F-CUBE FACTORY: A FUZZY OLAP SYSTEM FOR SUPPORTING IMPRECISSION

Miguel Delgado1 Carlos Molina2∗ Lázaro Rodríguez-Ariza3 Daniel Sánchez1 M. Amparo Vila1
1.Dpt. Computer Science and A.I., School of Computer Science, University of Granada
2.Dpt. Computer Science, School of Computer Science, University of Jaén
3.Dpt. Financial Economy and Accounting, School of Management, University of Granada
Email: mdelgado@decsai.ugr.es, carlosmo@ujaen.es, lazaro@ugr.es, daniel@decsai.ugr.es, vila@decsai.ugr.es

ABSTRACT: The special needs of the OLAP technology was the main cause of the use of a multidimensional view of the data. As a result of the use of this technology in new fields of knowledge (e.g. medical data) and the integration of data from heterogeneous sources, it has become necessary for multidimensional models to support new needs. To model complex or nor well defined domains or to integrate data from semi/non-structured sources (e.g. Internet) or with incompatibilities in their schemata is complicated using crisp multidimensional models. In these situations we need a model able to manage imprecision in the structures and data as a result of the modelling and/or integration. In this paper we present an OLAP system based on a fuzzy multidimensional model that uses fuzzy logic. The use of fuzzy structures (hierarchies and facts) and the definition of the normal OLAP operations (roll-up, drill-down, dice, slice and pivot) enable the model to manage the imprecision of these situations, hiding at the same time the complexity to the user by means of user views.

Keywords: Multidimensional Model, OLAP, Fuzzy Logic

1 INTRODUCTION

Since the appearance of the OLAP technology ([5]) different proposals have been made to give support to the special necessities of this technology. In the literature we can see two different approaches. One of this is to extend the relational model to support the structures and operations typical of OLAP. The first one following this idea can be found in [12]. From then on, more proposes have appeared ([13]) and most of the present relational systems include extension to represent datucubes and operate over them. The other approach is to develop new models using a multidimensional view of the data. Many authors proposed model in this way ([1, 3, 4, 15]). In the early 70’s, the necessity of flexible models and query languages to manage the ill-defined nature of information in DSS is identified ([11]). Nowadays, the application of the OLAP technology to other knowledge fields (e.g. medical data) and the use of semi-structured (e.g. XML) and non-structured (e.g. plain text) sources introduce new requirements to the models. Now the systems need to manage imprecision in the data and facts) and the definition of the normal OLAP operations (roll-up, drill-down, dice, slice and pivot) enable the model to manage the imprecision of these situations, hiding at the same time the complexity to the user by means of user views.

Nevertheless, these models focus the imprecision in the facts. Continuing using rigid hierarchies made very difficult to model some problems that can be translate into loss of information when we need to merge data from different sources with some incompatibilities in their schemata.

What we present in this paper is an OLAP system based on a fuzzy multidimensional model that uses fuzzy relation to model the hierarchies as well as fuzzy facts ([7]). Merging information given by experts may improve the multidimensional schemata in some situations. To help in this task, the model allows to define the hierarchical relation using linguistic labels ([8]) that is more intuitive for the expert than using concrete values.

In the next section we present the multidimensional model under F-CubeFactory system. In section 3 the system is presented. The last section is dedicated to conclusions and future work.

2 FUZZY MULTIDIMENSIONAL MODEL

In this section we briefly introduce the fuzzy multidimensional model. A more detailed description can be found in [7, 8]. Here we only present the main concepts needed to understand the model implemented.

2.1 Fuzzy Multidimensional structure

Definition 1 A dimension is a tuple $d = (l_i, \leq_d, l_\bot, l_\top)$ where $l_i = l_i, i = 1,..., n$ so that each $l_i$ is a set of values $l_i = \{c_{i1}, ..., c_{im}\}$ 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 $\forall c_{ij} \in l_i \exists c_{kp} \in l_k / c_{ij} \subseteq c_{kp}$. $l_\bot$ and $l_\top$ are two elements of $l$ so that $\forall l_i \in l I l_\bot \leq_d l_i \leq_d l_\top$.

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

**Definition 2** For each dimension \( d \), the domain is \( \text{dom}(d) = \bigcup l_i \).

In the example above the domain of the dimension \( \text{Age} \) is \( \text{dom} \text{(Age)} = \{1, ..., 100, \text{Young}, \text{Adult}, \text{Old}, \text{Yes}, \text{All} \} \).

**Definition 3** For each \( l_i \), the set \( H_i = \{l_j/l_j \neq l_i \land l_j \leq d_l \land \neg \exists l_k \; l_j \leq d_k \leq d_l \} \)

and we call this the set of children of the level \( l_i \).

Using the same example, the set of children of the level \( \text{All} \) is \( H_{\text{All}} = \{\text{Group}, \text{Legal age}\} \). In all dimensions that we can define this set for the base level will be always the empty set, as you can see from the definition of set of children.

**Definition 4** For each pair of levels \( l_i \) and \( l_j \) such that \( l_j \in H_i \), we have the relation

\[
\mu_{ij} : l_i \times l_j \rightarrow [0, 1]
\]

and we call this the kinship relation.

The degree of inclusion of the elements of a level in the elements of their parent levels can be defined using this relation. If we use only the values 0 and 1 and we only allow an element to be include with degree 1 by a unique element of its parent levels, this relation represents a crisp hierarchy.

Following the example, the relation between the levels \( \text{Legal age} \) and \( \text{Age} \) is of this type. The kinship relation in this situation is

\[
\mu_{\text{Legal age}, \text{Age}}(\text{Yes}, x) = \begin{cases} 1 & \text{if } x \in [18, 100] \\ 0 & \text{in other case} \end{cases}
\]

\[
\mu_{\text{Legal age}, \text{Age}}(\text{No}, x) = \begin{cases} 1 & \text{if } x \in [1, 17] \\ 0 & \text{in other case} \end{cases}
\]

If we relax these condition and we allow to use values in the interval \([0,1]\) without any other limitation, we have a fuzzy hierarchical relation. This allows represent several hierarchical relations in a more intuitive way. An example can be seen in the figure 2 where we present the group of ages according to linguistic labels. Furthermore, this fuzzy relation allows to define hierarchies in which there is imprecision in the relationship between elements in different levels. In this situation, the value in the interval shows the degree of confidence in the relation.

**Figure 2** Kinship relation between levels \( \text{Group} \) and \( \text{Age} \)

For each pair of levels \( l_i \) and \( l_j \) of the dimension \( d \) such that \( l_j \leq d_l \land l_j \neq l_i \), the relation \( \eta_{ij} : l_i \times l_j \rightarrow [0, 1] \) is defined as

\[
\eta_{ij}(a, b) = \bigoplus_{l_k \in H_l} (\mu_{l_k}(a, c) \otimes \mu_{l_j}(c, b)) \quad \text{in other case}
\]

where \( \otimes \) and \( \oplus \) are a t-norm and a t-conorm, respectively, or operators from the families MOM and MAM defined by Yager [19], which include the t-norms and t-conorms, respectively. This relation is called the extended kinship relation.

This relation gives us information about the degree of relation between two values in different levels inside the same dimension. To obtain this value, it considers all the possible paths between the elements in the hierarchy. Each one is calculated aggregating the kinship relation between elements in two consecutive levels using a t-norm. Then the final value is the aggregation of the result of each path using a t-conorm.

As an example, we will show how to calculate the value of \( \eta_{\text{All}, \text{Age}}(\text{All}, 25) \). In this situation we have two different paths. Let see each one:

- **All - Legal age - Age.** In the figure 3.a you can see the two ways to get to 25 from All going pass the level legal age. The result of this path is \((1 \otimes 1) \oplus (1 \otimes 0)\).

- **All - Group - Age.** This is a situation very similar to the previous one. In the figure 3.b you can see the three different paths going through the level Group. The result of this path is \((1 \otimes 0.7) \oplus (1 \otimes 0.3) \oplus (1 \otimes 0)\).

Now we have to aggregate these two values using a t-conorm to obtain the result. If we use the maximum as t-norm and the minimum as t-conorm, the result is \((1 \otimes 0.7) \oplus (1 \otimes 0.3) \oplus (1 \otimes 0)\) = \((1 \otimes 0) \oplus (0.7 \otimes 0.3) \oplus 0\) = \(1 \otimes 0.7 \oplus 0\). So the value of \( \eta_{\text{All}, \text{Age}}(\text{All}, 25) \) is 1.

**Definition 5** For each pair of levels \( l_i \) and \( l_j \) of the dimension \( d \) such that \( l_j \leq d_l \land l_j \neq l_i \), the relation \( \eta_{ij} : l_i \times l_j \rightarrow [0, 1] \) is defined as

\[
\eta_{ij}(a, b) = \bigoplus_{l_k \in H_l} (\mu_{l_k}(a, c) \otimes \mu_{l_j}(c, b)) \quad \text{in other case}
\]

where \( \otimes \) and \( \oplus \) are a t-norm and a t-conorm, respectively, or operators from the families MOM and MAM defined by Yager [19], which include the t-norms and t-conorms, respectively. This relation is called the extended kinship relation.

**Figure 3** Example of the calculation of the extended kinship relation. a) path \( \text{All} - \text{Legal age} - \text{Age} \) b) path \( \text{All} - \text{Group} - \text{Age} \)

Now we have to aggregate these two values using a t-conorm to obtain the result. If we use the maximum as t-norm and the minimum as t-conorm, the result is \((1 \otimes 0.7) \oplus (1 \otimes 0.3) \oplus (1 \otimes 0)\) = \((1 \otimes 0) \oplus (0.7 \otimes 0.3) \oplus 0\) = \(1 \otimes 0.7 \oplus 0\). So the value of \( \eta_{\text{All}, \text{Age}}(\text{All}, 25) \) is 1.

**Definition 6** We say that any pair \((h, \alpha)\) is a fact when \( h \) is an \( m \)-tuple on the attributes domain we want to analyze, and \( \alpha \in [0, 1] \).

The value \( \alpha \) controls the influence of the fact in the analysis. The imprecision of the data is manage by assigning an \( \alpha \) value representing this imprecision. When we operate with the facts, the aggregation operators have to manage this values in the calculations. The arguments for the operator can be seen as fuzzy bag due to they are a set of values with a degree in the
interval \([0,1]\) than can be duplicated. For a characterization of fuzzy bag see [6]. The result of the aggregation has to be a fact too. So, in the fuzzy case the definition of aggregation operators is the following.

**Definition 7** Been \(\tilde{B}(X)\) all the possible fuzzy bags defined using elements in \(X\), \(\tilde{P}(X)\) the fuzzy power set of \(X\), and \(D_x\) a numeric or natural domain, we define an aggregation operator \(G\) as a function \(G : \tilde{B}(D_x) \rightarrow \tilde{P}(D_x) \times [0,1]\).

When we apply an aggregation operator, we resume the information of a bag of values into an unique value. Not always is possible to undo this operations. So if we want to undo operations that reduce the level of detail in a DataCube, we need something to prevent this problem. So we define the object history that stores the aggregation states of a DataCube.

**Definition 8** An object of type history is the recursive structure

\[
H^0 = \Omega \quad H^{n+1} = (A, l_b, F, G, H^n)
\]

where \(\Omega\) is the recursive clause, \(F\) is the fact set, \(l_b\) is a set of levels \((l_{b1}, \ldots, l_{bn})\), \(A\) is an application from \(l_b\) to \(F\) \((A : l_b \rightarrow F)\), \(G\) is an aggregation operator.

Now we can define the structure of a fuzzy DataCube.

**Definition 9** A DataCube is a tuple \(C = (D, l_b, F, \Lambda, H)\) such that \(D = (d_1, \ldots, d_n)\) is a set of dimensions, \(l_b = (l_{b1}, \ldots, l_{bn})\) is a set of levels such that \(l_{bi}\) 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 defined as \(A : l_{b1} \times \ldots \times l_{bn} \rightarrow F\), giving the relation between the dimensions and the facts defined.

If \(\bar{c} = (c_1, \ldots, c_n)\) we have \(A(\bar{c}) = \emptyset\), this means that there isn’t defined a fact for this combination of values.

**Definition 10** We say a DataCube is basic if \(l_b = (l_{1\perp}, \ldots, l_{n\perp})\) and \(H = \Omega\).

### 2.2 The Linguistic DataCube

One possibility to extend and improve the hierarchies of multidimensional schemes is to incorporate the knowledge of experts about that hierarchies being usually given in a linguistic manner, but let us point out that in many cases the experts not only use fuzzy or imprecise classes in the nodes but they also express vaguely the hierarchical relations themselves. It is obvious that considering a hierarchy with linguistic assessment is equivalent to consider the kinship and the extended kinship relations to be linguistic. In the following \([0,1]\) will denote the set of fuzzy numbers on the unit interval.

In this section we introduce the main concepts of linguistic hierarchies. For a deeper study see [8]. The first concept that have to be redefined is the kinship relation to consider the linguistic relations.

**Definition 11** For each pair of levels \(l_i\) and \(l_j\) with \(l_i \in H_i\) there exist a relation \(\tilde{\mu}_{ij} : l_i \times l_j \longrightarrow [0,1]\) which is called the kinship relation.

In the linguistic case, to aggregate the linguistic relations we need to use operators with a behavior similar to t-norms and t-conorms with some limitations (see [8]). Several definitions of linguistic aggregation operators may be found in the literature (see [10], [16]). We have defined another operator ([8]), that can act as t-norm or t-conorm over linguistic values according to a parameter, using a fuzzy extension of the OWA operator ([20]).

**Definition 12** An aggregation operator \(A^\text{OM}_\beta\) is a function \(A^\text{OM}_\beta : [0,1]^n \rightarrow [0,1]\) defined as

\[
A^\text{OM}_\beta(a_1, \ldots, a_n) = OWA_w^\text{OM}((a_1, \ldots, a_n))
\]

where \(OWA_w^\text{OM}\) is a Fuzzy OWA operator with ranking method \(OM\), \(w\) a weight vector having \(w_1 = \beta\), \(w_n = 1 - \beta\), and \(w_i = 0\) for all \(i \in \{2, \ldots, n - 1\}\).

Now we can defined the extended kinship relation.

**Definition 13** For any levels \(l_i\) by \(l_j\) on dimension \(d\), such that \(l_i \leq l_j\) and \(l_j \neq l_i\), the extended kinship relation \(\tilde{\eta}_{ij} : l_i \times l_j \rightarrow [0,1]\) will be given by

\[
\tilde{\eta}_{ij}(a, b) = \begin{cases} 
\tilde{\mu}_{ij}(a, b) & \text{if } l_j \in H_i \\
A^\text{OM}_\beta(P_{l_j}, \ldots, P_{l_n}) & \text{otherwise}
\end{cases}
\]

where \(l_k \in H_k\) and \(P_{l_k} = A^\text{OM}_\beta(\delta_{c_1}, \ldots, \delta_{c_n})\|c_i \in l_k\), being \(\delta_{c} = A^\text{OM}_\beta(\tilde{\mu}_{sk}(a, c), \tilde{\eta}_{ik}(c, b))\).

### 2.3 Operations

Once we have the structure of the multidimensional model, we need the operations to analyze the data in the DataCube. Over this structure we have defined the normal operations of the multidimensional model:

- **Roll-up:** go up in the hierarchies to reduce the detail level.
- **Drill-down:** go down in the hierarchies to increase the detail level.
- **Dice:** project over the DataCube using a condition.
- **Slice:** reduce the dimensionality of the DataCube.
- **Pivot:** change the order of the dimensions.

For a detail definition of the operation see [7, 8]. In these operations the hierarchical relations are very important. As an example, in the roll-up operation we need to know the facts related with the values at the detail level desired. This set is defined as follow.

**Definition 14** For each value \(c_{ij}\) belonging to \(l_i\) we have the set

\[
F_{c_{ij}} = \begin{cases} 
\bigcup_{l_k \in H_i} \left\{ F_c \mid c_{kp} \in l_k \wedge \tilde{\mu}_{k}(c_{ij}, c_{kp}) > 0 \right\} & \text{if } l_i \neq l_k \\
\{ h/h \in F \wedge \exists c' \in C(c') = c \} & \text{if } l_i = l_k
\end{cases}
\]

where \(c' = (c_{1}, \ldots, c_{ij}, \ldots, c_{n})\). The set \(F_{c_{ij}}\) represents all the facts that are related to the value \(c_{ij}\).
In the case of linguistic hierarchies we use the linguistic kinship relation. So, the definition of \( F_{c_{ij}} \) has to be changed to use it.

**Definition 15** For each value \( c_{ij} \) belonging to \( l_r \) we have the set

\[
F_{c_{ij}} = \left\{ \frac{\mu_{c_{ij}}(c_{ij})}{c_{jk}} \in I_k \cap \mu_{\pi_k}(c_{jk}) \neq 0 \quad \text{if } l_r \neq l_b \\
\{ h/h \in F \cap \exists \; \gamma(c) = h \} \quad \text{if } l_r = l_b \right. \\
\text{where } \gamma = (c_1, c_{ij}, ..., c_n). \quad \text{The set } F_{c_{ij}} \text{ represents all the facts that are related to the value } c_{ij}.
\]

**2.4 User view**

We have presented a structure that manages imprecision by means of fuzzy logic. We need to use aggregation operators on fuzzy bags in order to apply some of the operations presented. Most of the methods previously documented give a fuzzy set as a result. As this situation can make the result difficult to understand and use in a decision process, we propose a two-layer model: one of the layers is the structure presented in the previous section; and the other is defined on this, and its main objective is to hide the complexity of the model and provide the user with a more understandable result. In order to do so, we propose the use of a fuzzy summary operator that gives a more intuitive result but which keeps as much information as possible. Using this type of operator, we shall define the **user view**.

**Definition 16** Given a summary operator \( M \), we define the **user view** of a DataCube \( C = (D, l_b, F, A, H) \) using \( M \) as the structure \( C_M = (D, l_b, F_M, A_M) \) where \( A_M(a_1, ..., a_n) = M(A(a_1, ..., a_n)) \), \( F_M \) is the range of \( A_M \).

We can define as many user views of a DataCube as the number of summary operators used. Therefore, each user can have their own user view with the most intuitive view of data according to their preferences by using a DataCube. As an example of this type of operator, we can use the one proposed in [2]. This operator proposes the use of the fuzzy number that best fits, in the sense of fuzziness, the fuzzy set or fuzzy bag. We can use more simple operators as the **weighted average**. As an example we’ll apply both operators to the fuzzy bag \{1/1, 1/2, 0.9/0.5, 0.8/2.3, 0.2/0.3, 0.1/2.5\}:

- **Linguistic summary.** Using this operator, the result is \((1, 2, 0.5)\) which linguistic expression associated is “more or less between 1 and 2”.

- **Weighted average.** In this situation, the value shown to the user is 1.4.

As you can see, in both case the user get a more intuitive access to the results.

To give a intuitive way to interpret the result is important, as shown by Codd et al. in the 11th OLAP product evaluation rule ([5]). Most of the times the user will understand better a graphic than a table with the results. Present systems use charts to show the result to the decisor. In our model, to provide a graphical way, is even more important due to the fact that to interpret fuzzy values is complicate even to experts in fuzzy logic.

We propose two methods to represent fuzzy numbers in a graphical way as an user view. Both approaches are shown in Figure 4. In Figure 4.a the approach followed is to use a color gradient to represent the membership grade of the values. The other approach (Figure 4.b) consists in change the width of a bar to represent the membership.

Both can be use to construct charts. An example is shown in Figure 5. This example represent fuzzy values related to crisp ones (the labels). In some situation, represent fuzzy values related to fuzzy labels can be interesting. Following the first approach we can do it. So, what we do is to aggregate the membership values in both axis, using a t-norm, and use the result to build the color gradient. Figure 6 shows an example of chart where the labels are defined using linguistic labels.

**3 F-Cube Factory**

The system is completely build using Java language and it was design keeping in mind future extension for the multidimensional model. Now the software implements three DataCubes models:

- **ROLAP model:** the system can manage DataCubes using a relational database to store the DataCube and to obtain the data to build new DataCubes.

- **MOLAP crisp model:** DataCubes are also stored using a purely multidimensional structure implemented in Java.

- **MOLAP fuzzy/linguistic model:** this model implements the fuzzy and linguistic multidimensional model presented. It uses a MOLAP way to manage the fuzzy DataCubes.

We can differentiate two main parts in the system: the server is the one that implements the main functionality, and the clients, which are the interface to the user to the server functionality trying to give a simple and intuitive access to the DataCube. In next section we present some details of each part of the system.
3.1 F-CubeFactory Server

The server architecture is shown in Figure 7. The most important modules in the server are these:

- DataCubes module: this module implements the three DataCube models previously mentioned. It gives a homogenous access to the multidimensional structure to the rest of the modules. One of the main functionalities is the queries. The efficiency is very important because OLAP systems have to give support for ad-hoc queries in a reasonable time. In the fuzzy DataCube this is even more important due to the fact that each query implies the aggregation of a great amount of kinship relations. To improve the efficiency the system precomputes the extended kinship relations from each level to the basic level. This task is carried out when building the fuzzy DataCube. A DataCube is built one time mean while we use the same DataCube for a lot of queries, so the time spent in aggregating the kinship relations is only taken when the user does not suffer the delay.

In this module is included the user views for the fuzzy DataCubes. To add new user views to the server is very easy: you only need to extend a Java class and register in the server configuration. The calculation of a user view is only made the first time the system need the fact and stores it to be used the next times the system needs it.

- Aggregation functions module: This module interact with the previous one when we want to change the detail level, which is translated in a query. It has implemented the normal function for crisp DataCubes (max, min, sum, average and count) and fuzzy ones, using an adaptation of Rundensteiner and Bic’s operators.

**Definition 17** 

Been \( R \) an operator defined by Rundensteiner and Bic ([18]), and \( \tilde{F} \) a fuzzy bag over the facts. We define the operator \( G_R \) as \( G_R(\tilde{F}) = (R(\tilde{F}'), 1) \), where \( \tilde{F}' = \{\alpha/h \text{ such that } (h, \alpha) \in \tilde{F}\} \).

Adding new aggregation function is as easy as in the case of user views.

- Server API module: this module implement the API to access all the functionality in the server. This is the access point for the clients.

3.2 F-CubeFactory Client

The main objectives of the client are:

- The client has to be light enough to be use in a normal personal computer.

- And the most important is that is has to implement an intuitive access to the server functionality.

The client is web based, so the user only need to access to a web site using a normal web browser. Figure 8 shows the aspect of the user interface. The user only has to select the option needed to access to a DataCube without needing to know any DML or DDL language.

The resulted DataCubes of queries are shown to the user using tables (Figure 9) and charts (Figure 5 was built using this functionality) for all type of DataCubes.
4 CONCLUSIONS AND FUTURE WORK

In this paper we have presented an OLAP system prototype that implements a fuzzy multidimensional model to represent imprecision using fuzzy and linguistic hierarchies and fuzzy facts. The system has been developed keeping in mind to add new features in the future. Now we are working on OLAP mining over the multidimensional model presented. The next step is to implement in F-CubeFactory the results of this research and to proof the system using a real case.

REFERENCES


