Improving the Understandability of OLAP Queries by Semantic Interpretations

Carlos Molina[1,⋆], Belen Prados-Suárez[2], Miguel Prados de Reyes[3], and Carmen Peña Yañez[3]

[1] Department of Computer Sciences, University of Jaen, Jaen, Spain
carlosmo@ujaen.es
[2] Department of Software Engineering, University of Granada, Granada, Spain
belenps@ugr.es
[3] Computer Science Department, San Cecilio Hospital, Granada, Spain
{prados,camenpy}@decsai.ugr.es

Abstract. Everyday methods providing managers with elaborated information making more comprehensible the results obtained of queries over OLAP systems are required. This problem is relatively recent due to the huge amount of information they store, but so far there are few proposals facing this issue, and they are mainly focused on presenting the information to the user in a comprehensible language (natural language). Here we go further and introduce a new mathematical formalism, the *Semantic Interpretations*, to supply the user not only understandable responses, but also semantically meaningful results.

Keywords: Queries interpretation, OLAP, Fuzzy Logic, Semantic Interpretation.

1 Introduction

Nowadays more enterprises and big organizations require advanced methods providing managers with elaborated and comprehensible information. It is especially relevant in the cases of organizations working over OLAP systems due to the immense amount of information that is stored using datacubes.

This problem is relatively recent so there are not a lot of proposals that face this problem. Most of the existing techniques are focused on presenting the information to the user in a comprehensible language for him/her; i.e. in natural language. This is the case of the linguistic summary methods, that analyze great amount of data to provide the user with results of the *"Q of the X verify the*

⋆ The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía) under project P07-TIC03175 "Representación y Manipulación de Objetos Imperfectos en Problemas de Integración de Datos: Una Aplicación a los Almacenes de Objetos de Aprendizaje", the Spanish Government (Science and Innovation Department) under project TIN2009-08296 and also by project UJA11/12/56 from the University of Jaen.

H.L. Larsen et al. (Eds.): FQAS 2013, LNAI 8132, pp. 176–185, 2013.

property Y" structure, where Q is a quantifier. However this is not enough when the user needs the result of the query to be semantically meaningful.

An example of this situation takes place when a manager of a group of health centers has to evaluate the performance of the medical doctors. This manager may query about the number of patients that are attended by a given doctor, obtaining, as a result the number of 15 patients per day. This value does not show whether this doctor works a lot or, otherwise, attends to very few patients. Therefore, it would be necessary to perform the query comparing the results with the attendance values of the other medical doctors working at the same center. With it the manager may get the conclusion that all the staff at the same center have a similar productivity; however, he/she still does not know if it is a good productivity or not: the value obtained does not have the same meaning if the health center attends to a small population than if it is at a big crowded city. Hence, to know if this number is appropriated, it would be necessary to perform a query comparing this value with the ones of the medical doctors working at other health centers with similar characteristics. In other words: 15 patients/day may be a good rate in a small center (where the productivity uses to be medium) but a bad rate in a big hospital (where the average productivity uses to be high or very high).

In this example the same user has performed three different queries over the same data but with distinct purposes, each requiring a different interpretation according to the granularity of the information with which this data is compared.

The research field closer to the problem of the meaning of the queries is, as mentioned above, the linguistic summary field. According to Bouchon-Meunier, B. & Moyse [2], proposals in this scope can be categorized in two groups. On the one hand can be found the proposals using fuzzy logic quantifiers [14,17,15,7,9,1,12,11]. On the other hand, proposals base on nature languages generation (NLG) [16,13,6,8,4,5].

Nevertheless, all of these techniques doesn't take into account the granularity of the information and just tell the user "how many of the X verify Y", when what the user really wants is to analyze the same data item from different points of view (alone, compared with a small set or with a bigger set) each with a different meaning.

This is why in this paper we introduce the concept of Semantic Interpretation of the results of queries on a datacube. To this purpose in section 2 we present the multidimensional model used as reference, whereas in section 3 we describe then notion of semantic interpretations. Next section presetns the adapted multidimensional model including the Semantic Interpretations. In Section 5 an ilustrative example is shown. The last section presents the main conclusions.

2 Multidimensional Model

The base for the semantic interpretation is a multidimensional model to store the data and query it. In this section we briefly present the model. A detail definition can be found in [10].

2.1 Multidimensional Structure

In this section we present the structure of the fuzzy multidimensional model.

Definition 1. *A dimension is a tuple* $d = (l, \leq_d, l_\perp, l_\top)$ *where* $l = l_i, i = 1, ..., n$ *so that each* l_i *is a set of values* $l_i = \{c_{i1}, ..., c_{in}\}$ *and* $l_i \cap l_j = \emptyset$ *if* $i \neq j$, *and* $\leq_d$ *is a partial order relation between the elements of* l *so that* $l_i \leq_d l_k$ *if* $\forall c_{ij} \in l_i \Rightarrow \exists c_{kp} \in l_k / c_{ij} \subseteq c_{kp}$. $l_\perp$ *and* $l_\top$ *are two elements of* l *so that* $\forall l_i \in l \; l_\perp \leq_d l_i \leq_d l_\top$.

We denote level to each element l_i. To identify the level l of the dimension d we will use $d.l$. The two special levels $l_\perp$ and $l_\top$ will be called *base level* and *top level* respectively. The partial order relation in a dimension is what gives the hierarchical relation between levels.

Definition 2. *For each pair of levels* l_i *and* l_j *such that* $l_j \in H_i$, *we have the relation* $\mu_{ij} : l_i \times l_j \to [0, 1]$ *and we call this the* **kinship relation**.

If we use only the values 0 and 1 and we only allow an element to be included with degree 1 by an unique element of its parent levels, this relation represents a crisp hierarchy. If we relax these conditions and we allow to use values in the interval [0,1] without any other limitation, we have a fuzzy hierarchical relation.

Definition 3. *We say that any pair* (h, α) *is a* **fact** *when* h *is an m-tuple on the attributes domain we want to analyze, and* $\alpha \in [0, 1]$.

The value α controls the influence of the fact in the analysis. The imprecision of the data is managed by assigning an α value representing this imprecision. Now we can define the structure of a fuzzy DataCube.

Definition 4. *A DataCube is a tuple* $C = (D, l_b, F, A, H)$ *such that* $D = (d_1, ..., d_n)$ *is a set of dimensions,* $l_b = (l_{1b}, ..., l_{nb})$ *is a set of levels such that* l_{ib} *belongs to* d_i, $F = R \cup \emptyset$ *where* R *is the set of facts and* $\emptyset$ *is a special symbol,* H *is an object of type history,* A *is an application def ined as* $A : l_{1b} \times ... \times l_{nb} \to F$, *giving the relation between the dimensions and the facts defined.*

For a more detailed explication of the structure and the operations over then, see [10].

2.2 Operations

Once we have the structure of the multidimensional model, we need the operations to analyse the data in the datacube. In this section we present the elements needed to apply the normal operations (roll-up, drill-down, pivto and slice).

Definition 5. *An aggregation operator* G *is a function* $G(B)$ *where* $B = (h, \alpha)/(h, \alpha) \in F$ *and the result is a tuple* (h', α').

The parameter that operator needs can be seen as a fuzzy bag ([3]). In this structure there is a group of elements that can be duplicated, and each one has a degree of membership.

Definition 6. *For each value a belonging to d_i we have the set*

$$F_a = \begin{cases} \bigcup_{l_i \in H_{l_i}} F_b / b \in l_j \wedge \mu_{ij}(a,b) > 0 & \text{if } l_i \neq l_b \\ \{h/h \in H \wedge \exists a_1, ..., a_n A(a_1, ..., a_n) = h\} & \text{if } l_i = l_b \end{cases} \quad (1)$$

The set F_a represents all the facts that are related to the value a.

With this structure, the basic operations over datacubes are defined: roll-up, drill-down, dice, slice and pivot (see [10] for definition and properties).

3 Semantic Interpretation

In this section we present the inclusion of semantic interpretations in the fuzzy multidimensional model and the query process using those. Next section presents the structure of the semantic interpretations. Section 3.2 studies the aggregations functions related to the semantic of the results. The last section presents the process of the query.

3.1 Structure

A Semantic Interpretation (*SI*) is a structure associated to each fact. Elements:

- $L = L_1, ..., L_m$: a set of linguistic labels over the basic domain. The set has not to be a partition but this characteristic is desirable.
- $f_a(L, c)$: a function to adapt the labels in L to a cardinality c. As c can be a fuzzy set the function has to be able to work with this kind of data. The function f_a has to be continue and monotone.
- $G = G_1, ..., G_n$: a set indicating the aggregation functions that keep unaltered the meaning.

Multiple *SI* can be associated to each measure. On each fact we have to store as a metadata the cardinality associated to the value. This value means the number of values that were aggregated to obtain this value but depends on the aggregation function used. Next section present the study about this value according the the type of aggregations applied.

When a value is going to be shown, the system applies the semantic interpretation to translate the value into a label. In this process we can differentiate two different approaches:

- *Independence interpretation.* In this situation each value is studied without considering the context (the rest of the values) so we obtained an independence interpretation of the value. In this case, the cardinality to adapt the labels is the one store in the value.
- *Relative interpretation.* In this case, the values are compared with the other facts in the query so the interpretation is relative to the complete query so the cardinality to adapt the labels depends on the complete set of values. In this case, the system calculates the average cardinality of all the values and uses this value to adapt the labels.

3.2 Aggregation Functions

Aggregation functions have an important role in the query process and in the semantic of the results. In this section we will study the different aggregation function type we can find according to the cardinality of the results and if a change of semantic occurs.

Let be a set of value $V = v_1, ..., v_m$, which set of cardinality is $C = n_1, ..., n_m$, considering these two factors, we can classify the functions in three categories:

- *Aggregators.* These functions aggregate the values and the cardinality is the sum of the cardinalities of each value

$$c = \sum_{i}^{m} n_i \tag{2}$$

  The only aggregation function that satisfies this behaviour is the *sum*.
- *Summaries.* In that case, the functions take a set of values and obtain a value that summaries the complete set. Then the cardinality has to represent the average cardinality of the values.

$$c = \frac{\sum_{i}^{m} n_i}{m} \tag{3}$$

  Most of the statistic indicators are in this category (*maximum, minimum, average, median, percentiles*, etc.).
- *Others.* These functions represent a complete change of the semantic of the values so the result has to be considered in a new domain. In that case, the cardinality should be established to 1.

$$c = 1 \tag{4}$$

  In this category we found functions like the *variance* or the *count*.

Once we have studied the aggregation functions we have all the elements to show the query process with the SIs in datacubes

4 Semantic Interpretations for DataCubes

One we have define the formalism for SI, we introduce this concept in the multidimensional model previously define. To be able to use the SI we need to add information to each fact that represent the cardinality for the concrete value. So, we have to redefine the fact (definition 3) including the metadata.

Definition 7. *We say that any tuple* (h, α, m) *is a* **fact** *when* h *is an* m*-tuple on the attributes domain we want to analyze,* $\alpha \in [0, 1]$ *and* m *the metatada for these values.*

In the m element of the structure we introduce the cardinality c needed for the SI. This value is updated each time we query the datacube so, the aggregators have to work with this metadata.

Definition 8. *An aggregation operator* G *is a function* $G(B)$ *where* $B = (h, \alpha, m)/(h, \alpha, m) \in F$ *and the result is a tuple* (h', α', m').

4.1 Query Process

In this section we present the query process considering the use of the *SI*. We can differentiate two phases on the query process: the OLAP query over the DataCube and the report with the results. In both phases the *SI* are involved in a different way. Let show the process on each one:

- *Query over the DataCube*. In this phase is where the values are calculated. Inside this process we have to calculate the metatada of each one so, in the next phase, the values can be shown using the *SI*. The cardinality is calculated over each value considering the aggregation function used as shown in section 3.2. In this process the system has to control if the semantic has changed. On each value the system check if the aggregation function used is in the set G of each *SI*. If the function is not included, then this *SI* is deleted. In next phase (the report) the user can used only the *SI*s that satisfies this restriction.
- *Report*. Once the query has finished the result is shown to the user in a report. In this process the user has to choose the way to represent the values (the *SI* to use) and the interpretation (independence or relative as shown in section 3.1). After these steps, the system adapts the labels of each value using the f_a functions and the right cardinality (the absolute or the average).

4.2 Learning the f_a Functions

In previous section we have presented the query process using *SI*. One of the phase adapts the labels in L so the labels are fitted to the new cardinality. This process is carried out using the f_a function. The quality of the result will depend on this function, so an important point is the process to define it. Asking the user for that function is not always possible because most of the times the user is not able to use a mathematic expression to define his/her interpretations. So, we propose to learn the functions. The learn process will have the followings steps:

1. First we ask the user for an interpretation over the basic domain so we can define the set L of labels over it.
2. To learn the function now the system runs some queries over the DataCube showing the results.
3. For each query the system ask the user to associate a label of L to the value. The system stores the associations and the cardinalities of each value.
4. With these associations the system tries to fit a function that satisfies the interpretation with the corresponding cardinality. In this process, the system will try continue and monotone functions to adapt the labels. If the fitted function has good quality (the adapted labels correspond to the labels associated by the user) the process end. In other case, the process go but to point 2 to show more queries so the system have more data to fit the function.

5 Example

In this section we will present a small example to show in details the propose method. Let suppose we have a simple datacube only with two dimensions (time and centre) and only one measure (number of patients). The hierarchies for both dimensions are shown in Figure 1.

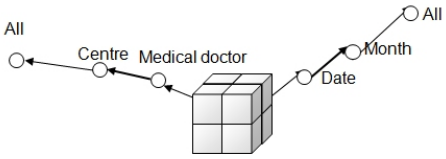

Fig. 1. Example datacube

For the measure we define a *SI* indicating if the number of patient attended by a doctor is Low, Normal or High. At base level (doctor and day) the fuzzy partition is shown in Figure 2.

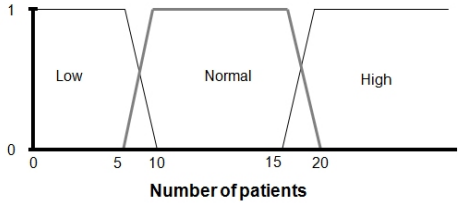

Fig. 2. Fuzzy partition over the measure Number of Patients

The *SI* is valid for aggregations like sum and average. The last aspect to define is the f_a function. In this example we suppose is lineal and just we multiply the points of the fuzzy label by the new cardinality (e.g. if Low is define as $(0, 0, 5, 10)$ for one doctor in a day, for two doctors the label will be $(2 \cdot 0, 2 \cdot 0, 2 \cdot 5, 2 \cdot 10) = (0, 0, 10, 20)$). Let suppose that we have two centres with different size. One (C1) is placed in a city and there are 500 medical doctors in the staff. The second one (C2) is placed in a small village and only 10 doctors are working in that centre. If a manager asks the system to calculate the number of patients attended by both centres each month we can get the Table 1.

The result shows very different results for each centre and it is not easy interpretable due to differences in the size of each one. Let apply our proposal and obtain the label that best represent each value. For centre C1 we have to calculate the cardinality of the result so we can adjust the labels. We have the datacube defined in the granularity doctor by day, so, for each month we have aggregate the values for 500 medical doctors a 20 working days for month, so

Table 1. Query results

Centre	Month	Patients
C1	January	125,000
C1	February	130,000
...	...	...
C2	January	4,200
C2	February	5,000
...	...	...

the cardinality is $500 \cdot 20 = 10.000$. We adjust the fuzzy partition for this new cardinality as shown in Figure 3. In the case of centre C2 then the cardinality is $10 \cdot 20 = 200$ (Figure 4).

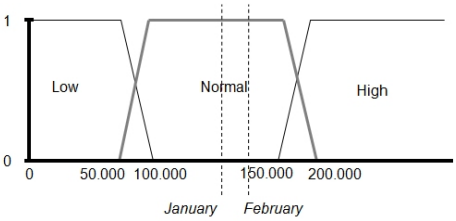

Fig. 3. Labels adaptation for query for centre C1

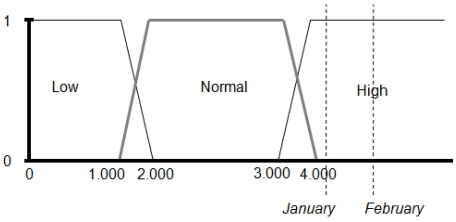

Fig. 4. Labels adaptation for query for centre C2

Table 2. Query results using *Semantic Interpretation*

Centre	Month	Patients	Label
C1	January	125,000	Normal
C1	February	130,000	Normal
...	...	...	...
C2	January	4,200	High
C2	February	5,000	High
...	...	...	...

In the figures we have indicated the values for the Table 1, so we have the labels associated to each result. In Table 2 we have added the label associated to each value.

If we use the *SI* in the example we see how the values are adapted so the user has the interpretation of the values directly.

6 Conclusions

In this paper we have introduced the new concept of Semantic Interpretation, that provides the OLAP systems with the new capability of querying about the same given item with different purposes obtaining in each case a result with a different meaning. With our proposal, the semantic of the results of the query can be distinct and adapted to the needs of the user, by taking into account the granularity of the information considered.

References

1. Bosc, P., Lietard, L., Pivert, O.: Extended functional dependencies as a basis for linguistic summaries. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 255–263. Springer, Heidelberg (1998), http://dl.acm.org/citation.cfm?id=645802.669203
2. Bouchon-Meunier, B., Moyse, G.: Fuzzy linguistic summaries: Where are we, where can we go? In: 2012 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr), pp. 1–8 (2012)
3. Delgado, M., Martín-Bautista, M., Sánchez, D., Vila, M.: On a characterization of fuzzy bags. In: De Baets, B., Kaynak, O., Bilgiç, T. (eds.) IFSA 2003. LNCS, vol. 2715, pp. 119–126. Springer, Heidelberg (2003)
4. Goldberg, E., Kittredge, N.D., Using, R.I.: natural-language processing to produce weather forecasts. IEEE Expert 9, 45–53 (1994)
5. Portet, F., Reiter, E., Hunter, J., Sripada, S.: Automatic generation of textual summaries from neonatal intensive care data. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. LNCS (LNAI), vol. 4594, pp. 227–236. Springer, Heidelberg (2007)
6. Yu, J., Reiter, E., Sripada, J.H., Sumtime-turbine, S.: A knowledge-based system to communicate gas turbine time-series data. In: The 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems (2003)
7. Kacprzyk, J., Wilbik, A.: Linguistic summaries of time series using a degree of appropriateness as a measure of interestingness. In: Ninth International Conference on Intelligent Systems Design and Applications, ISDA 2009, pp. 385–390 (2009)
8. Danlos, L., Combet, F.M., Easytext, V.: an operational nlg system. In: ENLG 2011, 13th European Workshop on Natural Language Generation (2011)
9. Lietard, L.: A new definition for linguistic summaries of data. In: IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), pp. 506–511 (2008)
10. Molina, C., Rodriguez-Ariza, L., Sanchez, D., Vila, A.: A new fuzzy multidimensional model. IEEE Transactions on Fuzzy Systems 14(6), 897–912 (2006)

11. Castillo-Ortega, R., Marín, N., Sánchez, D.: A fuzzy approach to the linguistic summarization of time series. Journal of Multiple-Valued Logic and Soft Computing 17(2,3), 157–182 (2011)
12. Rasmussen, D., Yager, R.R.: Finding fuzzy and gradual functional dependencies with summarysql. Fuzzy Sets Syst. 106(2), 131–142 (1999), http://dx.doi.org/10.1016/S0165-0114(97)00268-6
13. Sripada, S., Reiter, E., Davy, I.: Sumtime-mousam: Configurable marine weather forecast generator. Tech. rep. (2003)
14. Yager, R.R.: A new approach to the summarization of data. Information Sciences 28, 69–82 (1982)
15. Yager, R.R.: Fuzzy summaries in database mining. In: Proceedings the 11th Conference on Artificial Intelligence for Applications (1995)
16. Yseop: Faire parler les chiffres automatiquement, http://www.yseop.com/demo/diagFinance/FR/
17. Zadeh, L.: A computational approach to fuzzy quantifiers in natural languages. Computers & Mathematics with Applications 9, 149–184 (1983)