The Hierarchical Agglomerative Clustering with Gower index: a methodology for automatic design of OLAP cube in ecological data processing context

Lucile Sautot, Bruno Faivre, Ludovic Journaux, Paul Molin

The Hierarchical Agglomerative Clustering with Gower index: a methodology for automatic design of OLAP cube in ecological data processing context

Lucile Sautot[a,c,d,*], Bruno Faivre[a], Ludovic Journaux[b], Paul Molin[c]

[a] *UMR CNRS/uB 6282 Biogéosciences, Université de Bourgogne, 6 bd Gabriel 21000 Dijon, France*
[b] *Laboratoire Informatique, Electronique et Image, UFR Sciences et Techniques, Université de Bourgogne, allée Alain Savary 21000 Dijon, France*
[c] *DSIP, Agrosup Dijon, 26 bd Petitjean 21000 Dijon, France*
[d] *AgroParisTech, 19 avenue du Maine 75732 Paris, France*

Abstract

The OLAP systems can be an improvement for ecological studies. In fact, ecology studies, follows and analyzes phenomenon across space and time and according to several parameters. OLAP systems can provide to ecologists browsing in a large dataset. One focus of current research on OLAP system is the automatic design of OLAP cubes and of data warehouse schemas. This kind of works makes accessible OLAP technology to non Information Technology experts. But to be efficient, the automatic OLAP building must take account into various cases.

Moreover the OLAP technology is based on the concept of hierarchy. Thereby the hierarchical clustering methods are often used by OLAP system designer.

In this article, we propose using hierarchical agglomerative clustering with a metric that comes from ecological studies (the Gower similarity index) to build automatically hierarchical dimensions in an OLAP cube. With this similarity index we can perform a hierarchical clustering on heterogeneous datasets that contains qualitative and quantitative variables.

We offer a prototypical automatic system which builds dimension for an OLAP cube and we measure the performances of this system according to the number of clustered individuals and according to the number of variables used for clustering. Thanks to these measures we can offer an approximation of performances with a large dataset.

Thereby the Gower index in a hierarchical agglomerative clustering permits the management of heterogeneous dataset with missing values in a context of automatic building of OLAP cube. With this methodology, we can build new dimensions based on hierarchies in the data, which are not evident. The data mining methods can complete the expert knowledge during the design of an OLAP cube, because these methods can explain the inherent structure of the data.

Keywords: OLAP; Hierarchical Agglomerative Clustering; Bird Population; Automatic Design

Introduction: use data mining for OLAP cube design

Since 1993, OLAP (On Line Analytical Processing) systems have been proposed to improve decision making process due to analysis of large datasets (Codd et al., 1993). This kind of software is designed to explore easily and quickly multidimensional data (Rivest et al., 2005). The word OLAP can be associated with a process, a kind of system or a kind of data (Jerbi et al., 2009). A basic Relational OLAP (ROLAP) system architecture consists of (i) a relational Data Base Management System (DBMS), that stores data in accordance with data warehousing paradigm; (ii) an OLAP server that implements the multidimensional model and OLAP operators on top of the DBMS; (iii) an OLAP client, that combines and synchronizes tabular and graphical displays and allows query building; (iv) an ETL tool that extracts data from heterogeneous sources, transforms them and loads them into a data warehouse.

In this paper, we are focused on design of OLAP schema, which is define by Usman as a collection of database objects, including tables, views, indexes and synonyms (Usman et al., 2010).

Several research works suggest modeling for OLAP schema, that either rely on existing models (Entity/Relationship, Object-Oriented, ...) or suggest new models (Lehner, 1998; Nguyen and Tjoa, 2000; Pedersen and Jensen, 1998; Tsois et al., 2001). Regardless of the methods chosen by the authors to define the rules of their models, these models are based on three concept of multidimensional modeling : *measures*, *dimensions* and *hierarchies* (Jerbi et al., 2009).

*Corresponding author. *Email address* : l.sautot@agrosupdijon.fr

Measures are defined as dynamical and dependent variables (Nguyen and Tjoa, 2000). They quantify the objects covered by the analysis, called "facts". A fact describes often an event (for example, the sales) that occurs within an organization which uses the decision making system. The organization wishes explain the fact (Wehrle et al., 2005).

Dimensions are defined as static and independent variables (Nguyen and Tjoa, 2000), that tally with analysis axes. A dimension guides the queries, which provides several views on data (Wehrle et al., 2005).

The dimensions of an OLAP schema can contain one or more hierarchies in data. Hierarchies provide a structure to the dimensions: the data of a dimension can be categorized according to various characteristics. Users of OLAP system are usually interested in aggregated data (for example, the average of the sales for some geographical areas). Thus hierarchies are aggregation levels of data (Mahboubi et al., 2012; Markl et al., 1999; Sarawagi et al., 1998). Each level of a hierarchy contains descriptors, named "attributes" (Romero and Abello, 2010). These attributes describe each member of each level.

To design an OLAP cube, we have to determine:

- What are the measures? *i.e.* what is the phenomenon we want to study and how to measure it? With a measure, we have to determine an aggregation function: do we use sum, average or count to join two values?
- What are the dimensions? *i.e.* what are the ways of our analysis? What are the parameters we want consider explaining measure variations? For each dimension, we have to determine hierarchies, *i.e.* data organization into the dimension, and attributes for the dimension members.

OLAP technology interests more and more fields and especially biology. An OLAP cube provides a very easy navigation into a data set, the possibility to build cross tabulation to analyze the data and the possibility to monitor a complex phenomenon, such as pollution of a bay (Mahboubi et al., 2013) or growth of a forest (Miquel et al., 2002). But biologists generally do not have skills to build and manage an OLAP system.

Thereby this needful high level of skills is an obstacle to democratizing of OLAP systems. Our objective in this article is to suggest an OLAP system that will be able to organize automatically hierarchies in a dimension. With this kind of system, OLAP design can be an automatic task and ultimately does not require specific IT skills.

To begin, we identified the type of automatic or semi-automatic approach, which are used to realize the design of a data warehouse or OLAP cube. Three types of approaches can be used to make the design of an data warehouse (Cravero and Sepúlveda, 2014; Tebourski et al., 2013): (i) Methods based on user specifications, or demand-driven approach; (ii) Methods based on available data, or data-driven approach; (iii) Mixed methods, or hybrid approach.

For example, oriented to demand-driven methods, we cite the work of Jovanovic *et al.*, who developed a methodology for designing a data warehouse (Jovanovic et al., 2014). This method is iterative: at each step, the system searches in the data that best correspond with information required by the user in terms of dimensions or facts. Data are modeled with an ontology.

Moreover, several other have proposed systems based on hybrid approach:

- Romero and Abello offer a hybrid methdology to build multidimensional schema from a relational database (Romero and Abello, 2010).
- Abdelhedi *et al.* have developed a prototype called CASE to build an OLAP cube with a hybrid method (Abdelhedi et al., 2011). The design is driven by both the data sources and the user specifications.
- As in many current works, Thenmozhi and Vivekanandan propose an automatic system to build the schema of a data warehouse from an ontology (Thenmozhi and Vivekanandan, 2013).

Finally, the following authors have worked on automatic data-driven systems and using data mining to build a data warehouse or an OLAP cube:

1. Eder *et al.* apply data mining algorithms such as auto-regression, auto-correlation, regression or fast Fourier transform on the data in a data warehouse (Eder et al., 2003). Their goal is to automatically detect the structural changes in a data warehouse, such as deleting, adding, merging member in a hierarchy.
2. Usman (Usman et al., 2010; Usman and Pears, 2010) provides a methodology to design automatically OLAP schema and data warehouses with hierarchical clustering. This author suggests a complete system to build OLAP systems with data sets. The system, which is proposed by Usman *et al.*, uses hierarchical agglomerative clustering to perform a pre-processing on the data. After that, the system identifies facts and dimensions into the clustered data. This system is able to build star schema, snowflake schema and constellation schema.
3. Rehman *et al.* propose a system to dynamically build hierarchies based on data from Twitter (Rehman et al., 2012). This paper has two Interests: a) The cube is built on original data, that are messages of users on a social network. b) Data mining is used to dynamically build hierarchies: thanks to data mining, the categories of network users described in hierarchies are updated automatically.

Moreover, the following authors use clustering algorithms to dynamically build or modify hierarchies in an OLAP cube:

1. Messaoud *et al.* propose a new OLAP operator named OPAC which allows to aggregate facts that refer to complex objects, such as images (Messaoud et al., 2004). This operator is based on hierarchical clustering algorithm. The prototype proposed by these authors incorporates a module to evaluate the quality of aggregations.
2. Favre, Bentayeb and Boussaid (Favre et al., 2006) suggest considering rules defined by the users during browsing in an OLAP system. These rules were used to change dynamically the data warehouse schema. The system, that Favre *et al.* have proposed, has a stable part and a dynamic part. The stable part of the system corresponds of a basic OLAP schema with a star schema. From this basis, each user can define rules to build hierarchies in each dimension. These hierarchies, which depend of the user rules, constitute the dynamic part of the system.
3. In 2008, Bentayeb offers create new levels in a hierarchy with the K-means algorithm (Bentayeb, 2008). Thereafter, Bentayeb and Khemiri propose in 2013 (Bentayeb and Khemiri, 2013) an operator, called ProCK, which, as in the work of Hubert and Teste (Hubert and Teste, 2009), permits to the user to dynamically change the hierarchies during the navigation. This operator uses a K-means algorithm modified to take into account the constraints defined by the user. This operator allows to define new levels in a hierarchy.
4. Teste and Hubert propose in 2009 a new operator that allows the user to dynamically change the hierarchies within the cube OLAP during navigation (Hubert and Teste, 2009).
5. Leonhardi *et al.* offer the user to create new dimension during navigation (Leonhardi et al., 2010). These authors propose to increase the OLAP cube exploration functionalities by providing the user data mining algorithms applying on data, which are selected in the warehouse.

On the other hand, Ceci *et al.* use a hierarchical clustering to integrate continuous variables as dimensions in an OLAP schema (Ceci et al., 2011). Their tool uses a modified BIRCH algorithm. It discretizes a continuous dimension in order that the user can perform operations on conventional querying a cube: Roll-up and Drill-down. These authors use data mining to incorporate in a cube OLAP new data, whose the type lends itself poorly.

These works present several interesting aspects. First these works suggest the use of an *a posteriori* modeling of OLAP schema, perform by user or by an algorithm. Furthermore these works offer to the user the possibility to build his own OLAP schema or to build an OLAP schema according to the own structure of data. This article is inspired by these viewpoints, and we build a system that offers to user the possibility to build his own OLAP schema with a data mining method.

In a biological study, measures and dimensions are clearly identified. But the data which describe a dimension do not necessarily have an apparent hierarchical structure:

- The dimension can contain several quantitative variables and not only categories.
- The variables are heterogeneous: the data set can contain quantitative variables, nominal variables and binary variables.
- The data set can contain blank values.

The presented previous works offer to build automatically OLAP systems with hierarchical use data set with binary and quantitative variables. We suggest to supplement these works with a similarity index comes from ecological analysis, the Gower index.

In this article we provide a methodology to build automatically a hierarchy with a biological data set that contains heterogeneous variables. Our approach is as follows:

- In the first part, we introduce foremost the data set that we use and the features of this data set.
- In a second part, we present several *a priori* OLAP schemas and their limitations.
- In a third part, we explain first how our system works. We present the hierarchical agglomerative clustering and we define what clustering parameters we need to perform the hierarchical agglomerative clustering with our data set. Next we explain what the Gower index is and what their interests are.
- In a fourth part, we suggest an evaluation of the needful memory and the needful calculation time according to the number of processed data.
- Finally we conclude on the system working and performances and we present our future work.

1. A data set from a large ecological study

Our data set comes from a census program for nesting birds along the Loire River (France) (Frochot et al., 2003). The STORI (*Suivi Temporel des Oiseaux nicheurs en Rivière*: Temporal Monitoring of Nesting Birds in River Valley) is a wide research program, which studies bird populations along the rivers. The objective of this program is the observation of temporal and spatial changes into bird populations. One hundred ninety eight points were defined along the river in the framework of this program. At each point the birds are identified with the IPA (*Indice Ponctuel d'Abondance*: Punctual Abundance Index) method (Blondel et al., 1981) during four census campaigns (1990, 1996, 2002 and 2011). Bird abundances were described by a semi-quantitative abundance index.

One of the main objectives of the STORI is studying global and local factors that explain these changes. In this context, the evolution of environments along the Loire River between 1990 and 2011 were described at each point in parallel with the IPA data, to find correlations between these populations and this environment.

In fact, the data set can be summarized by:

- A measure: bird abundances that can agglomerate with a sum or an average.
- Three dimensions to analyze the abundance: species, time and space.

In this context, we build an OLAP system to manage and store these data. The working of our system was described in another section (section 3). We build a data warehouse with a star schema and an OLAP schema with three dimensions. But the spatial dimension of the OLAP schema raises problems that were explained below.

To explain bird abundances we try to establish correlations between birds and landscapes. At each point, the river and the valley are described for several years. In fact many variables are defined only for one campaign. Moreover all kinds of variables are present: there are continuous variables, discrete variables, nominal variables and ordinal variables. The variables that describe landscapes are presented in the table below.

Variable types		1990	1996	2002	2011
Quantitative	Continuous	8	0	97	44
	Discrete	7	7	7	10
Qualitative	Ordinal	5	0	0	1
	Nominal	7	2	4	6
	Binary	5	0	0	3

Table 1: Number of variables used for landscape and river description according to the year.

This dimension has three interesting features:

- There is no intrinsic hierarchy into the description of environment along the river: except keys and station identifiers, only two station attributes (on 110) are linked by a functional dependency.
- Their attributes are heterogeneous.
- Their attributes are not defined for all campaigns.

As a consequence we suggest building automatically a hierarchy for this dimension because there is no explicit hierarchy in this dimension and we want offer to biologists the possibility of building their own OLAP schema.

In this article, we focus on this spatial dimension and our objective is generalizing the results that we obtain with these data.

2. *A priori* OLAP schema design: what are the limitations?

In the precedent section, we have presented the data set that we use in this study. The ideal OLAP schema to analyze these data is a three-dimensional schema with the abundance measurements as facts, a dimension that describes the species, a dimension that records the year of bird census and a dimension that describes the census stations (Figure 1). With this structure we can perform the analysis that is interesting in this ecological study: ecology scientists want characterize spatio-temporal changes into bird populations along the Loire River.

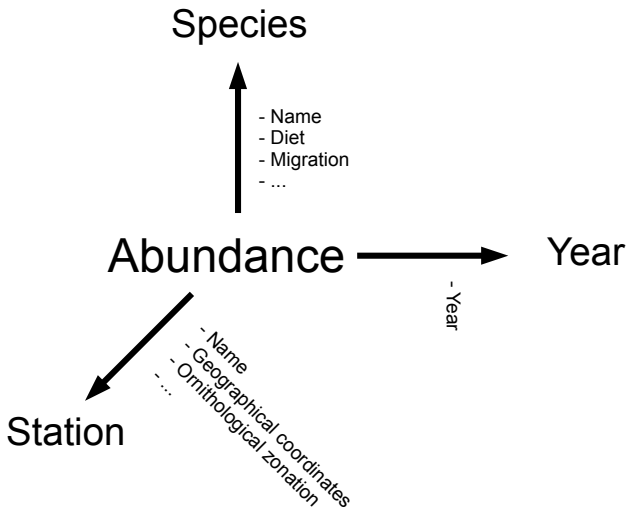

Figure 1: The dimensions of our analysis

But we have described some features of the data set which ban a simple three-dimensional schema. The spatial dimension, that describes the environment along the Loire River, is strongly correlated to the time dimension. The description of the environment is time dependent because:

- The values of some attributes, that describe the stations, change according to the time.
- Many attributes are not measured for all years.

Several models of data warehouse may be proposed to consider this correlation between spatial dimension and time dimension. The following solutions are presented at the conceptual level, according to MultiDimER notations (Malinowski and Zimanyi, 2006). Details of these notations are summarized in Appendix.

The first solution is a fact constellation schema (Figure 2). With this solution, there are two fact tables: a fact table for abundances according to species, stations and years and a fact table for environment descriptions according to stations and years. This solution is the more elegant solution. With this solution, the data storage is optimized. But the crossing between abundance data and environment data requires querying two independent cubes. Moreover qualitative variables cannot be stored in a fact table.

The second solution is a star schema (Figure 3). With this solution, there are a fact table for abundances according to species, time and stations. But the data, that describe the spatial dimension, are related to time. Thus each station is duplicated for each census campaign. Thereby the station n°1 in 1990 and the same station n°1 in 1996 are not considered as the same object in the OLAP cube. With this solution, the spatial consistency of the dataset is lost.

The third solution is a fact constellation schema (Figure 4). This kind of solution has been proposed by Miquel *et al.* in 2002 (Miquel et al., 2002). With this solution, we build a fact table for each census campaign. Each yearly fact table is linked to the "species" dimension and to a yearly "stations" dimension. The main disadvantage of this solution is the loss of the temporal consistency of the data set.

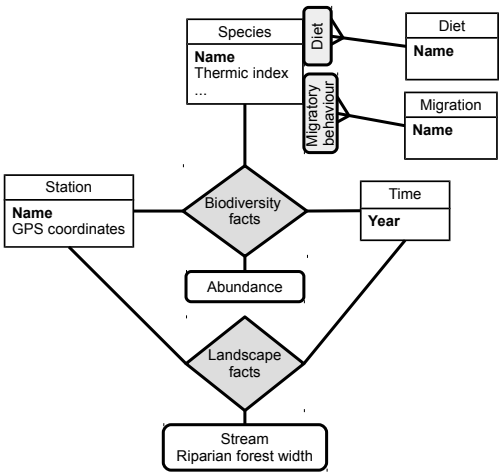

Figure 2: A fact constellation schema with a fact table for abundances and a fact table for environment description

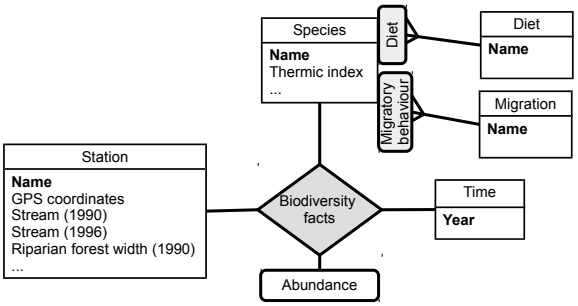

Figure 3: A star schema with a time-dependent spatial dimension

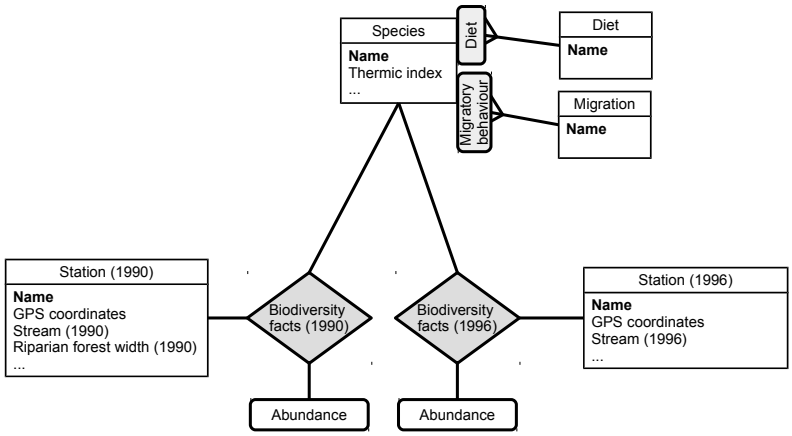

Figure 4: A fact constellation schema with a fact table for each census year

Finally, none of these three solutions can provide a perfect schema (Table 2). Thus we suggest in this article a solution to build a single spatial dimension. Thereby we obtain the three-dimensional cube that is shown on Figure1. To propose a spatial dimension, with a coherent hierarchy, we use a clustering method. This kind of method can detect a structure in a dataset. With a clustering method we can propose a prototype that builds automatically a dimension for an OLAP cube.

	Solution 1	Solution 2	Solution 3
Solution description	Fact constellation schema with a fact table for abundances and a fact table for environment descriptions	Star schema with a time-dependent spatial dimension	Fact constellation schema with a fact table with abundances for each census year
Limitations of the solution	Crossing between abundance data and environment data requires querying two cubes. Qualitative environmental variables cannot be stored.	Spatial consistency of the dataset is lost.	Temporal consistency is lost.

Table 2: Summary of the limitations of each solution

3. Proposition: an automatic hierarchy design for OLAP schema based on clustering method

To ease understanding of sections 3 and 4, we offer to clarify some vocabulary. In a clustering context, "individuals" are items, which will be classified. Moreover "variables" are descriptors of individuals. Variables are used

to perform the clustering algorithm, and to measure a distance between individuals. In this article, the clustering algortihm is perform in an OLAP context and is used to build a hierarchy. Thus, in the sections 3 and 4, "individual" is a synonym of "dimension member" and "variable" is a synonym of "attributes".

3.1. Prototype working

3.1.1. General working of the prototype

We build a prototype which is able to extract the relevant data from a data warehouse and to design and publish a new hierarchy in a dimension. We suggest a system which performs a hierarchical clustering on a table in a database. This system deduces the organization of the hierarchy from the clustering process. Next it updates the OLAP schema, the dimension table in the data warehouse and the OLAP cube in XML.

The working of this system has several steps (the number of steps tallies with the number on the Figure 5):

1. The system recovers data and meta data from the database. The data that the system uses are: data that describe the dimension, data type (text or numeric) of each variable in the dimension and relationship between facts and processed dimension.
2. The system identifies the type of each variable. This identification is compulsory because the calculation of a hierarchical agglomerative clustering needs knowledges about type of each variable. The identification of a variable type can be performed by the user. In this case the variables types can be asked to the user or recorded as metadata in the data warehouse. Otherwise it is possible to determine automatically the type of a variable according to the type of data (text or numeric) and the number of values. This second point was explained in the subsection 3.1.4.
3. The system performs the hierarchical agglomerative clustering with the Gower index (See subsection 3.1.2 and subsection 3.1.3).
4. According to the result of hierarchical clustering, the system creates a table in the data warehouse. The first column identifies the points and each other column is a level in the hierarchical clustering. In fact, the first column is the lower level of the hierarchy and a primary key. The values of this first column were used as foreign keys in the fact table. This step updates the OLAP schema. In our case each row is a census point along the river (section 1).
5. According to the result of hierarchical clustering, the system updates the XML file that describes the OLAP cube with the new hierarchy. This new hierarchy is the calculated hierarchy. The XML file specifies the data organization in the cube and the metadata. After the creation of the cube, this cube is published on the OLAP server.
6. After the creation of the new hierarchy in the data warehouse and after the publishing of the new cube, the users of the OLAP system can use the new cube thanks to the dedicated interface.

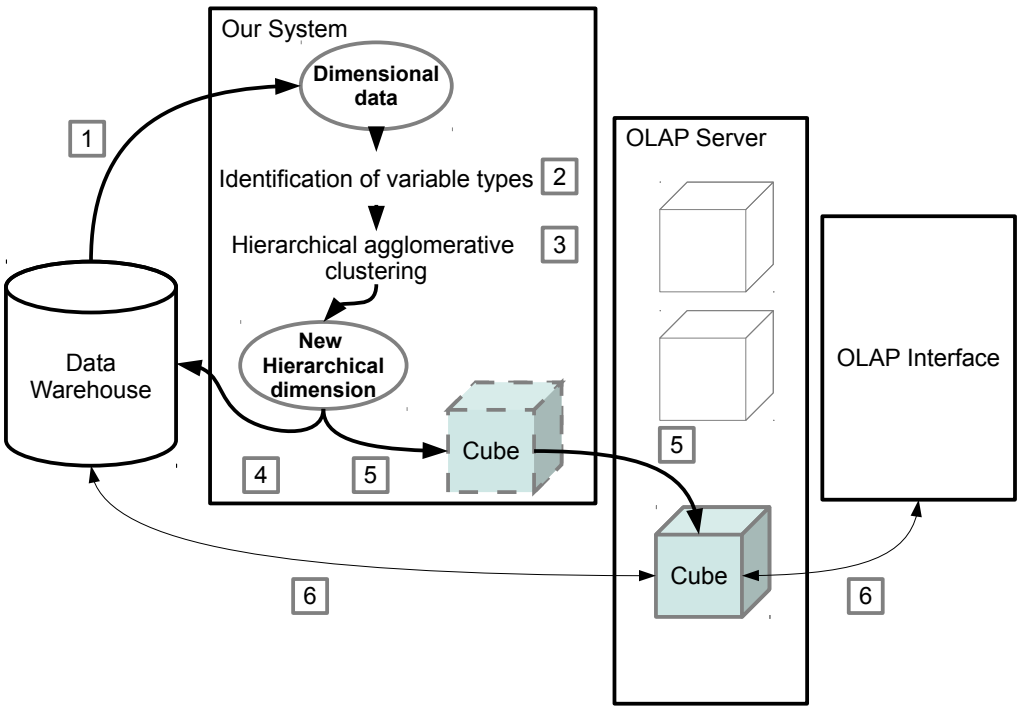

Figure 5: The working of our prototype

3.1.2. Focus on clustering method: the hierarchical agglomerative clustering

During designing an OLAP schema, hierarchies are classically built by hand. For an automatic system, we need use an algorithm to build hierarchies. We suggest using hierarchical agglomerative clustering. Hierarchical clustering has been used in OLAP systems to improve performances of queries (Markl et al., 1999) or to design OLAP schema (Usman et al., 2010).

The hierarchical agglomerative clustering is a clustering method. This method is an unsupervised method (*i.e.* no learning is needful). The aim of this method is the building of a hierarchy for find groups into the data. In a hierarchical agglomerative clustering, each branch of the built hierarchy is a cluster. This method has several steps (Tuffery, 2011):

1. Calculation of distances between individuals.
2. Choice of the two nearest individuals.
3. Aggregation of the two nearest individuals in a cluster. The cluster is now considered an individual.
4. Go back to the step 1 and loop while there is more than one individual.

The results of a hierarchical agglomerative clustering can be showed as a tree which represents the distance between the individuals (Jain et al., 1999).

To perform a hierarchical agglomerative clustering, we have to define:

- A metric to measure the distance between individuals.
- A method to aggregate individuals in cluster.

The problem with our data set is qualitative variables. With qualitative variables we cannot define a cluster like the centroid of these members. To measure the distance between two clusters, we calculate the average of all distances between all individuals in each cluster. We use unweighted average linkage. Several linkage methods can be used: unweighted average distance (UPGMA), furthest distance, shortest distance and weighted average distance (WPGMA). We use UPGMA, because, with no knowledge on the data structure, this linkage appears like the best summary of the distance between two clusters (Kojadinovic, 2004).

The distance between two individuals must mix quantitative and qualitative variables. The traditional metrics like Manhattan distance, Euclidian distance or Minkowski distance are not relevant in the case of a mixed data set.

Thereby we suggest measuring the distances between individuals with an similarity index that comes from biology: the Gower similarity index (subsection 3.1.3).

3.1.3. Focus on distance measurement: the Gower index

The Gower index is designed to measure similarity between two individuals that are defined by heterogeneous variables (Gower, 1971). The Gower index is a classical similarity index, which is often used in an ecological study or in a modeling work (Segurado and Araujo, 2004; Westphal et al., 2007). The Gower index is calculated as follow:

- $I_1$and I_2 are two individuals.
- N is the number of variables used to define the individuals.
- w_i is a weight. If the variable n°i is not define for I_1 or I_2, then $w_i = 0$. Else $w_i = 1$.
- $S_i(I_1, I_2)$ depends of the type of the variable n°i called V_i :
  - If variable n°i is qualitative then:
    - If $V_i(I_1) = V_i(I_2)$ then $S_i(I_1, I_2) = 1$,
    - Else $S_i(I_1, I_2) = 0$
  - If variable n°i is quantitative then: $S_i(I_1, I_2) = 1 - \frac{|V_i(I_1) - V_i(I_2)|}{Max(V_i) - Min(V_i)}$

in the following equation

$$S_G(I_1, I_2) = \frac{\sum_{i=1}^{N}[w_i S_i(I_1, I_2)]}{\sum_{i=1}^{N}[w_i]}$$

Some features of the Gower index can be detailed. First, the Gower index is a similarity index. Thus if a Gower index value between two individuals is close to 1, it means that the two individuals are very similar.

Secondly we explain the building of the Gower index. The calculation of Gower index corresponds to a weighted average. In fact, we calculate a similarity value between two individuals for each variable. The Gower index is the weighted average of these similarities according to variables. The Gower index distinguishes qualitative variables and quantitative variables. On the one hand this similarity index treats a qualitative variable with a boolean. If the individuals are in the same class, the boolean is equal to 1. Else the boolean is equal to 0. On the other hand this similarity index treats the quantitative variables as follow: we calculate a distance between two individuals with the absolute value of the difference. This absolute difference is divided by the range (the difference between maximum and minimum) of the variable. With this division, the difference between two individuals according to a variable is independent of the range of the variable. Finally, the fraction is subtracted to 1. Thereby we obtain the similarity between two individuals according to one variable.

Now we can calculate the similarity between two individuals according to each variable. But we need define weights for each variable. The weights permit to manage the missing values. When we calculate the Gower index between two individuals, sometimes a variable is undefined for an individual. In this calculation, the undefined variable is weighted to 0: this variable is excluded of the Gower index calculation. Thereby, we manage missing values with variable weights. Moreover, with the weights, we can manage the importance of each variable. If the user want give more importance to a variable, he can fix accordingly the weight of each variable.

We propose to calculate the Gower index for an example (Table 3, Table 4 and Table 5).

The following table is the description of the variables that we use in this example:

Variable name	Variable type	Minimum value	Maximum value
Altitude	Quantitative	0	1410
Confluence	Qualitative	-	-
Bank	Qualitative	-	-
Current	Qualitative	-	-
Substratum	Qualitative	-	-
Aquatic vegetation	Qualitative	-	-
Salinity	Quantitative	0	35
Slope	Quantitative	0	120
Valley width	Quantitative	0	2950

Table 3: Variables used for the example

The following table is the description of two stations, which are described with the previous variables:

Variable name	Station n°1	Station n°11
Altitude	1410	899
Confluence	No	No
Bank	0	1-15
Current	<10	10-25
Substratum	mud and silt	blocks
Aquatic vegetation	0	1-15
Salinity	0	0
Slope	120	3.6
Valley width	0.2	11

Table 4: Individuals used for the example

The following table shows the members of the formula for calculation of the similarity index:

Variable name	w_i	S_i
Altitude	1	0.64
Confluence	1	1
Bank	1	0
Current	1	0
Substratum	1	0
Aquatic vegetation	1	0
Salinity	1	1
Slope	1	0.03
Valley width	1	0.99
Sum	*9*	*3.66*

The following formula is the calculation of the similarity between station n°1 and station n° 11:

$$S_G = \frac{\sum w_i S_i}{\sum w_i} = \frac{3.66}{9} \simeq 0.41$$

Table 5: Calculation of Gower index of similarity between two stations

3.1.4. Focus on the determination of a variable type

In our system, the user tells if the variable is quantitative or qualitative. But if the number of variable is very important or if the information is missing, we can imagine that the system find the type of variable itself. Type of a variable depends of type of data (text or number) and the number of appearance of each values (Table 6). Two cases are very easy to solve:

1. If data are numbers and if the number of values is approximately equal to the number of individuals, then the variable is quantitative.
2. If data are texts and if the number of values is very smaller than the number of individuals, then the variable is qualitative.

Two cases are more problematic:

1. If data are texts and if the number of values is approximately equal to the number of individuals. In this case, the question is: does the comparison between two character strings make sense? If the comparison between two character sequences makes sense, this comparison is possible and a similarity between two value can be calculated. Else the variable is probably a primary key, a unique name for each individual. If this variable is a primary key, it does not provide benefit for the clustering process. Thereby this type of variables will be excluded to the clustering process.
2. If data are numbers and if the number of values is smaller than the number of individuals, then the variable can be a qualitative variable recorded with numbers or a discrete quantitative variable.

In these two problematic cases, the system can asks the user what the type of the variable is.

		Number of values	
		Number of values ≈ Number of individuals	Number of values << Number of individuals
Data type	Text	Primary key	Qualitative
	Number	Quantitative	?

Table 6: How to determine the type of a variable?

The problem is: what is the limit of the number of values for a qualitative variable encoded with numeric data? To solve this problem we use several data sets to build a decision tree. Thus, to find the threshold for our data set, we have to consider a learning variable set, which has the same characteristics as our variable set.

Therefore, we have built a data set that contains qualitative and quantitative variables. This dataset should contain 198 individuals (as our data set). We have built this dataset with external datasets, which come from the UCI Machine Learning Repository (Bache and Lichman, 2013).We choose multivariate datasets *i.e.* datasets which contains qualitative and quantitative variables. These datasets contain data about:

- Physical measurements of Abalone[1]
- Census income [2]
- Steel annealing data[3]
- Ward's Automotive Yearbook [4]
- Cylinder bands in rotogravure printing [5]
- Horse disease[6]
- Housing [7].

In our data set, we have 198 individuals. So we choose 198 individuals in each dataset from UCI Machine Learning Repository. Each item used for the learning is a variable. And, for the learning phase, we want consider variables, which are not in our environmental and ornithological data set. Thus the building of the learning variable set is very time consuming. We have limited the learning variable set so that the number of variables has an order of magnitude near of our data set. With 129 variables, we have a learning variable set quite similar to our data.

We make a decision tree with 129 variables from the external datasets (Rokach et al., 2008). A decision tree is a classification method, which has the advantage of providing automatically explicit rules. The rules of our decision tree are presented on the Figure 6.

[1]Warnick J. Nash and Tracy L. Sellers and Simon R. Talbot and Andrew J. Cawthorn and Wes B. Ford, "The Population Biology of Abalone (Haliotis species) in Tasmania - Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait.", Marine Resources Division, Marine Research Laboratories - Taroona, Departement of Primary Industry and Fisheries - Tasmania (1994).

[2]Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (1996).

[3]No reference is associated to this dataset.

[4]D. Kibler and D.W. Aha and M. Albert, "Instance-based prediction of real-valued attributes", Computational Intelligence 5 (1989), pp. 51-57.

[5]B. Evans and D. Fisher, "Overcoming process delays with decision tree induction", IEEE Expert 9, 1 (1994), pp. 60-66.

[6]No reference is associated to this dataset.

[7]D. Harrison and D.L. Rubinfeld, "Hedonic prices and the demand for clean air", J. Environ. Economics & Management 5 (1978), pp. 81-102.

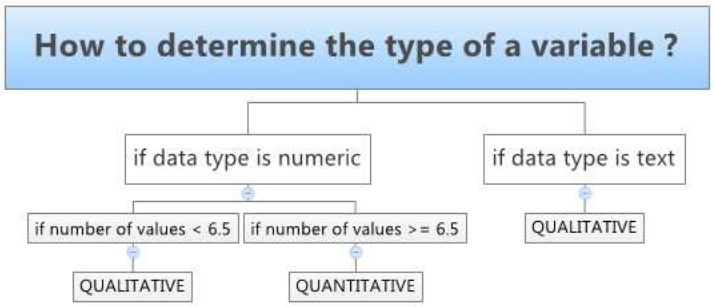

Figure 6: Decision tree to decide if a variable is quantitative or qualitative

If we apply this decision tree (Figure 6) to our data set, 10 variables on 110 are badly classified. These ten variables are quantitative variables with a very small number of values, and with the decision tree we consider that these ten variables are qualitative. This kind of error (a quantitative variable considered like a qualitative variable) is not a serious problem because in this situation, similar values are well processed and the algorithm neglects the similarity between two near values. On the other hand, a qualitative variable considered like a quantitative variable is a serious problem because the calculations performed by the algorithm have no meaning.

In conclusion we can determine automatically if a variable is qualitative or quantitative with metadata like data type and number of values. But the classification is not totally reliable. Thereby we recommend fixing a confidence interval:

- If the data type is text then the variable is qualitative.
- If the data type is numeric:
  - If the number of values is higher as 6 values then the variable is quantitative.
  - If the number of values is lower as 6 or equal to 6 values then the type of variable is problematic and the system must ask this type to the user.

3.2. Comparison between a priori *schema and calculated schema*

We detail several *a priori* OLAP schemas and their limitations in the 2. The schema that we obtain with the prototype is presented on the Figure 7. The structure of the new schema is a star schema. The structure is like of the structure, that is showed on the Figure 3. The fact table contains the bird abundances. The fact table is linked to three dimensions: the species dimension, which described the bird species, the temporal dimension and the new dimension. The new dimension is, for our example, a spatial dimension. This new dimension contains a hierarchy and this hierarchy is the result of the hierarchical agglomerative clustering. The new schema has the same structure as the natural dimensionality of the data set.

A calculated hierarchy is presented on the Figure 8.

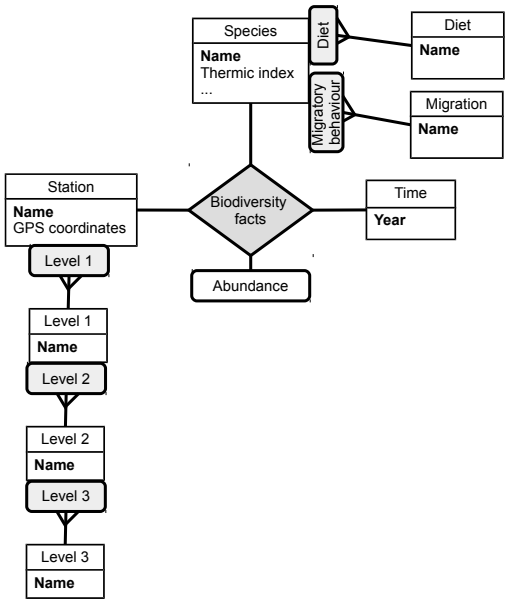

Figure 7: A star schema with the new hierarchical dimension

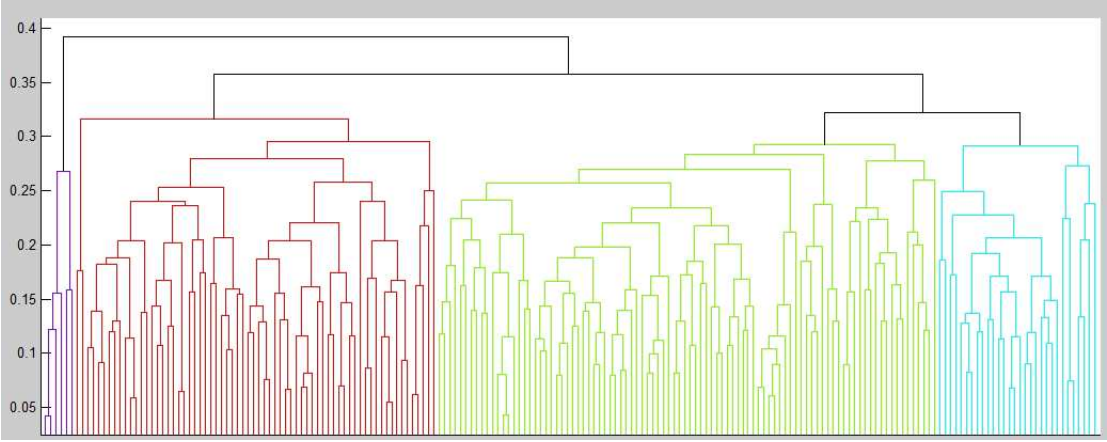

Figure 8: One hierarchy built by the system

4. System performances

In the context of this study we work with a dimension that contains approximately 200 objects (the census points along the Loire River. See section 1). But OLAP systems are designed to manage large quantities of data. Thus we suggest measuring performances of our system in order to predict calculation time and needful memory with a larger data set.

The system performances can be measured by two ways:

- The needful time for calculation of the hierarchy with Gower index.
- The number of levels of the obtained hierarchy. This number of levels tallies with the number of columns of the table which represent the new calculated hierarchy in the database. Thus the number of levels is an estimation of the needful memory to save the hierarchy.

The calculation time and the number of levels were measured according to the number of individuals and the number of variables used to build the hierarchy. The number of input data is reflected in these two parameters and we can expect that the impact of these parameters is independent to the computer configuration.

On the Figure 9 we show the number of levels according to the number of individuals and the number of levels according to the number of variables. About these graphs, we note that:

- The theoretical minimum of levels according to the number of individuals obeys to a logarithmic function (Devroye, 1986).
- The number of levels according to the number of individuals is near to this minimum: an asymptotic behavior.
- By contrast, the number of variables has no effect on the number of levels.

To model the number of level according to the number of individuals, the two best models are a power function or a logarithmic function. Despite the fact that the power function has a correlation coefficient higher ($R^2 = 0.54$) than the correlation coefficient of the logarithmic function ($R^2 = 0.47$), we believe that the logarithmic function is more relevant, because we know that the minimum follows a logarithmic function.

Moreover the best model for the number of levels according to the number of variables is a quadratic function. But the x^2 coefficient and the x coefficient are very near to 0. We can except that the number of variables has a very little impact on the number of levels. The correlation coefficient for this model is very low ($R^2 = 0.02$).

We note that the correlation coefficients are low for each estimation of number of levels.

Thus the hierarchical agglomerative clustering performed with a Gower index as distance measurement produces binary trees whose height depends of the number of individuals. The average height of these binary trees is very near the minimum height. The needful memory used to record the hierarchy is so near the minimum.

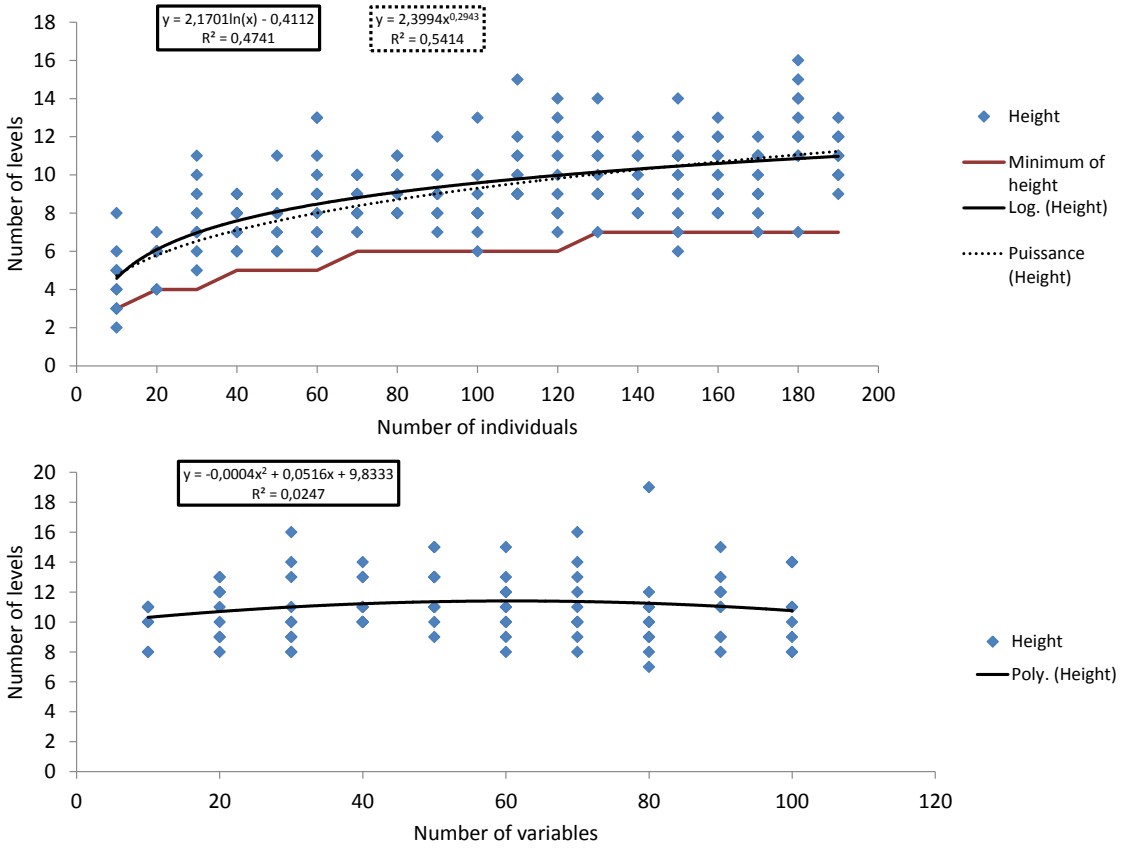

Figure 9: Height of the hierarchy according to number of individuals and according to number of variables

On the Figure 10 we show the calculation time according to the number of individuals and the number of variables. We note that:

- The calculation time according to the number of variables obeys to a linear function.
- The calculation time according to the number of individuals obeys to a quadratic function.

The complete model, which can express the calculation time according to a linear function of the number of variables and a quadratic function of the number of individuals, is:

$$t(v, M) = b_1 M^2 + b_2 M + b_3 M^2 v + b_4 M v + b_5 v + b_6$$

In this formula, t is the estimated calculation time, M is the number of individuals, v is the number of variables and b_i with i in $\{1, 2, 3, 4, 5, 6\}$ are coefficients that depend on the configuration of the computer which perform the hierarchy calculation.

We perform a stepwise linear regression to fix the coefficients. The coefficients, which can be statiscally considered equal to zero, are removed. We obtain a formula like:

$$t(v, M) = (b_1 + b_3 v) M^2 + b_2 M + b_6$$

With the computer, that we use for the performances tests, we obtain $b_1 = 1.83\text{x}10^{-3}$, $b_2 = -1.06\text{x}10^{-6}$, $b_3 = 1.51\text{x}10^{-5}$and $b_6 = 1.15$. The correlation coefficient between this model and the measured calculation time is equal to 99.7%. On the Figure 11 we show the measured calculation time and the model that we suggest above. The estimation shows well the changes of calculation time according to the number of individuals and the number of variables.

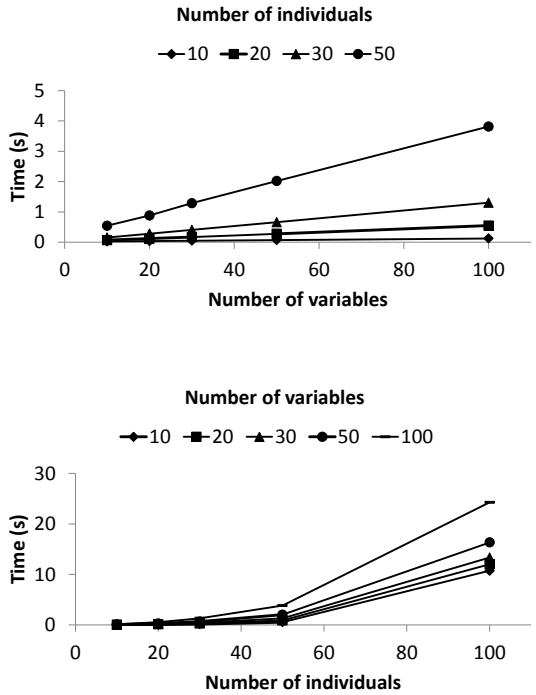

Figure 10: Calculation time according to the number of individuals and the number of variables

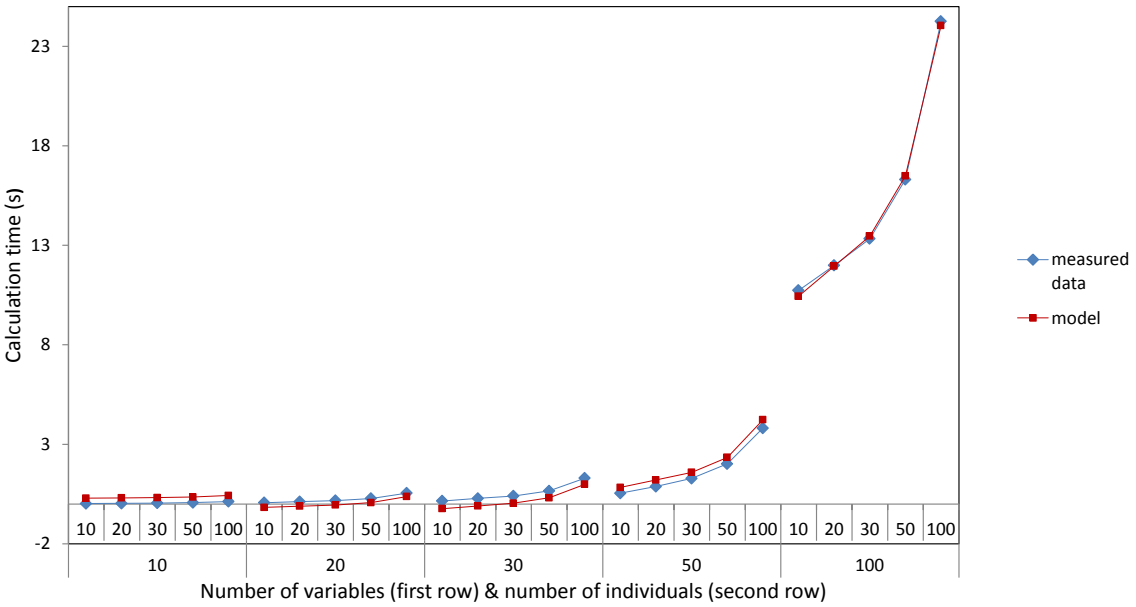

Figure 11: Calculation time according to number of variables (first row of X axis) and to number of individuals (second row of X axis) and an estimation of calculation time

These performance tests have been performed on the following configuration:

- The computer has a Intel® Core™ 2Duo processor and 4Go RAM.
- The Operating System (OS) is a Windows 7, 32-bit (© Microsoft Corporation).
- The prototype runs on the software MATLAB® 2011 (© MathWorks).

Discussion

Discussion about the system that we have proposed

In this part, we discuss about the system that is proposed and we suggest perspectives to improve the prototype. First, we discuss about the clustering method. Secondly, we discuss about the use of the Gower index. Thirdly, we discuss about a perspective of cluster characterization.

The use of hierarchical agglomerative clustering

We use a hierarchical agglomerative clustering, that provides a complete hierarchy of the data. But the prototype works perfectly with another clustering algorithm, like the K-means algorithm. Thereby our prototype can work with several clustering algorithm. It will be interesting to compare hierarchical and simple clustering algorithm. Thereby we know which type of clustering method is more efficient to build a new hierarchy in an OLAP schema.

Secondly, we use an unweighted average distance as a linkage method. But there are several linkage methods. The use of a linkage method could be chosen by the user if he has knowledge about his data set. Else, we could propose to user several hierarchies, which are obtained with several linkage methods. The user could choose his favorite hierarchy. There are two ways to show the hierarchies at the user: the system can present the result of hierarchical agglomerative clustering with different parameters or the system can provide to the user the possibility to test the new cube (Bimonte et al., 2013).

The use of the Gower index

The using of the Gower index to perform a hierarchical agglomerative clustering asks some questions.

First, to perform a hierarchical agglomerative clustering with the Gower index, we need to know what the type of each variable is. In the subsection 3.1.4, we suggest a way to determine automatically the type of a variable. But this method is not perfect and there is an error risk. In our case we obtain approximately 10% error. However we identify two types of error and with our data set we obtain the less problematic errors. Thus the type of a variable should be determined by an algorithm or directly by the user, and the database must save the metadata that indicate the type of the variable.

Secondly, we can question the calculation of the Gower index. A hierarchical agglomerative clustering with the Gower index permits building a hierarchy with a multitype data set. But this Gower index poses two problems:

- Foremost, the processing of a variable depends on the type of the variable. Thus we are not sure that all the variables have the same weight in the calculation process of the Gower index.
- Otherwise, the presence of qualitative variables bans the calculation of a centroid or an average individual. Thus the comparison between two clusters can be problematic.

Thus the Gower index permits the integration of qualitative variables in a clustering methodology. But these qualitative variables must be used cautiously.

Finally, the calculation of Gower index requires knowledge about the type of variables (qualitative or quantitative). But there is a third variable type: ordinal variables. Ordinal variables are qualitative variables but there is an order relationship between the classes of the variables. For example, an ordinal variable is a variable that can take the values {very low, low, medium, high, very high}. This variable is qualitative. But we know that the value 'very low' is closer to 'low' than 'very high'. A calculation of distance is therefore possible between two values of this variable. For the moment, the Gower index is not defined for the ordinal variables and the ordinal variables are treated as qualitative variables. It would be interesting to define the Gower index for ordinal variables. But the automatic detection of ordinal variables would be difficult.

How can the calculated clusters be characterized?

The final point of this discussion, which is focused on our prototype, is about cluster characterization. With a data mining method, we determine a hierarchy in the data. But after this calculation, the clusters should be characterized. Thereby the system could find a label for each cluster. We can expect that a statistical method could find a label for each cluster. We develop now an opinion to find label for each cluster.

We define four main clusters in our data with the hierarchy on the Figure 8. We perform statistical test to determine which variables are related to clusters. We perform Chi² test for qualitative variable and ANOVA test for quantitative variables. With these tests, we know which variables are significantly related to the clusters. On the Figure 12, the variables significantly related to the clusters have a p-value under the significance level of 5%. We can see on this figure, that the land cover of aquatic environment (MIAQ) and the land cover of urban area (URBA) are not significantly related (with a significance level of 5%) to the clusters. All other variables are significantly related to the clusters.

If we consider a significant related variable, we can characterize each cluster. For example, the maximum height of riparian forest is near to 0 m for the stations of the cluster n°1 and between 10 and 35 m for the stations of the cluster n°4 (Figure 13). According to the Figure 13, the cluster n°1 is characterized by low values of maximum height of riparian forest, the cluster n°2 and the cluster n°3 is characterized by medium values of maximum height of riparian forest and the cluster n°4 is characterized by high values of maximum height of riparian forest. On this figure, the red line represents the median.

If this kind of methodology is developed and automatized, the system could be find label for each data clusters. There is a notch around the median. If the notches of two boxplot do not overlap, we can conclude that the medians differ with 95% confidence.

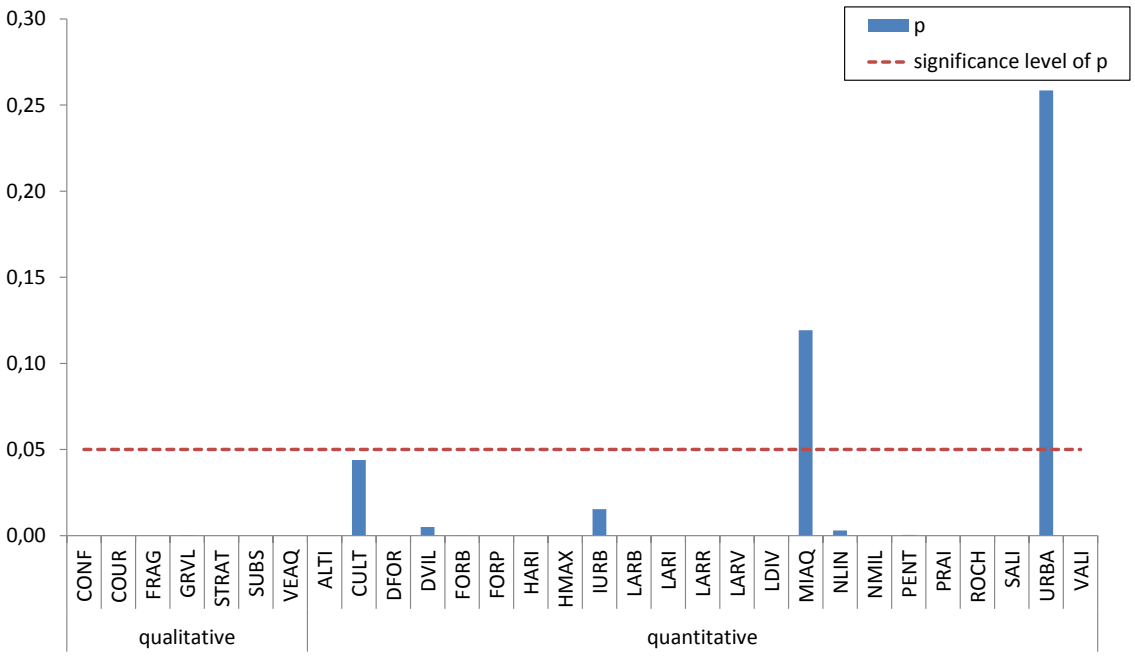

Figure 12: p-values of statistical tests for each variable, which are used to build the hierarchy

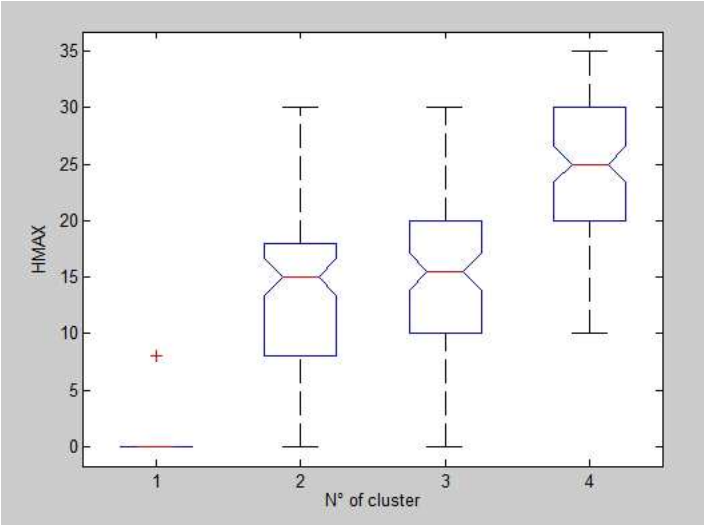

Figure 13: Values of the maximum height of riparian forest (HMAX, in meters) according to the clustering results

Discussion about the system performances

In this part, we discuss about the performances of the system that is proposed and we suggest perspectives to improve the prototype performances. In fact, we have made choices about the data mining method, which is used to calculate the new hierarchy. But these choices have a strong impact on the calculation time of a new hierarchy.

First, the hierarchical agglomerative clustering permits to obtain a complete hierarchy of the data. But we can think that the system can work with another clustering method, like the K-means clustering algorithm. A more

simple clustering method may offer better calculation performances. But we know that with an algorithm, like K-means algorithm, the calculated hierarchy will be simple, with only a level. Thus, improving performances with a simpler algorithm produces a simpler hierarchy. The question is: when the hierarchical agglomerative clustering is gainful ? *i.e.* when the hierarchical agglomerative does provide an interesting hierarchy (no more simple and no more complex), which warrants the high calculation time ?

Secondly, our clustering algorithm is not optimized. But we think that the performances of our prototype can be improved, because several steps of the calculation can be parallelized.

Thereby, the calculation time performances can be widely improved.

Conclusion

In this article, we presented a method to build automatically new hierarchies in a dimension with a clustering algorithm. The prototype that we have built is able to design and publish a new OLAP schema and a new OLAP cube from a table of a data warehouse.

Our system loads the data from a data warehouse. Next the system calculates a hierarchy with a hierarchical agglomerative clustering. But, the data sets, which are used in ecology, contain often qualitative variables and quantitative variables. Moreover a data set can contain missing values. To manage this data set and perform a hierarchical agglomerative clustering, we use a similarity index to characterize the distance between two records. This similarity index is the Gower index, an index comes from the ecology. The Gower index permits to mix qualitative and quantitative variables and so this similarity index permits the comparison between individuals that are described by heterogeneous variables. Moreover the Gower index manages missing values. To compare two individuals, this similarity index calculates a weighted average of similarities. Similarities are calculated for each variable and the formula depends on the type of variable (qualitative or quantitative). The weights concern the variables and permit to manage missing values.

Using the Gower index entails the identification of the type of variables. This identification can be entrusted to the user. But the type of a variable can be also determined by an algorithm according the data type (text or numeric) and the number of values. To automatize the decision process about the type of variable, we construct a decision tree with external data sets. The decision tree classifies the variable according to the data type (text or numeric) and the number of values. We point the threshold of the number of values: if the data type is numeric and is the number of values if lower than 6 then the variable is qualitative. Else, if the data type is numeric and is the number of values if higher than 6 then the variable is quantitative.

After the calculation of the new hierarchy, the system builds a new dimension in the data warehouse and publishes the cube on the OLAP server with a XML file.

Thus with this kind of method we can build a hierarchy based on the structure of the data, when the dimension contains heterogeneous data or when the data are not hierarchical.

We have measured the performances of our prototype. We have measured the needful calculation time and the needful memory to perform a hierarchical agglomerative clustering with the Gower index. We approximate the needful memory with the height of the binary tree which is the result of a hierarchical clustering algorithm. These performance measurements show that:

- The height of the calculated tree is follows a logarithmic function according to the number of individuals and is a constant according to the number of variables.
- The calculation time follows a quadratic function according to the number of individuals and a linear function according to the number of variables.

The calculation time performances are not very satisfactory. Indeed a good performance for an algorithm is a time function under the linear function, like logarithmic function. The algorithm, that we have written to calculate hierarchy with the Gower index, has a calculation time function equal to a quadratic function according to the number of hierarchy members. But this algorithm is not optimized and we expect that some calculations can be parallelized. Thereby the calculation time performances can be improved.

In conclusion, the data mining, and in particular the clustering methods, permits to analyze the structure of the data. This structure can be used to build dimensions automatically in an OLAP cube. This type of analysis can resolve problems of OLAP cubes modeling, in particular if the data set contains missing values, or inconsistency according to space or time.

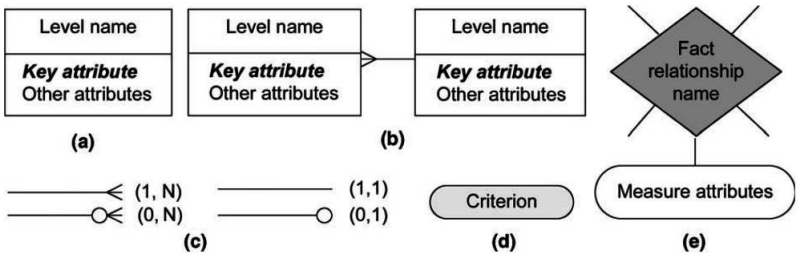

Figure 14: Notations for multidimensional model: (a) level, (b) hierarchy, (c) cardinalities, (d) analysis criterion, and (e) fact relationship.

Appendix: MultiDimER notations

As a reminder, we provide the notations defined by Malinowski and Zimanyi in (Malinowski and Zimanyi, 2006) to describe a data warehouse at the conceptual level. The following figure summarizes the notations :

References

[1] Abdelhedi, F., Pujolle, G., Teste, O., Zurfluh, G., 2011. Computer-aided data-mart design, in: 13th International Conference on Entreprise Information Systems (ICEIS 2011).

[2] Bache, K., Lichman, M., 2013. UCI machine learning repository.

[3] Bentayeb, F., 2008. K-means based approach for olap dimension updates, in: 10th International Conference on Enterprise Information Systems (ICEIS), pp. 531–534.

[4] Bentayeb, F., Khemiri, R., 2013. Adapting olap analysis to users constraints through semantic hierarchies, in: Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS 2013), pp. 160–167.

[5] Bimonte, S., Edoh-Alove, É., Nazih, H., Kang, M.A., Rizzi, S., 2013. Protolap: Rapid olap prototyping with on-demand data supply, in: Proceedings of the sixteenth international workshop on Data warehousing and OLAP, ACM. pp. 61–66.

[6] Blondel, J., Ferry, C., Frochot, B., 1981. Estimating Numbers of Terrestrial Birds. Studies in avian biology.. RALPH and SCOTT Eds.. volume 6. chapter Point counts with unlimited distance. pp. 414–420.

[7] Ceci, M., Cuzzocrea, A., Malerba, D., 2011. Olap over continuous domains via density-based hierarchical clustering, in: 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2011), pp. 559–570.

[8] Codd, E., Codd, S., Salley, C., 1993. Providing olap (on-line analytical processing) to user-analysts : An it mandate. Codd and Dat, Inc 32, 31.

[9] Cravero, A., Sepúlveda, S., 2014. Multidimensional design paradigms for data warehouses: A systematic mapping study. Journal of Software Engineering and Applications (JSEA) 7, 53–61.

[10] Devroye, L., 1986. A note on the height of binary search trees. Journal of the ACM (JACM) 33, 489–498.

[11] Eder, J., Koncilia, C., Mitsche, D., 2003. Automatic detection of structural changes in data warehouses, in: Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003), pp. 119–128.

[12] Favre, C., Bentayeb, F., Boussaid, O., 2006. A knowledge-driven data warehouse model for analysis evolution. Frontiers in Artificial Intelligence and Applications 143, 271.

[13] Frochot, B., Eybert, M., Journaux, L., Roché, J., Faivre, B., 2003. Nesting birds assemblages along the river loire: result from a 12 years-study. Alauda 71, 179–190. Tiré à part.

[14] Gower, J., 1971. A general coefficient fo similarity and some of its properties. Biometrics 27, 857–871.

[15] Hubert, G., Teste, O., 2009. Analyse multigraduelle olap, in: EGC 2009, pp. 241–252.

[16] Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. ACM Computing Survey 31, 264–322.

[17] Jerbi, H., Ravat, F., Teste, O., Zurfluh, G., 2009. Applying recommendation technology in olap systems, in: Enterprise Information Systems. Springer, pp. 220–233.

[18] Jovanovic, P., OscarRomero, AlkisSimitsis, AlbertoAbelló, Mayorova, D., 2014. A requirement-driven approach to the design and evolution of datawarehouses. Information Systems URL: http://dx.doi.org/10.1016/j.is.2014.01.004i. http://dx.doi.org/10.1016/j.is.2014.01.004i.

[19] Kojadinovic, I., 2004. Agglomerative hierarchical clustering of continuous variables based on mutual information. Computational Statistics & Data Analysis 46, 269 – 294.

[20] Lehner, W., 1998. Modeling large scale olap scenarios, in: In Advances in Database Technology - EDBT'98, volume 1377 of LNCS, Springer. pp. 153–167.

[21] Leonhardi, B., Mitschang, B., Pulido, R., Sieb, C., Wurst, M., 2010. Augmenting olap exploration with dynamic advanced analytics, in: 13th International Conference on Extending Database Technology (EDBT 2010).

[22] Mahboubi, H., Bimonte, S., Deffuant, G., Chanet, J.P., , Pinet, F., 2013. Semi-automatic design of spatial data cubes from simulation model results. International Journal of Data Warehousing and Mining 9, 70–95.

[23] Mahboubi, H., Faure, T., Bimonte, S., Deffuant, G., Chanet, J.P., , Pinet, F., 2012. New Technologies for Constructing Complex Agricultural and Environmental Systems. P. Papajorgji and F. Pinet. chapter A Multidimensional Model for Data Warehouses of Simulation Results. pp. 1–18.

[24] Malinowski, E., Zimanyi, E., 2006. Hierarchies in a multidimensional model: From conceptual modeling to logical representation. Data and Knowledge Engineering 59, 348–377.

[25] Markl, V., Ramsak, F., Bayer, R., 1999. Improving olap performance by multidimensional hierarchical clustering, in: Proc. of IDEAS 99, pp. 165–177.

[26] Messaoud, R.B., Boussaid, O., Rabaséda, S., 2004. A new olap aggregation based on the ahc technique, in: DOLAP 2004, ACM Seventh International Workshop on Data Warehousing and OLAP, pp. 65–72.

[27] Miquel, M., Bédard, Y., Brisebois, A., Pouliot, J., Marchand, P., Brodeur, J., 2002. Modeling multi-dimensional spatio-temporal data werehouses in a context of evolving specifications. International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences 34, 142–147.

[28] Nguyen, T.B., Tjoa, A.M., 2000. An object oriented multidimensional data model for olap, in: In Proc. of 1st Int. Conf. on Web-Age Information Management (WAIM), number 1846 in LNCS, Springer. pp. 69–82.

[29] Pedersen, T.B., Jensen, C.S., 1998. Multidimensional data modeling for complex data.

[30] Rehman, N.U., Mansmann, S., Weiler, A., Scholl, M.H., 2012. Discovering dynamic classification hierarchies in olap dimensions, in: ISMIS 2012 : 20th International Symposium on Methodologies for Intelligent System, pp. 425–434.

[31] Rivest, S., Bédard, Y., Proulx, M.J., Nadeau, M., Hubert, F., Pastor, J., 2005. Solap technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. ISPRS journal of photogrammetry and remote sensing 60, 17–33.

[32] Rokach, L., Maimon, O., Miamon, O.Z., 2008. Data Mining with Decision Trees : Theory and Applications. volume 69 of *Machine Percpetion and Artificial Intelligence*. World Sctientific Publishing Co.

[33] Romero, O., Abello, A., 2010. Automatic validation of requirements to support multidimensional design. Data & Knowledge Engineering 69, 917–942.

[34] Sarawagi, S., Agrawal, R., Megiddo, N., 1998. Discovery-driven exploration of olap data cubes, in: In Proc. Int. Conf. of Extending Database Technology (EDBT'98, Springer-Verlag. pp. 168–182.

[35] Segurado, P., Araujo, M.B., 2004. An evaluation of methods for modelling species distributions. Journal of Biogeography 31, 1555–1568.

[36] Tebourski, W., Karâa, W.B.A., Ghezala, H.B., 2013. Semi-automatic data warehouse design methodologies: a survey. International Journal of Computer Science Issues (IJCSI) 10, 48–54.

[37] Thenmozhi, M., Vivekanandan, K., 2013. A tool for data warehouse multidimensional schema design using ontology. International Journal of Computer Science Issues (IJCSI) 10, 161–168.

[38] Tsois, A., Karayannidis, N., Sellis, T., 2001. Mac: Conceptual data modeling for olap, in: 3rd International Workshop on Design and Management of Data Warehouses (DMDW 2001, p. 2001.

[39] Tuffery, S., 2011. Data mining and statistics for decision making. John Wiley & Sons.

[40] Usman, M., Asghar, S., Fong, S., 2010. Data mining and automatic olap schema generation, in: Digital Information Management (ICDIM), 2010 Fifth International Conference on, IEEE. pp. 35–43.

[41] Usman, M., Pears, R., 2010. A methodology for integrating and exploiting data mining techniques in the design of data warehouses, in: Advanced Information Management and Service (IMS), 2010 6th International Conference on, IEEE. pp. 361–367.

[42] Wehrle, P., Miquel, M., Tchounikine, A., 2005. A model for distributing and querying a data warehouse on a computing grid, in: Proceedings of 11th International Conference on Parallel and Distributed Systems, IEEE. pp. 203–209.

[43] Westphal, M.I., Field, S.A., Possingham, H.P., 2007. Optimizing landscape configuration : A case study of woodland birds in the mount lofty ranges, south australia. Landscape and Urban Planning 81, 56–66.