact, in this step we can choose one of the categories of recommendation: a content-based or a collaborative or a hybrid method. The second step is the filter; this step consists in selecting the candidates' recommendations. The last step is the guide; this step consists in organizing the candidates' recommendations. Furthermore, we describe some methods that recommend queries for helping users to explore data. In fact, those methods can be classified into two categories, the first category exploits the profile in [5, 6, 25–27] and so does the second category with the log of queries [20, 22–24]. In fact, we find that the methods proposed by [5, 6, 25–27] take as inputs of the algorithm the profile of the current user. Also, the methods proposed by [22–24] take as inputs the log of OLAP sessions of queries. Indeed, we find that the inputs of the algorithm can be: a schema, an instance, the current OLAP query, the current OLAP session, the previous OLAP sessions [5, 6, 20, 22–27]. Besides, we find that the output of the methods proposed by [22–24] is a query, those proposed by [5, 6, 20] the output is a set of queries and only the method proposed by [25–27] the output is a set of tuples. So far, we have found that the methods proposed by [5, 6, 25–27] are content based methods and the methods proposed by [20, 22–24] are collaborative methods. Adding to that, we have discovered that, in the filter step of the proposed algorithm, the method proposed by [5, 6, 20, 25–27] gives the possibility to select the candidate's queries recommendation. Besides, for computing the candidate's recommendations the methods proposed by [25–27] apply the maximum entropy theory; also the methods proposed by [5, 6] use a graphic model; as well the methods proposed by [22–24] apply the Markov model. However, the guiding step was applied in the methods proposed by [20, 22–24]. And finally, we find that the methods proposed by [22–27] used the SQL language, the methods proposed by [5, 6, 20] used the MDX language.

3 Recommendation System

In this section we detail our approach for recommending SOLAP queries. To help the user to go forward in his exploration of the spatial data cube, we propose an approach for recommending SOLAP queries. Our approach consists of three steps. The first step consists in computing all the generalized sessions of SOLAP queries of the log. The second step is the filter which consists in predicting the candidates SOLAP queries. The last step is the guide that consists in ordering the candidates SOLAP queries. But in order to recommend queries, we need to compute the similarity between queries. So, in our context, we need to compute the similarity between SOLAP queries by taking into account spatial relations as topological, orientation and distance metric relations.

3.1 Spatial Similarity Measures

In SOLAP queries, we found the use of the three main categories of spatial relations, which are defined in the literature as follows: topological relation, direction relation and metric distance relation [9, 14, 16]. So, to compare between two SOLAP queries, we need to measure the similarity between the topological relation, direction relation and metric distance relation, invoked in each query.

3.1.1 Topological Distance

A spatial scene could be a topological relation between spatial objects [10, 17, 18]. The topological distance between two SOLAP queries takes into account not only the two spatial topological relations TR invoked by a SOLAP query q and TR' invoked by a SOLAP query q'; with TR and $TR' \in \{disjoint, meet, overlap, coveredBy, equal, covers, contains, inside\}$; but also spatial objects (a, b) and (a', b') invoked in each query q and q', respectively.

If $a \neq a'$ and $b \neq b'$ then we measure the similarity distance between TR and TR', and the similarity distance between objects a, a', b and b' invoked in each query q and q', respectively. The distance between objects represents the similarity between references according to the schema of the spatial data warehouse. In this case, to measure the topological distance, we use the distance proposed by [11], in order to measure the similarity between TR and TR'. So, the $Dist_{topR}(TR, TR')$ represents this similarity and takes a value by comparing between TR and TR' lake as indicated in table, which represents in Fig. 1.

	disjoint	meet	equal	inside	coveredBy	contains	covers	overlap
disjoint	0	1	6	4	5	4	5	4
meet	1	0	5	5	4	5	4	3
equal	6	5	0	4	3	4	3	6
inside	4	5	4	0	1	6	7	4
coveredBy	5	4	3	1	0	7	6	3
contains	4	5	4	6	7	0	1	4
covers	5	4	3	7	6	1	0	3
overlap	4	3	6	4	3	4	3	0

Fig. 1 Topological distance [11]

In this case, the topological distance between two SOLAP queries $Dist_{top}$ is calculated as follows:

$$Dist_{top}(TR(a,b),TR'(a',b')) = Dist_{topR}(TR,TR') + \sum_{i=1}^{n} Dist_{objects}(r_1,r_2) \quad (1)$$

If $a = a'$ and $b = b'$ then we measure the similarity distance between TR and TR' which represents the topological relationships invoked in each query q and q', respectively. In this case, the topological distance is computed as follows:

$$Dist_{top}(TR(a,b),TR'(a',b')) = Dist_{topR}(TR,TR') \quad (2)$$

3.1.2 Direction Distance

To compute the direction distance between two spatial direction scenes invoked in SOLAP queries q and q'. We propose to adopt the direction distance proposed by [16]; which based on the 9-direction system north, northwest, west, southwest, south, southeast, east, northeast, and equal to represent directions [11, 16]. This distance measure represents the transformation cost from any direction to any other, each transformation is equal to 2 as shown in Fig. 2.

The direction distance between two SOLAP queries $Dist_{Dir}(q,q')$ is calculated as follows:

$$Dist_{Dir}(q,q') = \sum_{i=1}^{2} Cost(TOR(a,b),TOR'(a',b')) \quad (3)$$

With:

- $TOR(a,b)$ and $TOR'(a',b')$ represent two spatial direction scenes invoked in q and q', respectively.
- $Cost$ is the transformation cost.

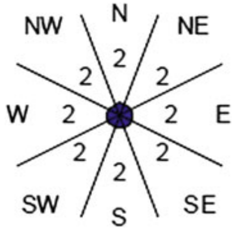
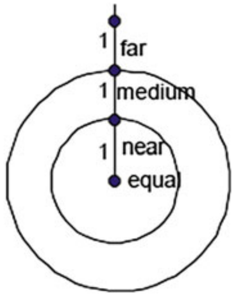


Fig. 2 Direction distance [16]

Fig. 3 Metric distance [16]



3.1.3 Distance in Term of the Metric Distance

To compute the metric distance between two spatial metric distance scenes invoked in SOLAP queries q and q'. We propose to adopt the distance proposed by [16]; which is based on the traditional 4-granularity metric distance equal, near, medium, far [11, 16]. This distance measure represents the cost of transitions for one granularity to another,each transformation is equal to 1 as shown in Fig. 3.

The metric distance between two SOLAP queries $Dist_{MetD}(q, q')$ with two spatial scenes based on the metric distance $TD(a, b)$ and $TD'(a', b')$ invoked in q and q', respectively, is calculated as follows:

$$Dist_{MetD}(q, q') = Dist_{MetD}(TD(a, b), TD'(a', b')) = \sum_{i=1}^{2} \sum_{j=1}^{2} a_{ij} \tag{4}$$

With:

- a, b, a' and b' are spatial objects invoked in each query q and q'.
- i and j are the number of objects invoked in q and q', respectively.
- $$a_{i,j} = \begin{cases} = 0 & if \quad i \quad is \quad equal \quad to \quad j \\ = 1 & if \quad i \quad is \quad near \quad to \quad j \\ = 2 & if \quad i \quad is \quad medium \quad to \quad j \\ = 3 & if \quad i \quad is \quad far \quad to \quad j \end{cases}$$

3.1.4 Spatial Distance

SOLAP queries launched by users in the cube have spatial scenes. Each scenes represents a spatial representation of one or more spatial objects. This scene could be a topological relation, direction relation and metric distance relation between them [10, 17, 18]. Given two SOLAP queries q and q'; the spatial similarity measure between them $Dist_{SpatialR}(q, q')$ is modeled as follows:

$$Dist_{SpatialR}(q, q') = Dist_{top}(q, q') + Dist_{Dir}(q, q') + Dist_{MetD}(q, q') \tag{5}$$

3.2 SOLAP Queries Recommendation System

The first step is to generalize sessions of SOLAP queries saved in the log, which contains all the previous sessions of SOLAP queries, in order to obtain a generalized log. In this step, we propose to use our spatial similarity measures proposed in the previous subsection, to compute the similarity between SOLAP queries. Also, we propose to use the method of TF-IDF (Term Frequency-Inverse Document Frequency) [8] to evaluate the importance of terms like spatial measures, spatial dimensions, etc. In order to know which dimensions or measures are spatial types. It is used to extract all the spatial dimensions and measures from the schema of the spatial data cube. Finally, for the classification of SOLAP queries, we choose the Hierarchical Ascendant Classification (HAC) [12]. It is used for classifying queries in two different classifications one contains the SOLAP queries without spatial data: OLAP queries and the other contains the SOLAP queries. The second step uses the result obtained in the first step to search the most similar sessions to the generalized current session. That's why, we need to search among the set of generalized sessions of SOLAP queries the ones that are the most similar to the generalized current session and also the query representing its session. This step gives the possibility to move from a candidate SOLAP session to a SOLAP candidate query. The results obtained in this step can be a set of candidates SOLAP queries or an empty set. If we obtain an empty set, the recommendation of queries is done by the default function, which implements the idea proposed by [15]. And if we obtain a set of candidates SOLAP queries, we sort this set in the order of the most similar to the query that represents the current session. So, we order this set first by calculating the distance between candidates SOLAP queries and the SOLAP query representing the current session and that is by using the quick sort in order to order them.

Algorithm 1 Spatial-OLAP-RS (*Sc*, *Log*)

Input: *Sc*: The current session,
Log: the log of sessions of SOLAP queries.
Output: An ordered set of SOLAP queries recommendations.

```
1: Loggeneralized ← Generalize(Log)
2: CandidatesSOLAP ← Filter(Sc, Loggeneralized)
3: if (CandidatesSOLAP ≠ ϕ) then
4:     Guid(CandidatesSOLAP, QuickSort)
5: else
6:     Default(Loggeneralized)
7: end if
```

4 Experimentation

In this section, we present the results of the experiment that we have conducted to assess the capabilities of our approach. This system gives the possibility to recommend an ordered set of SOLAP queries for the user, after launching the current session. First, to navigate in the spatial data cube the current user launched a sequence of SOLAP queries by using the SOLAP server over a spatial data warehouse. All the previous sessions of SOLAP queries are stored in the log. In fact, our *Spatial-OLAP-RS* system recommends an ordered set of SOLAP queries to the current user. Our experiment evaluates the efficiency of our approach proposed to recommend SOLAP queries. The performance is presented in Fig. 4 according to various log sizes. These log sizes are obtained by playing with two different parameters that change over the time. Those two parameters are as follows:

- X: represents the number of sessions store in the log, it ranges from 10 to 150.
- Y: represents the maximum number of queries that could be launched for per session, it ranges from 10 to 50.

The goal of our experimentation is to measure the execution time taken by applying our proposed approach. So, we conclude that the time taken to recommend SOLAP queries increases with the log size but remains highly acceptable as shown in Fig. 4.

To evaluate the performance of our approach, we choose to use the precision indicator. It's widely used to assess the quality and performance of recommendations systems [7]. In fact, the precision indicator reflects the fraction of recommendations which are the most relevant to the user preferences. In our case, the precision indicator is obtained as follows:

$$Precision = |\{RelevantSOLAPQueriesRecommendation$$

$$\cap ProposedSOLAPQueriesRecommendation\}|$$

$$/|\{ProposedSOLAPQueriesRecommendation\}|$$

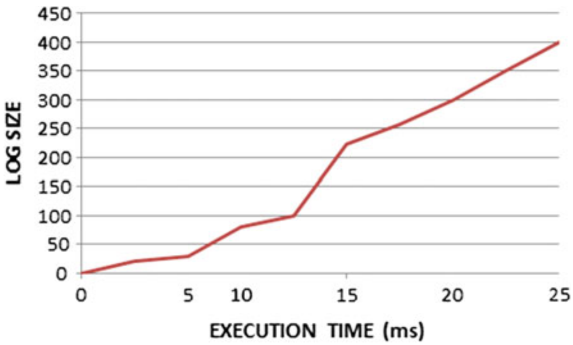


Fig. 4 Performance of our approach

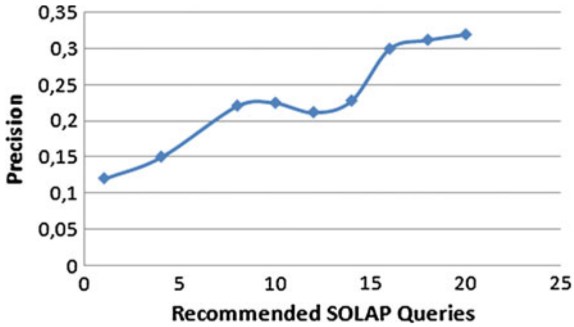


Fig. 5 Precision of *Spatial-OLAP-RS*

In order to calculate the precision indicator, we propose to use a human evaluation technique. We asked 10 persons with SOLAP skills; to launch a set of 5 SOLAP queries to achieve some objectives of data analysis, and retrieve the need information. So, we launch the same SOLAP queries proposed by the 10 persons in our *Spatial-OLAP-RS* system, we obtain a set of recommended SOLAP queries, for each person proposition. The result obtained of the computed precision with our *Spatial-OLAP-RS* system is presented in the Fig. 5.

We note that when the number of recommended queries is less then 5 queries, the precision is low, it's represent 15 %; but it's increase more and more with the augmentation of the number of the recommended queries.

5 Conclusions

In this paper, we proposed a recommendation system in order to help a current user in his exploration of a spatial data cube. For that purpose, we suggested an approach for generating recommendations of Spatial OLAP queries to the current user. Adding to that, to validate our approach, we evaluated the efficiency of our approach proposed to recommend SOLAP queries. Future work consists in going further in the recommendations. We would like to improve our proposed approach by recommending SOLAP queries based on users behaviors and profiles.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Badard, T.: Lopen source au service du gospatial et de lintelligence daffaires. Geomatics Sciences Department, avril 2011
3. Badard, T., Dubé, E.: Enabling geospatial business intelligence. Geomatics Sciences Department, Sept 2009

4. Bdard, Y., Rivest, S., jose Proulx, M.: Spatial on-line analytical processing (solap): Concepts, architectures, and solutions from a geomatics engineering perspective. In: Data Warehouses and OLAP: Concepts, Architecture, and Solutions, pp. 298–319. Press (2006)
5. Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D.: A personalization framework for OLAP queries. In: Proceedings of DOLAP 2005, ACM 8th International Workshop on Data Warehousing and OLAP, pp. 9–18, Bremen, Germany, 4–5 Nov 2005
6. Bellatreche, L., Mouloudi, H., Giacometti, A., Marcel, P.: Personalization of MDX queries. In: 22èmes Journées Bases de Données Avancées, BDA 2006, Lille, Actes (Informal Proceedings), 17–20 octobre 2006
7. Bellogin, A., Castells, P., Cantador, I.: Precision-oriented evaluation of recommender systems: an algorithmic comparison. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pp. 333–336. ACM, New York, NY, USA (2011). http://doi.acm.org/10.1145/2043932.2043996
8. Brouard, C.: Comparaison du modèle vectoriel et de la pondération tf*idf associée avec une méthode de propagation d'activation. In: CORIA, pp. 1–10. Neuchâtel, France, Apr 2013
9. Bruns, H.T., Egenhofer, M.J.: Similarity of spatial scenes. In: 7th Symposium on Spatial Data Handling, pp. 31–42 (1996)
10. Clementini, E., Felice, P.D.: A comparison of methods for representing topological relationships. Inf. Sci. Appl. **3**(3), 149–178 (1995)
11. Egenhofer, M.J., Al-Taha, K.K.: Reasoning about gradual changes of topological relationships. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio-Temporal Reasoning in Geographic Space: Proceedings of the International Conference GIS, pp. 196–219. Springer, Berlin, Heidelberg (1992)
12. Fablet, R., Bouthemy, P.: Motion-based feature extraction and ascendant hierarchical classification for video indexing and retrieval. In: Proceedings of the 3rd International Conference on Visual Information Systems, VISual99. Lecture Notes in Computer Science, vol. 1614, pp. 221–228. Springer Verlag, Amsterdam, The Netherland (1999)
13. Franklin, C.: An introduction to geographic information systems: linking maps to databases. Database **15**(2), 12–21 (1992)
14. Glorio, O., Mazón, J.N., Garrigós, I., Trujillo, J.: A personalization process for spatial data warehouse development. Decis. Support Syst. **52**(4), 884–898 (2012)
15. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
16. Li, B., Fonseca, F.: Tdd: a comprehensive model for qualitative spatial similarity assessment. Spat. Cogn. Comput. **6**(1), 31–62 (2006)
17. Li, J.Z., Zsu, M.T., Tamer, M.: Point-set topological relations processing in image databases. In: In First International Forum on Multimedia and Image Processing, pp. 1–54 (1998)
18. Egenhofer, M.J., Franzosa, R.D..: Point-set topological spatial relations. Int. J. Geogr. Inf. Syst. **2**(5), 161–174 (1991)
19. Marketos, G.: Data warehousing & mining techniques for moving object databases. Ph.D. thesis, Department of Informatics, University of Piraeus (2009)
20. Negre, E.: Exploration collaborative de cubes de donnes. Ph.D. thesis, Universit Franois Rabelais of Tours, France (2009)
21. Rivest, S., Bdard, Y., Proulx, M., Nadeau, M., Hubert, F., Pastor, J.: Solap technology: merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. J. Photogram. Remote Sens. (ISPRS) **60**(1), 17–33 (2005)
22. Sapia, C.: On modeling and predicting query behavior in olap systems. In: Proceedings of INTL Workshop on Design and Management of Data Warehouses (DMDW 99), SWISS LIFE, pp. 1–10 (1999)
23. Sapia, C.: Promise: predicting query behavior to enable predictive caching strategies for olap systems. In: Kambayashi, Y., Mohania, M., Tjoa, A. (eds.) Data Warehousing and Knowledge Discovery. Lecture Notes in Computer Science, vol. 1874, pp. 224–233. Springer, Berlin Heidelberg (2000)

24. Sapia, C., Alexander, F., Erlangen-nrnberg, U.: Promise: modeling and predicting user behavior for online analytical processing applications. Ph.D. thesis, Technische Universitt Mnchen (2001)
25. Sarawagi, S.: Explaining differences in multidimensional aggregates. In: Proceedings of the 25th International Conference on Very Large Data Bases, VLDB '99, pp. 42–53. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
26. Sarawagi, S.: User-adaptive exploration of multidimensional data. In: VLDB, pp. 307–316. Morgan Kaufmann (2000)
27. Sathe, G., Sarawagi, S.: Intelligent rollups in multidimensional olap data. In: Proceedings of the 27th International Conference on Very Large Data Bases, VLDB '01, pp. 531–540. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
28. Stefanovic, N., Han, J., Koperski, K.: Object-based selective materialization for efficient implementation of spatial data cubes. IEEE Trans. Knowl. Data Eng. **12**(6), 938–958 (2000)