A linear programming-based framework for handling missing data in multi-granular data warehouses

Sandro Bimonte, Libo Ren, Nestor Koueya

A Linear Programming-based Framework for Handling Missing Data in Multi-granular Data Warehouses

Sandro Bimonte

TSCF, INRAE, University of Clermont, Campus Cezaux, Aubiere, France, Sandro.bimonte@inrae.fr

Libo Ren

CRCGM, University of Clermont, Campus Cezaux, Aubiere, France, Libo.REN@uca.fr

Nestor Koueya

LIMOS, University of Clermont, Campus Cezaux, Aubiere, France, nestor.koueya@isima.fr

ABSTRACT

Data Warehouse (DW) and OLAP systems are first citizens of Business Intelligence tools. They are widely used in the academic and industrial communities for numerous different fields of application. Despite the maturity of DW and [i]OLAP systems, with the advent of Big Data, more and more sources of data are available, and warehousing this data can lead to important quality issues. In this work, we focus on missing numerical and categorical in presence of aggregated facts. Motivated by the lack of a formal approach for the imputation of this kind of data taking into account all type of aggregation functions (distributive, algebraic and holistic), we propose an new methodology based on linear programming. Our methodology allows dealing with the relaxed constraints over classical SQL aggregation functions. The proposed approach is tested on two well-known datasets. Experiments show the effectiveness of the proposed approach.

1 Introduction

A Data Warehouse (DW) is a "*subject-oriented, integrated, time-variant and non-volatile collection of data to support the decision-making process*" (Inmon, 1996). Warehoused data are analyzed using OLAP systems enabling online exploration of data stored according to the *multidimensional model*. Warehoused data are represented according to analysis different axes (dimensions) and facts. Dimensions are organized in hierarchies composed of levels. Facts represent the analysis subjects, and they are described by numerical measures. Measures are aggregated along dimensions hierarchies using aggregation functions (e.g. sum, min, max, etc.). However, complex multidimensional structures and aggregation functions are usually used in real applications. Measures can be collected on aggregated hierarchy levels, which are then aggregated and stored in the DW. This issue can be due to several reasons, for example when data sources are collected at different granularities (for example, a sensor can send hourly or daily data according to its battery level), when errors appear in the Extraction, Transformation and Loading (ETL) process (for example, detailed data is deleted), etc. Therefore, multi-granular facts have been studied in several works that provide conceptual (Boulil et al., 2015), logical and physical models (Iftikhar & Pedersen, 2010) to handle them in DW and OLAP systems. Multi-granular facts are strictly related to aggregation functions. A wide literature exists about aggregation functions in DW and OLAP systems. According to Gray et al. (1997), aggregation functions can be classified into three different groups. The first group corresponds to distributive functions (e.g. sum, count, etc.), which calculate aggregated facts of the selected granularity level from the values already aggregated at the lower level (e.g. yearly amounts can be calculated by summing quarterly values). The second group corresponds to algebraic functions (for example the average) that calculate aggregated values from stored intermediate results (for example, the average of an amount per year can be calculated from the sum of measures and its count at quarter level). The third group corresponds to holistic functions that cannot be calculated from intermediate results. In this case, the measure of aggregated facts must be calculated by using the measures of facts at the lowest granularity level (e.g. distinct count, rank, etc.).

In this work, we focus on the management of missing data considering multi-granular facts and most common distributive, algebraic, and holistic SQL aggregation functions. Indeed, incomplete data is endemic to real-world data (Dyreson et al., 2003). This statement is more relevant today in the context of big data with a variety of sources and data acquisition tools. As shown by several works, this quality issue can have important drawbacks on the result of analysis based on incomplete data, since such analysis may be inaccurate, and can lead to misleading decisions (Little et al., 2002). Therefore, motivated by the importance of the management of incomplete data, several studies have been done in the context of relational and statistical databases (Dyreson, 1997). There exist different approaches for handling incomplete data. Among them, imputation, which consists in using statistical processes to estimate missing data, is one of the most widely used approach in the context of DW (Rubin, 1987; Graham, Olchowski, and Gilreath 2007). However, to the best of our knowledge, existing works in the DW context do not take into account multi-granular facts in the imputation process, and they do not support holistic aggregation functions, such as distinct count. Indeed, estimation functions can output values that do not fit with multi-granular facts. Therefore, motivated by the lack of a formal framework for imputation of measures in multi-granular DW, we propose a generic approach, called *adjustment*, which enables to use detailed and aggregated facts in the imputation process solving the limitations of existing work. Our proposal considers the most common aggregation functions supported by Relational Database Management Systems (DBMS) (such as Postgresql, Oracle, etc.) and OLAP servers (such as Mondrian, Microsoft Analysis Server, etc.): sum, count, max and min as distributive functions, average as algebraic function, and distinct count as holistic function. Our approach is implemented as a linear programming problem.

Let us note that, since the choice of the right imputation method depends on the particular used dataset, the estimation functions used by our *adjustment* approach are considered as input parameters. We classify them into two categories:

- "*Horizontal functions*" that estimate the missing facts based on the relations of similarities, dissimilarities, correlations, etc. of the facts with the same dimensions levels.

- "*Vertical functions*" that use aggregated facts as the only parameter to estimate detailed missing facts.

Our adjustment approach is generic since it does not provide any suggestion for the choice of the most appropriate vertical and horizontal methods, but it focuses on coupling these methods to improve the results of horizontal functions by means of aggregated facts. In this way, it improves the imputation performance for any warehoused dataset. Therefore, the choice of particular vertical and/or horizontal imputation functions is let to decision-makers and DW experts.

Finally, its implementation and validation are also presented, by means of two commonly used datasets in DW and data mining academic research and industrial communities.

Our approach is very similar to spatial, temporal and spatio-temporal disaggregation methods that have been widely studied in literature (Sax et al., 2013), (Plumejeaud et al., 2011). However, these methods are usually very complex, and they fit with a particular application domain. In the context of DW, decision-makers and DW experts are not always able to choose and implement the best disaggregation method for two main reasons. Firstly, warehoused data coming from different external sources (meteorological, retail, etc.) necessitate advanced skills for each of them. Secondly, they usually have not research profile skills to be able to understand complex disaggregation methods proposed in literature. Therefore, there is a need for an approach being able to work also with simple estimation methods. For example, let us consider some marketing decision-makers who want to analyze the impact of the weather on the sales of a retail company using OLAP systems. Then, external weather data are integrated in the DW with internal sales data. In the case of missing values in the weather dataset, decision-makers and DW experts of the retail company must estimate them, but they have not sufficient skills to select the best estimation or disaggregation method to use (Gagnon et al., 2017). However, they can easily apply some simple estimation like average, mean, etc. which do not provide the best estimation possible, but represent a simple base to handle with missing values. Therefore, once missing values have been estimated, they can explore the warehoused data by means of OLAP operators, and keep in mind the values that have been estimated, and so pay more attention to these values in their analysis.

The rest of this paper is organized as follows. Section 2 gives the motivation of this study with an analytical case study. Section 3 details related work, Section 4 and Section 5 present the multidimensional and multi-granular model used by our approach. Section 6 describes the proposed adjustment approach, and Section 7 presents experimental results.

2 Motivation

In this section, we present the motivation of our work using a retail case study. In particular, we use the FoodMart DW. It is a dataset supplied with some existing commercial Data Warehouse and OLAP systems, such as Pentaho Mondrian, Microsoft Analysis Services, etc.

The multidimensional model of this case study is presented in Figure 1 using the UML profile for DW proposed by (Boulil et al. 2015). The FoodMart DW has been modified to fit with the research issues of this work by simplifying several dimensions, and introducing multi-granular facts.

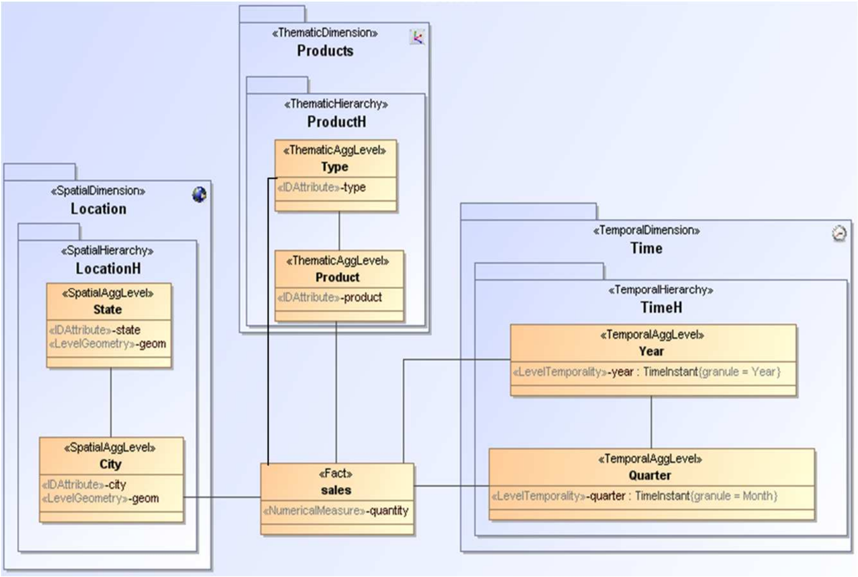


Figure 1. *Sales DW schema*

This DW consists of three dimensions: a spatial dimension (Location) representing cities stores grouped by region, a classical temporal dimension, and a thematic dimension representing products. Instances of dimensions (i.e. members) are detailed in Figure 2. The measures are the number of sales, which are summed together along all dimensions, and the distinct count of products.

This DW allows analyzing the quantities of sold of products according to the spatial and temporal dimensions. For example, the DW allows decision-makers to understand the temporal periods where TVs are the most sold products. Moreover, the distinct count of products allows analyzing the range of sold products in order to improve the sales of some particular unsold product. For example, the decision-makers can discover that TVs are not sold during a particular period of the year in the shops of NY. The key objective of the multi-granular fact is to store both detail and aggregated data at different levels of granularity (Iftikhar & Pedersen, 2010). The Time dimension allows storing factual data at level year and level quarter (i.e. multi-granular fact). In addition, the Product dimension is associated to the fact with the aggregated level. A sale can be associated to a particular product or to a product type.

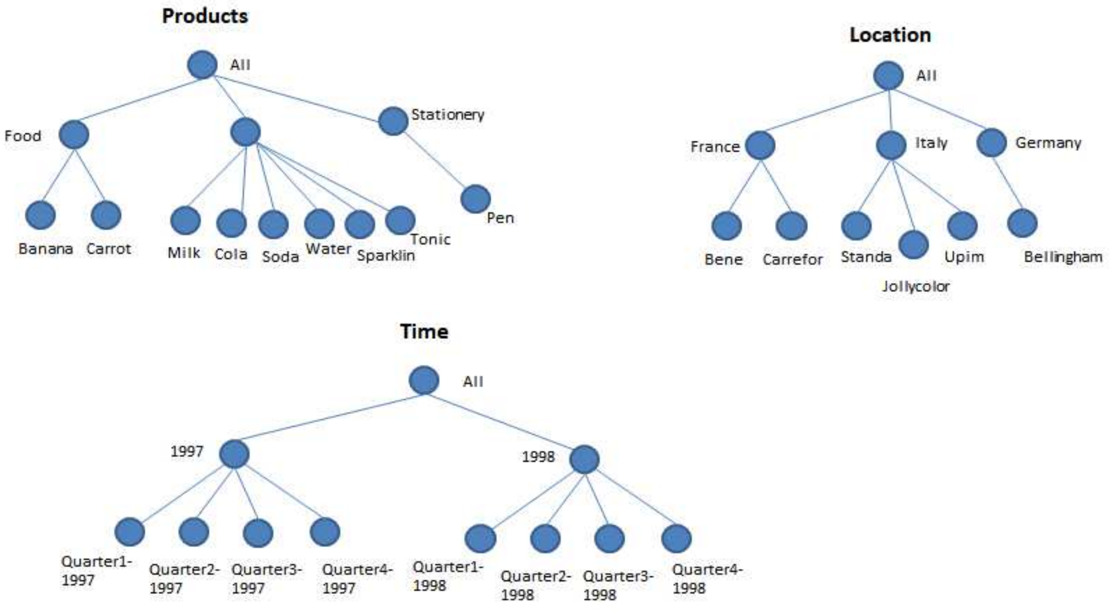


Figure 2. *Members of dimensions*

Moreover, some missing values are present in the warehoused data due to some communication problems with the data server or some errors in the ETL. Missing values can concern measures and dimensions members. Therefore, data could be stored at different granularities in the DW.

An example of factual data is shown in table 1. Measure values could be associated to the lowest dimension level (for example, row1: [Bellingham; milk; Quarter 1-1997; 20;milk]), or to coarser levels (for example, row6: [Bellingham; Drink; 1997; 90;Drink]) (i.e. multi-granular facts).

Query	Id row	Dimensions			Measure + Aggregation functions				
					Measure	Agg. functions		Measure	Agg. functions
		Location	Product	Time	Quantity SUM	Sum	Avg	Product	Distinct count
Q3	1	Bellingham	Milk	Quarter 1-1997	20			Milk	
	2	Bellingham	Milk	Quarter 2-1997	46			Milk	
	3	Bellingham	Milk	Quarter 3-1997	?			Milk	
	4	Bellingham	Milk	Quarter 4-1997	?			Cola	
	5	Bellingham	Cola	Quarter 4-1996	10			Milk	
	6	Bellingham	Milk	1997		90		Milk	
Q4	7	France	Drink	Quarter 4-1998		166		Drink	1
	8	Carrefor	?	Quarter 4-1998	30			?	
	9	Carrefor	Milk	Quarter 3-1998	15			Milk	
	10	Carrefor	Gyn	Quarter 2-1998	56			Gyn	
	11	Carrefor	?	Quarter 1-1998	65			?	
Q1	12	Standa	Bananes	Quarter 3-1998	20			Bananes	
	13	Standa	Bananes	Quarter 2-1998	10			Bananes	
	14	Standa	Bananes	Quarter 1-1998	?			Bananes	
	15	Upim	Bananes	Quarter 1-1998	30			Bananes	
Q1b	16	JollyColor	Pen	1998	10			Pen	
	17	JollyColor	Pen	Quarter 3-1998	?			Pen	
Q2	18	Bene	?	Quarter 1-1998	10			?	
	19	Bene	Sparklin	Quarter 2-1998	11			Sparklin	
	20	Bene	Tonic	Quarter 3-1998	13			Tonic	
	21	Bene	?	Quarter 4-1998	12			?	
	12	France	Drink	Quarter 1-1998	7			Drink	2
	22	Bene	Drink	1998	36			Drink	4

Table 1. *Multi-granular facts with missing values*

Missing values could be:

- measures values (for example, the sales value associated to [Bellingham; Milk; Quarter 3-1997] (row 3) is missing), and
- dimensions members. Let us consider the fact associated to [Carrefor; ?; Quarter 4-1998] (row 8). 30 products have been sold, but the sold of products in detail for the 4h quarter of 1998 is not known.

Different OLAP queries can be answered using this DW, as shown in table 2.

Id	**Query**	**Used data**	**Issue**
Q1	What is the total of banana sold in 1998 by Standa?	Rows 12-13-14	Numerical measure "quantity" is missing for row 14
Q1b	What is the quantity of pen sold in Quarter 3 of 1998 by JollyColor?	Row 17	Numerical measure "quantity" is missing for row 17
Q2	What is the product sold by Bene in Quarter1-1998?	Row 18	Alphanumeric measure "Product" (member) is missing for row 18
Q3	What is the quantity of milk sold per quarter in 1997 by *Bellingham*?	Rows 1-2-3-4	Numerical measure "quantity" is missing for rows 3-4
Q4	What is the product sold per quarter in 1998 by *Carrefor*?	Rows 8-9-10-11	Alphanumeric measure "Product" (member) is missing for row 8

Table 2.*Examples of OLAP queries*

Queries Q1, Q1b and Q3 use a numerical measure with the distributive aggregation function SUM. They involve some missing numerical measures.

Queries Q2 and Q4 use the distinct count (i.e. holistic aggregation) on the members of the Product dimension. They also involve some missing alphanumeric measures.

Let us analyze the possible solutions for the previously described queries.

For the query Q1, the measure of the row 14 is missing. It is possible to use a simple estimation function such as the nearest on the spatial dimension or on the temporal dimension. In the case of the spatial dimension, all factual data of the Upim shop (the nearest to Standa) in the first quarter of 1998 will be used. In the case of the temporal dimension, all factual data concerning the Standa store on sales of bananas in the nearest quarter of 1998 will be used. The results are shown in table 3. Let us note that, since there are not aggregated facts for missing values, then other methods using aggregated data cannot be applied. The main problem of this approach is that the result depends on the used estimation function, and on data used by the estimation function.

Let us consider the query Q1b. Only the aggregated data (annual sales) is given and the detailed data needs to be estimated (quarterly sales). In this case, the yearly measure could be divided into the quarter level using the Denton-Cholette method (Table 3).

These approaches are different according to data used for the estimation. As described in the introduction, we can consider the first one as a horizontal function, since it uses only factual data at the most detailed levels. The second one can be considered as a vertical function since it uses exclusively aggregated facts.

The query Q2 concerns a missing member, since the product name is missing for rows 18 and 21. For this kind of estimation, classification algorithms like C4.5 can be used (Table 3).

Id	Solution				Problem
	Example		Used data	Result	
Q1	Temporal interpolation function	Nearest	Row 13	10	No
	Spatial interpolation function	Nearest	Row 15	30	No
					The choice of data used for estimation can affect the result

Q1b	Multi-granular function	Denton-Cholette	Row 16	2.5	No
Q2	Classification method	C4.5	Rows 19-20	Soda	No
Q3	Temporal interpolation function	Denton-Cholette	Row 2	46 46	The aggregation with SUM of Rows 1-2-3-4 does not respect the aggregated value of row 6 20+46+46+46 ≠ 90
Q4	Classification method	C4.5	Rows 9-10-11	Milk	The aggregation with distinct count (product) of Rows 8-9-10-11 does not respect the aggregated value of row 7

Table 3. *Solutions and problems of OLAP queries of table 2*

Finally, queries Q3 and Q4 raise a particular problem related to the estimation function and the aggregated facts. Indeed, for query Q3 a numerical horizontal estimation function can be applied. In the same way, a machine learning algorithm can be used for query Q4. The problem is that, once the measures are estimated, their aggregation using the sum and the distinct count are different from aggregated facts (Table 3). The usage of vertical method does not also represent an effective solution, since it does not take into account the values of the most detailed facts.

3 Related work

Some works investigate the representation of measures at different granularities in DW. Motivated by the lack of conceptual spatio-multidimensional models based on standard languages and supporting such complex modeling requirements, and in particular multi-granular facts, (Bimonte et al., 2014) present a new UML profile for complex spatial DW. They also propose an implementation in a classical Relational OLAP architecture. In the same line, (Damiani et al., 2006) propose a formal model for spatial DW, where measures are represented at different granularities via a new OLAP operator called *measure climbing* that allows to move to a coarser granularity of the measures. (Gascueña et al., 2009) also propose a conceptual model with a multi-representation of spatial members, and a logical implementation in Relational DBMSs. (McGuire et al., 2008) define a snowflake schema for environmental applications, where three dimensions represent the same spatial members at different resolutions. (Malinowski et al., 2007) provide a complete framework for spatial

DW using complex multidimensional structures, such as complex hierarchies and multi-granular facts. They propose a conceptual model and its implementation in a relational DBMS. (Atigui et al., 2015) study the reduction of data warehouse proposing a constellation schema, where detailed measures of a DW are aggregated and stored into another DW. (Iftikhar & Pedersen, 2010) propose different logical models, extending the classical multidimensional schemata for measures at different temporal granularities. All the above-described works mainly study conceptual and logical representation and querying of DW with multi-granular facts, but they do not provide any methods to use aggregated measures for the estimation of detailed measures.

Multi-granular database's models have been also proposed, and they can be considered quite similar to DW models, since they aim to represent data at different granularities such as hierarchies' levels of DW. Bertino et al. (2010) propose a multi-granular model for spatio-temporal data. The authors define two refinement/estimating functions (*restr/split*). However, these operators do not take into account existing data at the most detailed granularity. (Hegner et al., 2017) introduce integrity constraints into multi-granular database models. The authors define a set of formal constraints for the definition of hierarchy of granularities. This work can be considered very similar to our proposal, where the coarser granularity can be considered as a constraint for the detailed granularity. However, the authors focus more on the formalization of the constraints than on their implementation. (Zhuang et al., 2016) present a multi-granular database model, and an algorithm implemented in the Hadoop framework to solve queries over large datasets using data at different granularities. However, works on multi-granular database's models are not well adapted to OLAP analysis, since they do not explicitly define the concepts of dimensions and facts.

Several works try to "estimate" missing data. The presence of missing or incomplete data is commonplace in large real-world databases (Rubin, 1976). This problem is addressed in many fields such as statistics, databases, data mining etc. and many solutions are proposed (Wohlrab and Fürnkranz, 2011; Eekhout et al., 2012) focusing on imputation (Graham, Olchowski, and Gilreath 2007) to estimate missing values. Other works study the disaggregation of time series (Sax et al., 2013), spatial and spatio-temporal data (Plumejeaud et al., 2011). Disaggregation is defined as the process to impute detailed values from coarser values respecting some mathematical constraints (sum, average, first and last values) (Sax et al., 2013). These methods use aggregated values, and some existing detailed values. A wide literature exists about disaggregation, but to the best of our knowledge, most of these works provide ad-hoc methods adapted to particular datasets, such as soil data (Odgers et al., 2014), meteorological data (Gagnon et al., 2017), census data ((Li et al., 2007) for spatial data and (Monteiro et al., 2019) for spatio-temporal data)), etc. Some generic methods have also been proposed, such as symmetric disaggregation, stochastic allocation (Gallego et al., 2001) for spatial disaggregation, Denton-Cholette (Dagum et al., 2006) and Chow-Lin (Chow and Lin, 1971) for temporal disaggregation (a survey of temporal disaggregation can be found in (Buono et al., 2018)). To the best of our knowledge, these methods are the most similar to our proposal, but present some limitations:

(i) the distinct count constraint is not taken into account,

(ii) they cannot be applied to non-spatial, non-temporal or non-spatio-temporal data,

(iii) they are conceived for particular dataset and they are usually complex, and

(iv) they must be applied to particular well identified dataset. In other terms, contrary to DW (where data is organized in dimensions, and hierarchies with members, children, descendants and ancestors), the extraction of the used dataset has to be manually done, which is difficult in a complex huge dataset.

Finally, in the context of DW, (Abdelbaki et al., 2015) and (Sair et al., 2012) integrate OLAP with data mining, to take advantage of predictive methods available. In (Wu et al., 2002), log-linear and logistic models are combined to estimate missing facts from known values. In the context of spatial DW, (Ahmed and Miquel, 2005) and (Zaamoune, 2012) use spatial and temporal interpolation functions to estimate the missing facts. These prediction methods are horizontal functions, since they do not take into account the presence of aggregated measures. (Wu et al., 2002) transform the problem of estimating detailed facts from aggregate measures into a linear system. (Palpanas et al., 2005) estimate the detailed facts from aggregate measures using the principle of maximum entropy and iterative proportional fitting algorithm. (Amanzougarene et al., 2014) try to combine the two approaches, except that their strategy is limited to a single horizontal function, in addition their method is applicable only on non-numeric measures. Table 4 presents some works related to OLAP context that involve estimation of missing detailed facts. Contrary to disaggregation works, table 4 shows that all existing works in DW and OLAP context do not take into account aggregated measures.

	Method of Estimation	**Type of Measure**	**Estimation Approach**	**Aggregation constraint**
Sarawagi et al [2]	Log linear modeling	Numeric	Horizontal	No
Wu et al. 2002	Linear system	Numeric	Vertical	Yes
Palpanas et al. 2005	Maximum Entropy Information	Numeric	Vertical	Yes
Chen et al. 2005	Linear regression	Numeric	Horizontal	No
Sair et al. 2012	Regression tree	Numeric	Horizontal	No
Amanzougarene et al, 2014	Constraint Programming and KNN	Non-numeric	Horizontal/ Vertical	Yes
Abdelbaki et al., 2015	Modular neural network	Numeric	Horizontal	No

Table 4*: Existing work on imputation of missing values in DW context*

4 Multidimensional and multi-granular model

In this section, we formalize the main concepts used by our adjustment approach. In particular, we define main concepts for multidimensional and multi-granular models.

Dimensions and levels. A dimension, noted D_i, is composed of a set of levels (DL_i). Levels are organized into hierarchies using a partial order $\leq$.

The lowest level of the dimension is called "detailed level", while other levels are noted "aggregated levels".

Example 1. Considering the dimension $Time$ of figure 1, levels are {Year, Quarter}, and there is the hierarchy $Quarter \leq Year$. $Quarter$ is the detailed level, and $Year$ is an aggregated level for the dimension $Time$.□

Member: Given a level of DL_i, the set of instances, called members are organized as a tree.

A dimension is called spatial or temporal if members represent spatial or temporal data, respectively.

Example 2. The level $Quarter$ has: {"Quarter 1-1990", "Quarter 2-1900", "Quarter 3-1990", "Quarter 4-1990"} as members (Figure 2). □

Ancestor and descendants: The ancestor of a member *m* and a level *DLi,* denoted as *ancestor(m, DLi)*, is the ancestor member of *m* that belongs to the level *DLi*.

The descendants of a member m, *descendants(m)*, are all members belonging to the most detailed level and having m as ancestor.

Example 3. ancestor(Quarter 1-1990, year) = 1990**;** descendants(1990) = {Quarter 1-1990, Quarter 2-1900, Quarter 3-1990, Quarter 4-1990} □

Fact: A fact f is composed of its coordinates and measures:

- The coordinates of f denoted as c_f consist of the members of the different dimensions. $c_f = [m_1,..,m_d], m_i \in\ dom(D_i)$, for $i \in [1,d]$
- Measures are numeric or alphanumeric values {$meas_1$….$meas_m$}

Example 4. $c_f = [Belligham, milk, Quarter\ 3 - 1997]$ is set of coordinates of the fact f and the related measure is 20 (row 1 of Table 1).□

Detailed fact: A fact f^d is a detailed fact if its coordinates are members of detailed levels.

We denote F^d the set of detailed facts.

Example 5. $c_f = [Belligham, milk, Quarter\ 3 - 1997]$ belongs to a detailed fact because "Bellingham" is a detailed member of the dimension "Location", "Milk" is a detailed member of the dimension "Product", and "Quarter 3−1997" is a detailed member of the dimension "Time" (figure 2). □

Same levels facts: Let a fact f with coordinates $c_f[m_1,..,m_d]$, *same levels facts of f over* $m_i,..,m_j$ (denoted $samelevfacts(c_f, m_1,..,m_d)$) are all facts f_i with coordinates $c_{fi}[m_1,.m'i \dots m'j, m_d]$ where m'_l is member of the same level of m_l with $l \in [i\ ,j]$.

Let us note that we do not consider spatial facts (i.e. fact with spatial measures represented as geometrical objects or numerical results of spatial operators), as proposed by (Malinowsky et al., 2007), since our linear programming-based proposal (Section 6) does not support geometrical objects.

Example 6. $samelevfacts\ ([Belligham, milk, Quarter\ 3 - 1997],\ Quarter 3 - 1997)$ ={$[Belligham, milk, Quarter\ 1 - 1997]$, $[Belligham, milk, Quarter\ 2 - 1997]$, $[Belligham, milk, Quarter\ 4 - 1997]$}.□

Ancestor fact: Let a fact *f* with coordinates $c_f[m_1,..,m_d]$, *ancestor fact of f over* m_i *to DLi (denoted* ancestorFact(cfi, DLi)*)* is the fact *fi* with coordinates $c_{fi}[m_1,.m'i,m_d]$ where m'i = ancestor(m_i, DLi) with $i \in [1\,,d]$.

Example 7. $ancestorFact\ ([Belligham, milk, Quarter\ 3 - 1997],\ Year) =$ $\{[Belligham, milk, 1997]$.□

Descendants facts: Let a fact f with coordinates $c_f[m_1,..,m_d]$, descendants *fact of f over* m_i (*denoted descendantsFacts(cfi,* m_i*))* are facts *fi* with coordinates $c_{fi}[m_1, m'i, m_d]$ where *mi* = *descendants(*m_i*)* with $i \in [1\,,d]$

Example 8. $descendantsFact\ ([Belligham, milk, 1997], Year) =$ $\{[Belligham, milk, Quarter\ 1 - 1997], [Belligham, milk, Quarter\ 2 - 1997], [Belligham, milk, Quarter\ 3 - 1997],\ \ [Belligham, milk, Quarter\ 3 - \ 1997]\}$ □

In the proposed approach, we consider a multidimensional and multi-granular model, where a fact can be represented using lowest levels, but also using aggregated levels. We formalize multi-granular facts using the following definitions.

Aggregated fact: A fact f^a is an aggregated fact on the dimension *Di*, if there is a coordinate member that belongs to an aggregated level of *Di*.

We denote *F^a* the set of aggregated facts.

Example 9. $[Belligham, milk, 1997]$ belongs to an aggregated fact because "1997" is member of the aggregated level "Year".□

Finally, we introduce missing facts that will be calculated by our approach using detailed and aggregated facts.

Missing fact: A missing fact, noted f_m, is a detailed fact with at least one unknown measure value. We denote *Fm* the set of missing facts.

Example 10. The measure related to $c_{f_m} = [Belligham, milk, Quarter\ 2 - 1997]$ is unknown then f_m is a missing fact. □

Figure 3 shows a graphical representation of facts: aggregated, detailed and missing.

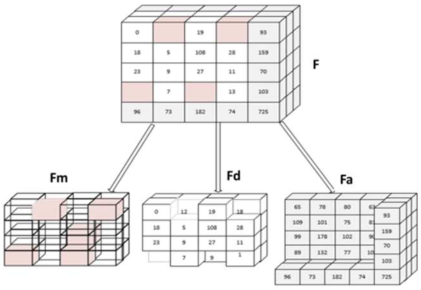


Figure 3.*Graphical representation of facts*

5 Horizontal and vertical imputation functions

In this section, we propose a formalization of horizontal and vertical categories of imputation functions using the multidimensional and multi-granular model described in Section 4. These definitions are generic since they are exclusively based on the DW elements used, and they do not refer to any particular dataset. In other terms, an imputation function is classified according to the warehoused data used for its computation (its domain). Therefore, the formalization of horizontal and vertical functions does not reflect the computational logic of the particular imputation function.

5.1 Horizontal functions

Imputation functions that belong to the horizontal category use the measures of same levels facts to estimate the missing fact.

Horizontal Estimation function: An horizontal estimation function is a function **Eh:** (F, fm) **-> Dom,** where:

- F is the set of same levels facts
- *fm* is the missing fact
- *Dom* is the domain of a measure of *fm*.

Let $meas_m$ the measure of the missing fact f_m, we define three classes of estimation functions according to the used same levels facts:

- ***Spatial horizontal functions*** use same levels facts over the spatial dimension,
- ***Temporal horizontal functions*** use same levels facts over the temporal dimension, and
- ***Spatio-temporal horizontal functions*** use same levels facts over the spatial and temporal dimensions.

Example 11. The spatial horizontal estimation function of table 3 used by the query Q1 is: ESpatialNearest(F,fm)->N.

The temporal horizontal estimation function of table 3 used by the query Q1 is: ETemporalNearest(F,fm)->N.□

5.2 Vertical functions

In this section, we formalize vertical functions used in our multi-granular and multidimensional model. Vertical functions use the measures of the ancestor to estimate the missing fact f_m.

Vertical Estimation function: Let DLi a dimension level,

A vertical estimation function is a function **Ev:** $(F, fm, DomDli)$ **-> Dom,** where:

- *F* is the set of facts
- *fm* is the missing fact
- *Dom* is the domain of $meas_m$
- *DomDli* is the domain of *DLi*
- the result is calculated using the measure of the $ancestor(f_m, DLi)$.

Example 12. The vertical estimation function temporal Denton-Cholette of table 3 used by the query Q1b is: ETDentot-Cholet(F, fm, Dom(Year))->N. □

6 Adjustment approach

As defined in Section 5, vertical functions are useful when there are not detailed facts that can be used by horizontal functions. Indeed, horizontals functions give more accurate results than vertical functions, since they use more values in the estimation process, contrary to the vertical functions that use only one aggregated value. However, as described in section 2, horizontal functions do not take into account aggregated values.

Therefore, in this section based on formalization of the multidimensional and multi-granular model, and vertical and horizontal functions proposed in sections 4 and 5 respectively, we present our adjustment approach that overcomes the above-described limits. For the sake of readability, we reconsider the example of table 1 with only data used in the case of the adjustment (queries Q3 and Q4). Then, we propose to work with data of table 5 for numerical measures, and table 6 for alphanumeric measures.

Dimensions			Measure	Aggregation du
Location	**Product**	**Time**	**Sales**	**SUM**
Bellingham	*Milk*	*Quarter 1-1997*	*20*	
Bellingham	Milk	*Quarter 2-1997*	*46*	
Bellingham	Milk	Quarter 3-1997	?	
Bellingham	*Milk*	*Quarter 4-1997*	*?*	
Bellingham	*Milk*	*1997*		*90*
Bellingham	*Cola*	*Quarter 1-1997*	*?*	
Bellingham	*Cola*	*Quarter 2-1997*	*?*	
Bellingham	*Cola*	Quarter 3-1997	10	
Bellingham	*Cola*	*Quarter 4-1997*	*10*	
Bellingham	*Drink*	*Quarter 1-1997*		*35*
Bellingham	*Drink*	*Quarter 2-1997*		*40*

Table 5. *Facts with missing numerical values.*

Dimensions		Measure	Aggregation function
Location	**Time**	**Product**	**Distinct count (product)**
Carrefor	*Quarter1-1998*	*?*	
Carrefor	*Quarter2-1998*	*Milk*	
Carrefor	*Quarter3-1998*	*Gyn*	
Carrefor	*Quarter4-1998*	*?*	
Carrefor	*1998*		*3*
CA	*Quarter 1-1998*		*2*
Bene	*Quarter1-1998*	*?*	
Bene	*Quarter2-1998*	*Sparklin*	
Bene	*Quarter3-1998*	*Tonic*	
Bene	*Quarter4-1998*	*?*	
CA	*Quarter4-1998*		*1*
Bene	*1998*		*4*

Table 6. *Facts with missing alphanumerical values.*

6.1 Preliminaries

To use the multidimensional dataset represented by the warehoused data in a linear programming approach (Sec 6.2), its formalization as uni-dimensional array is needed.

Uni-dimensional-array of facts: Let *d* dimensions, nb_k the number of members of the level DL_k of the dimension D_k, m_k^i the i-th member of DL_k and the position of m_k^i noted as i_k**,** then the position uni-dimensional array of the fact *f* with $cf\,(m_1^i, \ldots, m_d^i)$ is:

$$\sum_{k=1}^{d}\left(\prod_{l=k+1}^{d} nb_l\right) i_k \qquad (1)$$

Example 13. Let the position of the members: "Bellingham" = 0, "Milk" = 0, "Cola"=1, and "Quarter 1-1997" =0, "Quarter 2-1997"=1, "Quarter 3-1997"=2, "Quarter 4-1997"=3, then, position of cf=$[Belligham, Cola, Quarter\ 2-1997]$ in the uni-dimensional array is 5 (0*2*4+1*4+1) (Figure 4).

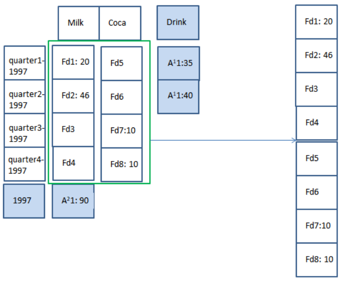


Figure 4. *Uni-dimensional array of table 5*

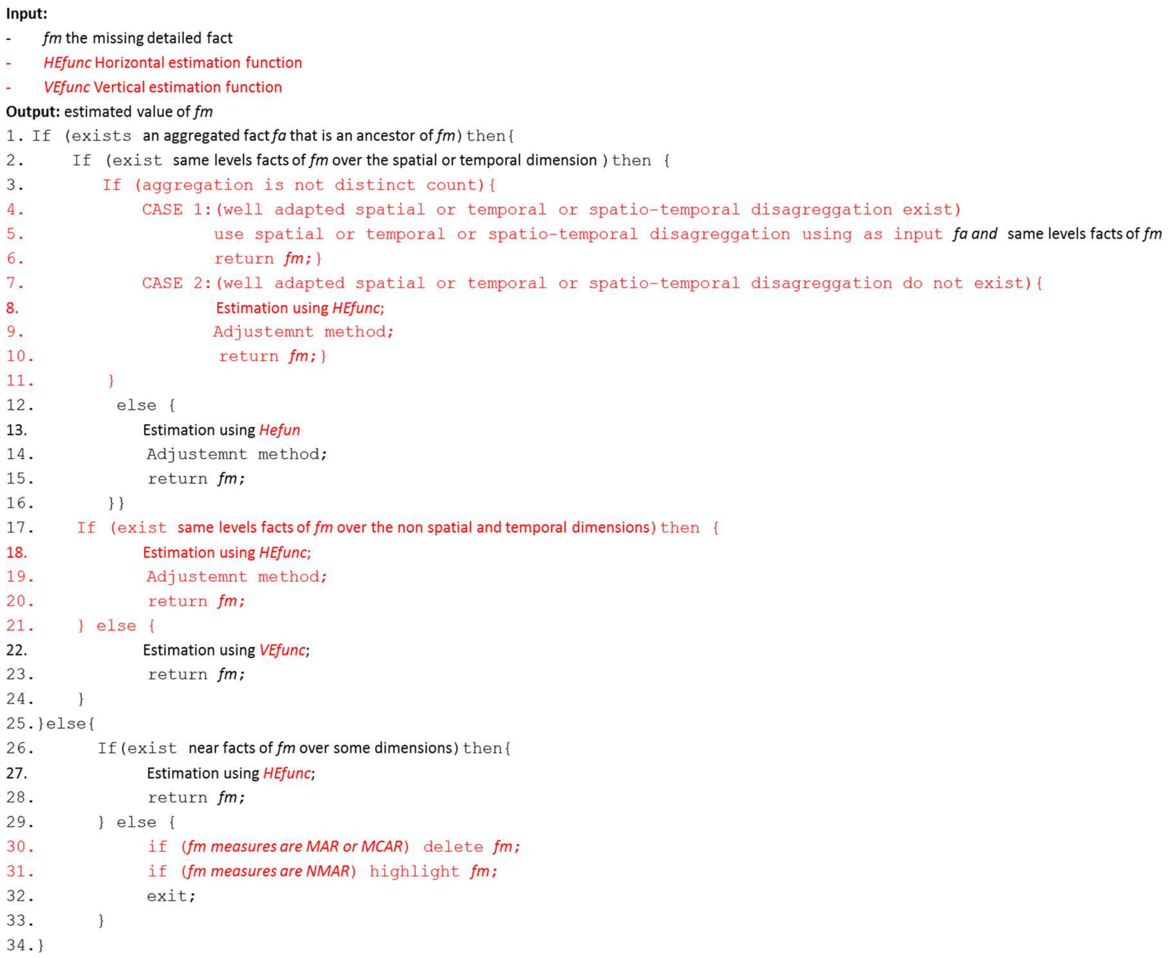
Figure 4 shows two views of table 5: a multidimensional representation (on the left), where each cell has a unique index that corresponds to its address in a uni-dimensional array, and a uni-dimensional array (right) that is the projection of the multidimensional representation.

6.2 Adjustment approach

In this section, we formalize the problem of missing values imputation in our multidimensional and multi-granular model represented using a uni-dimensional array (Sec 5.2.1). We propose a linear programming solution in Section 6.2.2

6.2.1 Overview of the adjustment approach

Our generic imputation process for multi-granular DW is described in the pseudo-algorithm of figure 5.



```
Input:
-   fm the missing detailed fact
-   HEfunc Horizontal estimation function
-   VEfunc Vertical estimation function
Output: estimated value of fm
1. If (exists  an aggregated fact fa that is an ancestor of fm) then{
2.     If (exist  same levels facts of fm over the spatial or temporal dimension ) then {
3.        If (aggregation is not distinct count){
4.            CASE 1:(well adapted spatial or temporal or spatio-temporal disagreggation exist)
5.                   use spatial or temporal or spatio-temporal disagreggation using as input fa and  same levels facts of fm
6.                   return fm;}
7.            CASE 2:(well adapted spatial or temporal or spatio-temporal disagreggation do not exist){
8.                   Estimation using HEfunc;
9.                   Adjustemnt method;
10.                  return fm;}
11.       }
12.        else {
13.           Estimation using Hefun
14.           Adjustemnt method;
15.           return fm;
16.       }}
17.    If (exist  same levels facts of fm over the non spatial and temporal dimensions) then {
18.           Estimation using HEfunc;
19.           Adjustemnt method;
20.           return fm;
21.    } else {
22.           Estimation using VEfunc;
23.           return fm;
24.    }
25.}else{
26.      If(exist  near facts of fm over some dimensions) then{
27.           Estimation using HEfunc;
28.           return fm;
29.      } else {
30.           if (fm measures are MAR or MCAR) delete fm;
31.           if (fm measures are NMAR) highlight fm;
32.           exit;
33.      }
34.}
```

Figure 5. *Pseudo-algorithm of imputation process*

The adjustment algorithm takes as input: the missing detailed fact (*fm*), a particular imputation function (*VEfunc*) belonging to the vertical category, and a particular imputation function (*HEfunc*) belonging the horizontal category. Data experts according to the characteristics of the warehoused data choose *VEfunc and HEfunc*. The impact of particular computations of *VEfunc* and *HEfunc* is transparent for our approach.

Then, the proposed method first checks the possibility to use aggregated facts (line 1). Otherwise, only the horizontal function *HEfunc* can be applied (lines 11-33). If an aggregated fact and same levels facts exist over the spatial and/or temporal dimensions, and the aggregation function used is not the distinct count, then the algorithm let the decision-makers use an existing disaggregation method (when it exists) (lines 3-10). Otherwise, our approach is used: the horizontal function *HEfunc* is used for estimation, and then the adjustment method is used (lines 7-10). Otherwise, the vertical method

(*VEfunc*) is applied (line 22). In the case when no aggregated facts and same levels facts exist, then two cases are possible: if the measures are Missing at Random (MAR) or Missing Completely at Random (MCAR), then the fact is deleted; if measures are Missing not at Random (MNAR), then the fact is kept but a special visualization in the OLAP client is provided. MAR, NMAR and MCAR characterize the missingness of the values. In particular (Little et al., 2002):

- Missing at Random (MAR): Missing at random means that the propensity for a data point to be missing is not related to the missing data, but it is related to some of the observed data.
- Missing Completely at Random (MCAR): The fact that a certain value is missing has nothing to do with its hypothetical value and with the values of other variables.
- Missing not at Random (MNAR): Two possible reasons are that the missing value depends on the hypothetical value (e.g. people with high salaries generally do not want to reveal their incomes in surveys) or missing value is dependent on some other variable's value.

In the first two cases, it is safe to remove the facts with missing values depending upon their occurrences, while in the third case removing facts with missing values can produce a bias in the OLAP analysis.

6.2.2 Problem formalization

Let us consider that missing facts, which have not aggregated values, have been estimated using horizontal or vertical functions. Therefore, in the rest of the section, we denote:

- *Fd* the uni-dimensional array composed of detailed facts,
- *Fe* the uni-dimensional array of missing facts, and
- *Ai* the uni-dimensional array of aggregated facts on the dimension *Di*.

Example 14. Let table 7 representing data of table 5 with estimated values.

Dimensions			**Measure**	**Aggregation func**
Location	**Product**	**Time**	**Quantity**	**SUM**
Bellingham	*Milk*	*Quarter 1-1997*	*20*	
Bellingham	*Milk*	*Quarter 2-1997*	*46*	
Bellingham	*Milk*	*Quarter 3-1997*	***33***	

Bellingham	*Milk*	*Quarter 4-1997*	***33***	
Bellingham	*Milk*	*1997*		*90*
Bellingham	*Cola*	*Quarter 1-1997*	***10***	
Bellingham	*Cola*	*Quarter 2-1997*	***10***	
Bellingham	*Cola*	*Quarter 3-1997*	*10*	
Bellingham	*Cola*	*Quarter 4-1997*	*10*	
Bellingham	*Drink*	*Quarter 1-1997*		*35*
Bellingham	*Drink*	*Quarter 2-1997*		*40*

Table 7. *Facts with estimated values*

Examples of the vector *Fd* and *Ai* using a sample of the data of our case study are shown in figure 6 and figure 7.

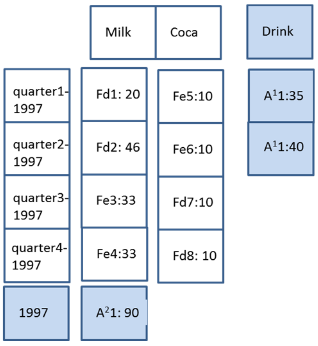
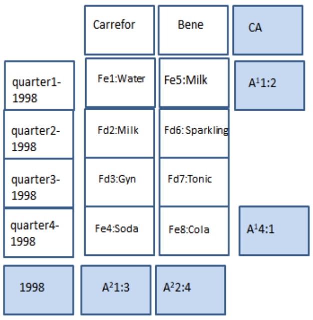


Figure 6. *Uni-dimensional representation of facts of table 5*



□

Figure 7. *Uni-dimensional representation of facts of table 6*

The main idea of the proposed adjustment method is to modify the estimated values, so that they comply with the aggregated fact.

Adjustment: Let an estimated value fe_k, the adjusted fact fd_k is a detailed fact such as: $fd_k = fe_k \mp \Delta_k$.

Then, we can define a global adjustment as finding $\vartriangle$ such that:

$$\vartriangle = \sum_{k} |fd_k - fe_k| \; (2)$$

6.2.3 Linear programming formulation

The adjustment step is defined as a linear programming model that minimizes a linear function with some constraints. The constraints may be equalities or inequalities.

Objective Function: Adjusted values consist of minimizing (2) and can be considered as the objective function of a linear program. Formally:

$$Minimize \sum_k \; |fd_k - fe_k| \; (3)$$

Constraints: Aggregated facts can be considered as a constraint as follows:

$$Agg\left(descendantsFacts(A_j^i), A_j^i\right) \leq \; A_j^i \quad \forall i = 1, \dots, d, \forall j = 1, .., \omega(i) \; (4)$$

where Agg is a generic aggregation function that takes as inputs the detailed facts and the aggregated fact, $\omega(i)$ is the number of aggregated facts of the *i-th* dimension..

In order to solve it, we have to define:

- the objective function as a linear function, and
- the aggregation constraint for the different aggregation functions.

6.2.3.1 Linearization of the objective function

As previously defined, the objective function $Minimize \sum_k \; |fd_k - fe_k|$ is not linear since the absolute value function is not a linear function. However, it is possible to reformulate it as linear equations.

The linearization of the objective function consists of using binaries variables B_i and adding variables X_i' as follows:

$$Minimize \sum_k \; |fd_k - fe_k| = Minimize \sum_k \; X_k'$$

with the following constraints:

$$\begin{array}{l} fd_k - fe_k \; + \; \eth \; * \; b_k \; >= \; X_k' \\ -fd_k + fe_k + \; \eth \; * \; (1 \; - \; b_k) >= X_k' \quad (5) \\ fd_k - fe_k \; <= \; X_k' \end{array}$$

$$-fd_k + fe_k <= X'_k$$

where ð is a big integer, and b_k is a Boolean variable

Example 16. Let us consider figure 4, missing fact are fd_3, fd_4, fd_5 and fd_6. Let suppose their estimated values are fe_3, fe_4, fe_5 and fe_6, respectively Then, the objective function is: *Minimize* $X'_3 + X'_4 + X'_5 + X'_6$ with the following constraints:

$$fd_3{:}?-fe_3 : 33 + \eth * b_3 >= X'_3 \qquad -fd_3{:}?+fe_3{:}33 + \eth * (1 - b_3) >= X'_3$$

$$fd_3{:}?-fe_3 : 33 <= X'_3 \qquad - fd_{3:}{:}?+fe_3{:}33 <= X'_3$$
$$fd_4{:}?-fe_4 : 33 + \eth * b_4 >= X'_4 \qquad -fd_4{:}?+fe_4{:}33 + \eth * (1 - b_4) >= X'_4$$
$$fd_4{:}?-fe_4 : 33 <= X'_4 \qquad - fd_4{:}?+fe_4{:}33 <= X'_4$$
$$fd_5{:}?-fe_5 : 10 + \eth * b_5 >= X'_5 \qquad -fd_5{:}?+fe_5{:}10 + \eth * (1 - b_5) >= X'_3$$
$$fd_5{:}?-fe_5 : 10 <= X'_5 \qquad - fd_5{:}?+fe_5{:}10 <= X'_5$$
$$fd_6{:}?-fde_6 : 10 + \eth * b_6 >= X'_6 \qquad -fd_6{:}?+fe_6{:}10 + \eth * (1 - b_6) >= X'_6$$
$$fd_6{:}?-fe_6 : 10 <= X'_6 \qquad - fd_6{:}?+fe_6{:}10 <= X'_6 \qquad \square$$

6.2.4 Aggregation constraints

The aggregation constraint depends on the used aggregation function. Indeed, the linearity of the aggregation constraints of distributive aggregation functions (i.e. sum, count and average) is not the same for holistic function such as the distinct count.

Therefore, in this section we present the linearity of the sum, min, max and count functions (Section 6.2.4.1), since they are representative of SQL distributive aggregation functions, the average for algebraic aggregation functions (Section 6.2.4.2), and the distinct count for holistic functions (Section 6.2.4.3).

6.2.4.1 Distributive aggregation functions: SUM, MIN, MAX and COUNT

In the above problem formalization, the aggregation function is generic. Therefore, in the following we specify it for the different aggregation functions.

For the aggregation function SUM, we can define it as described in the following. Let us note that the COUNT aggregation function is represented in the same way.

Aggregation constraint for SUM and COUNT.

The sum aggregation constraint can be defined as:

$$\sum \; descendantsFacts\left(A_j^i\right) < A_j^i \quad \forall i = 1, \dots, d, \forall j = 1, .., \omega(i) \qquad (6)$$

where $\omega(i)$ is the number of aggregated facts of the *i-th* dimension.

Example 17. Let consider figure 4. $A_1^2: 90$ is a known aggregate and his constraint can be defined as follows: $fd_1: 10 + fd2: 46 \; + fd3: 33 + fe_4: 33 \leq A_1^2: 90$

For the count, we suppose a function *ONE* that assigns the value "1" to a fact.

Then,

$$\sum \quad ONE\left(descendantsFacts\left(A_j^i\right)\right) < A_j^i \;\; \forall i = 1, \ldots, d, \forall j = 1, .., \omega(i) \;(7)$$

Aggregation constraint for MIN and MAX.

The aggregation constraint for MAX and MIN can be defined as:

MAX: ∀ fact f of $descendantsFacts\left(A_j^i\right)$

$$f \leq A_j^i \;\; \forall i = 1, \ldots, d, \forall j = 1, .., \omega(i) \qquad (8)$$

MIN: ∀ fact f of $descendantsFacts\left(A_j^i\right)$

$$f \geq A_j^i \;\; \forall i = 1, \ldots, d, \forall j = 1, .., \omega(i) \qquad (9)$$

6.2.4.2 AVERAGE algebraic aggregation function

Constraints of AVERAGE is defined by combining distributive constraints as follows:

$$\frac{\sum \quad descendantsFacts\left(A_j^i\right)}{\sum \quad ONE\left(descendantsFacts\left(A_j^i\right)\right.} \leq A_j^i$$
$$\forall i = 1, \ldots, d, \forall j = 1, .., \omega(i) \qquad (10)$$

6.2.4.3 DISTINCT COUNT holistic function

Let us consider the distinct count as holistic aggregation function for the aggregation constraint (Agg). As shown in section 2, the distinct count is usually applied to dimensions members that are alphanumeric values. Then, we will replace members' names with identifying integers. This choice does not affect the linear programming based approach that we propose as shown in the rest of the section.

Example 18. Figure 8 shows how members of figure 7 are replaced by integer identifiers.

Then, to translate the distinct count into a summative function we introduce Boolean variables y_k instead of x_k.

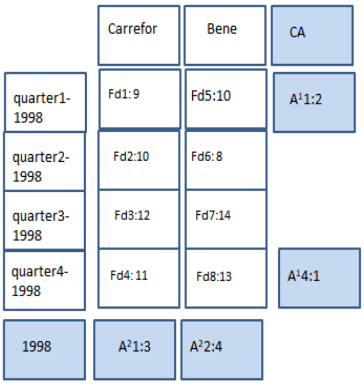
Example 19. Let us consider figure 8, the following constraints are expressed using the variable y_k,

$y_1 + y_2 + y_3 + y_4 < A_1^2 \qquad y_5 + y_6 + y_7 + y_8 < A_2^2 : 4 \qquad y_1 + y_5 < A_1^1 : 2$

$$y_4 + y_8 < A_4^1 : 1$$

$$3 = A_1^2 \quad where \qquad y_k \in [0,1]$$

□



	Carrefor	Bene	CA
quarter1-1998	Fd1:9	Fd5:10	A¹1:2
quarter2-1998	Fd2:10	Fd6:8	
quarter3-1998	Fd3:12	Fd7:14	
quarter4-1998	Fd4:11	Fd8:13	A¹4:1
1998	A²1:3	A²2:4	

Figure 8. *Members of figure 8 are replaced with integer identifiers.*

To solve y_k , we introduce new binaries variables Z_{kt} such that:

$$\{ Z_{kt} = 1 \; if \;\; Fd_k \neq Fd_t \qquad Z_{kt} = 0 \; if \; if \; Fd_k = Fd_t$$

(11)

Where: $\quad y_k = \prod_t \quad Z_{kt}$ (12)

Then, the relaxation technique helps to approximate (12) as follows:

$$\frac{|Fd_{k,} - Fd_t|}{ð} \leq Z_{kt} \leq |Fd_{k,} - Fd_t| * ð \qquad (13)$$

where ð is a big number.

From (11) and (12), we can deduce the constraints of y_k with respect to Z_{kt} as follow:

$$y_k \leq Z_{kt} \; \forall t = 1, .., k - 1 \qquad (14)$$

Moreover, let us note that (13) must also be linearized, since it has absolute values. Then, we can rewrite (13) as:

$(Fd_k - Fd_t) * \; ð + p_k \leq Z_{kt}$

$(-Fd_k + Fd_t) * ð + p_k \leq Z_{kt} \qquad k = 1, .., N, t = 1, .., k - 1$ (15)

$(Fd_k - Fd_t)/ð + ð * b_k + \;\; p_k \geq Z_{kt}$

$(-Fd_k + Fd_t)/ð + ð * b_k + p_k \leq ð - Z_{kt}$

$b_k, p_k \in \{0,1\}$

Let us note that $Fd_k - Fd_t$ allows to know if Fd_k is different from Fd_t.

Example 20. Let us consider the variable of y_1 of example 19. According to (14), the following constraints are defined:

$Y_1 \leq Z_{12}$ $Y_1 \leq Z_{13}$ $Y_1 \leq Z_{14}$ $Y_1 \leq Z_{15}$ $Y_1 \leq Z_{16}$ $Y_1 \leq Z_{17}$ $Y_1 \leq Z_{18}$

Moreover, for example for Z_{14}, the following constraints are defined according to (13):

$(Fd_1 - Fd_4{:}11) * ð + p_1 \leq Z_{14}$

$Fd_1 = 9$

$(-Fd_1{:}9 + Fd_4{:}11) * ð + P_1 \leq Z_{14} \quad k = 1,..,N, t = 1,..,k-1$

$(Fd_1{:}9 - Fd_4{:}11)/ð + ð * B_1 + P_1 \geq Z_{14}$

$(-Fd_1{:}9 + Fd_4{:}11)/ð + ð * B_1 + P_1 \leq ð - Z_{14}$

$B_k, P_k \in \{0,1\}$

We formally define linear constraints of distinct count.

Aggregation constraint for distinct count.

$\sum \quad y < A_j^i \ \forall i = 1, \dots, d, \forall j = 1$

$y_k \leq Z_{kt} \ \forall t = 1,..,k-1$

$(Fd_k - Fd_t) * ð + p_k \leq Z_{kt}$

$(-Fd_k + Fd_t) * ð + p_k \leq Z_{kt} \quad k = 1,..,N, t = 1,..,k-1$ (16)

$(Fd_k - Fd_t)/ð + ð * b_k + p_k \geq Z_{kt}$

$(-Fd_k + Fd_t)/ð + ð * b_k + p_k \leq ð - Z_{kt}$

$b_k, p_k \in \{0,1\}$

Using our approach and the constraints for all y_k and Z_{14}variables, one of the solutions is shown in table 8. We can note that estimated data of table 8 respect the distinct count aggregation constraint.

Dimensions		**Measure**	**Aggregation function**
Location	**Time**	**Product**	**Distinct count (product)**
Carrefor	*Quarter1-1998*	*Water*	
Carrefor	*Quarter2-1998*	*Milk*	
Carrefor	*Quarter3-1998*	*Gyn*	
Carrefor	*Quarter4-1998*	*Water*	
Carrefor	*1998*		*3*
France	*Quarter 1-1998*		*2*
Bene	*Quarter1-1998*	*Milk*	

Bene	*Quarter2-1998*	*Sparklin*	
Bene	*Quarter3-1998*	*Tonic*	
Bene	*Quarter4-1998*	*Water*	
France	*Quarter4-1998*		*1*
Bene	*1998*		*4*

Table 8. *Possible estimated solutions for table 6*

7 Experiments and validation

This section describes the experiments that we have carried out using the proposed adjustment approach on two datasets:

- **FoodMart** is a multidimensional dataset issued with many OLAP servers as described in section 2.
- **Car Evaluation** consists of six dimensions: Buying (buying price), Maint (price of the maintenance), Doors (number of doors), Persons (capacity in terms of persons to carry), Lug_boot (the size of luggage boot), and Safety (estimated safety). The measure is the acceptability of the car (Good, Not Good, Average, etc.).

DW and data mining research and industrial communities to evaluate our proposal commonly use these two datasets. Other academic benchmarks also exist (such as SSB, TPC-DS, etc.), but they are more complex and very similar to the ones used in this work.

The proposed approach is evaluated in terms of quality and computational time. The obtained results are compared to other approaches in the section 7.1, and the computation time is presented in section 7.2.

The experiments have been conducted on a laptop with an Intel Core-i5 CPU (2.3 GHz) with Windows 7 (64 bits). Horizontal and vertical functions have been implemented in C++. The proposed approach is implemented in C++ using IBM Cplex 12.6 libraries. Data are stored in the Postgresql.

7.1 Quality evaluation

Since aggregation constraints are related to the nature of aggregation function, we evaluate the quality of our adjustment method over three different aggregation functions, each belonging to a different category of aggregation function: sum (distributive), average (algebraic) and distinct count (holistic). Then, we compare our methodology to some horizontal and vertical functions. For the horizontal function, we evaluate if these methods violate or not the aggregation constraints.

Then, we evaluate the vertical function, which does not violate aggregation constraints, and we compare it to our proposal. This comparison is based on "Mean Absolute Error" defined as $MAE = \frac{1}{n}\sum_{i=1}^{n} \quad |fd_i - fe_i|$

We also study the dependence of the number of missing facts and the quality of the results.

7.1.1 Distributive aggregation functions: SUM

We consider the case of sum that is a distributive function. For its evaluation, we use only the FoodMart dataset, since it comes with numeric measures. Some measures are aggregated facts per year.

For the horizontal estimation function, we have used the following methods:

- "mean", which consists of using the average of known values, and
- "most frequent", which replaces the missing value by the most frequent known value.

As shown in figure 9, these two horizontal functions lead to conflicts with the aggregation constraints. All constraints are violated in the case of "most frequent", whatever the percentage of values missing. Constraints are also violated for the "mean". Then, it makes no sense to compare horizontal functions with our method, since our adjustment proposal always respects aggregation constraints.

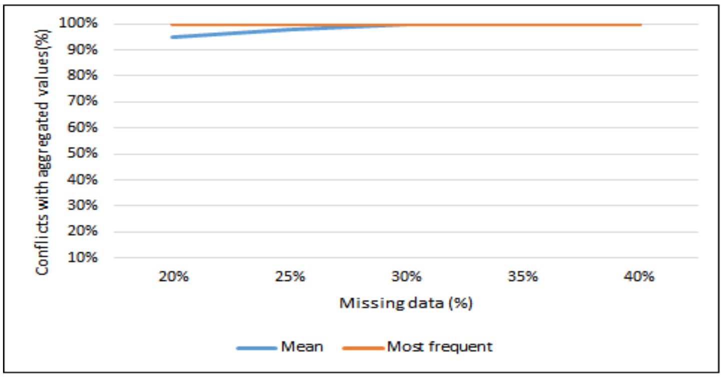


Figure 9. *FoodMart and SUM as aggregation constraint: constraints violation of "mean" and "most frequent" horizontal functions*

Figure 10 shows the comparison among our proposal and some common vertical functions: "Min", "Max" and "Split" (Bertino et al., 2010). These methods have not conflicts with aggregation constraints. Figure 10 shows that our proposal has a fewer MAE value compared to the three vertical functions. Moreover, the MAE value of the adjustment increases quasi linearly according to the number of missing values.

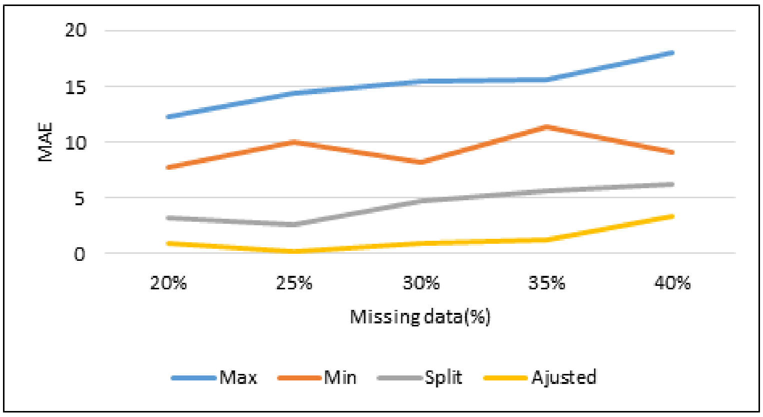


Figure 10. *FoodMart and SUM as aggregation constraint: quality of estimation of vertical functions (min, max, split) and our adjustment approach*

7.1.2 Algebraic aggregation functions: AVG

In this section, we consider the average as aggregation function. We observed that the violation of the aggregation constraints are the same of the sum (figure 10). This is obvious since the aggregated value is not used by the horizontal estimation functions.

To test the quality of the adjustment, approach we compare it with vertical functions Max, Min and Split. The results are shown in figure 11. Our proposal outperforms all the vertical functions. Like the case of sum, the MAE value of our adjustment approach grows quasi linearly with the number of missing data.

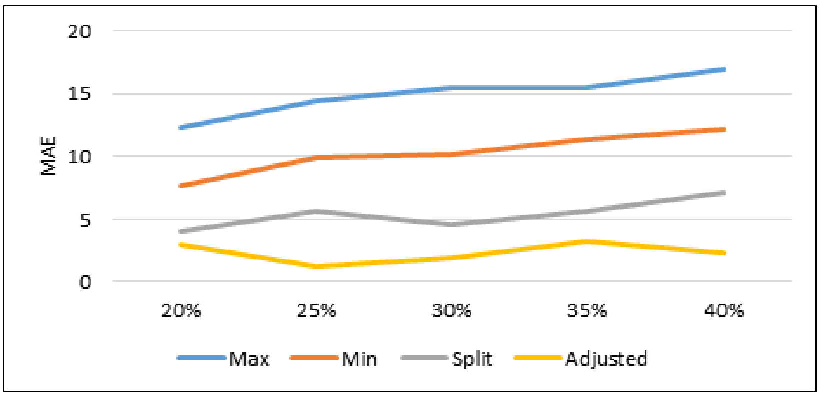


Figure 11. *FoodMart and AVG as aggregation constraint: quality of estimation of Vertical functions (min, max, split) and our adjustment approach*

7.1.3 Holistic aggregation functions: Distinct Count

To evaluate the case of "distinct count", we consider the two datasets "Car evaluation" (Sec 7.1.3.1) and "FoodMart" (7.3.1.2) since they provide very different results, which are discussed in section 7.3.1.3.

7.1.3.1 Car Evaluation

The Car Evaluation Database is created (Bohanec & Rajkovic, 1990) has six dimensions and the measure is the car evaluation value. The problem consists of estimating unfilled "Evaluation" measure. An example of the Car Evaluation dataset is shown in table 9.

Dimension						**Measure + Aggregation functions**	
						Measure	**Agg. Functions**
Buying	**Maint**	**Doors**	**Persons**	**Lug_boot**	**Safety**	**Evaluation**	**Distinc count**
Vhigh	Vhigh	3	4	Big	high	?	
Vhigh	High	3	4	Big	high	?	
Vhigh	Med	3	4	Big	high	Acc	
Vhigh	Low	3	4	Big	high	Acc	
High	Vhigh	3	4	Big	high	Unacc	
High	High	3	4	Big	high	Acc	
High	Med	3	4	Big	high	Acc	
High	Low	3	4	Big	high	?	
Med	Vhigh	3	4	Big	high	?	
Med	High	3	4	Big	high	Good	
Med	Med	3	4	Big	high	Vgood	
Med	Low	3	4	Big	high	Vgood	
Low	Vhigh	3	4	Big	high	Acc	
Low	High	3	4	Big	high	Vgood	
Low	Med	3	4	Big	high	Vgood	
Low	Low	3	4	Big	high	Vgood	
All	Vhigh	3	4	Big	high		2
All	High	3	4	Big	high		4
All	Med	3	4	Big	high		2
All	Low	3	4	Big	high		2

Vhigh	All	3	4	Big	high		2
High	All	3	4	Big	high		2
Med	All	3	4	Big	high		3
Low	All	3	4	Big	High		2

Table 9. *Car Evaluation: instances of the multi-granular facts with missing value*

We use the popular machine learning algorithms J48 and Random Forest (RF) supported by Weka.

We first randomly generate 20%, 25%, 30% and 35% of missing measure using the WEKA methods. Let us note that this experiment concerns the usage of our approach to estimate missing values. The MAR, MCAR and MNAR characterization of simulated measure values is not taken into account in this experiment, since we use them only when the estimation is not possible to delete or highlight measures (lines 30-35 of algorithm of Figure 5). Then, we estimate of missing "Evaluation" with J48 and RF, and we found that all constraints are violated with J48 and RF, as shown in figure 12.

We apply the adjustment algorithm to help respect constraints, as shown in figure 13.

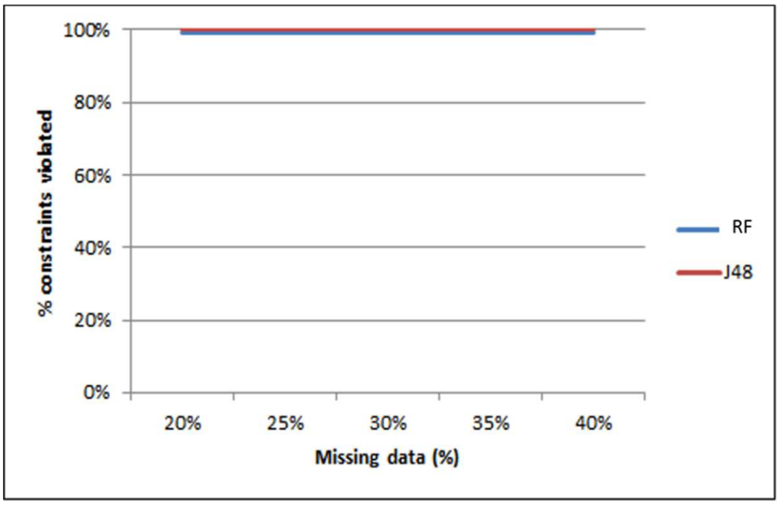


Figure 12. *Car Evaluation and distinct count as aggregation constraint: constraints violation of "RF" and "J48" horizontal functions*

Concerning the quality of estimation after adjustment, figure 14 shows the curve of the two methods in term of percentage of well-estimated elements using J48 and RF algorithms.

After adjustment, J48 has better result than RF. The quality of the result after the adjustment depends on the horizontal function used.

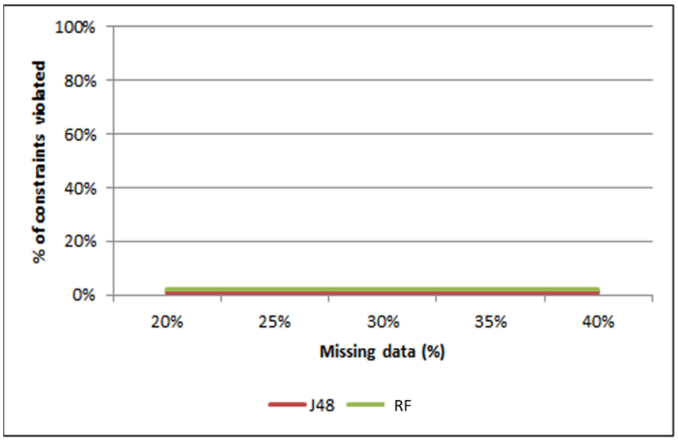


Figure 13. *Car Evaluation and distinct count as aggregation constraint: constraints violation of "RF" and "J48" horizontal functions after our adjustment approach*

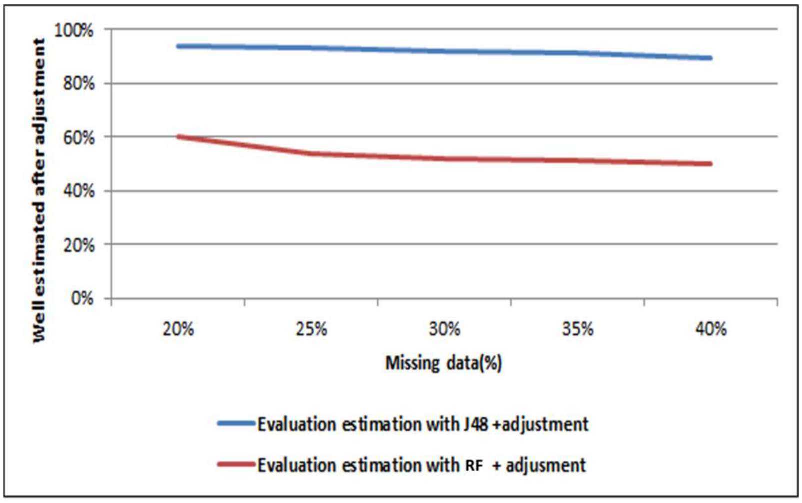


Figure 14. *Car Evaluation and distinct count as aggregation constraint: quality of estimation of missing value with J48 and RF with our adjustment approach*

7.1.3.2 FoodMart

We used for this experiment FoodMart that has been described in section 2. The problem is to estimate missing products. Estimation of missing products is important for stock management analysis. This case involves an alphanumerical measure, and its associated aggregation function is distinct count.

We use the machine learning algorithm J48 and RF for the estimation. To evaluate the quality of the estimation, we first randomly generate 20%, 25%, 30% and 35% of missing measure. After the estimation, as shown in figure 15, all constraints are violated with J48 and RF.

We apply the adjustment algorithm to help respect constraints as detailed in figure 15.

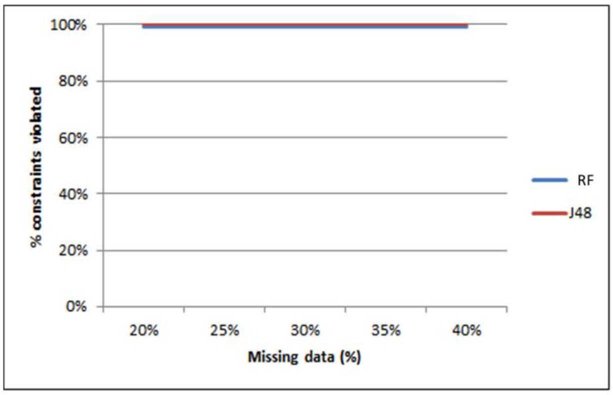


Figure 15. *FoodMart and distinct count as aggregation constraint: constraints violation of RF and J48 horizontal functions*

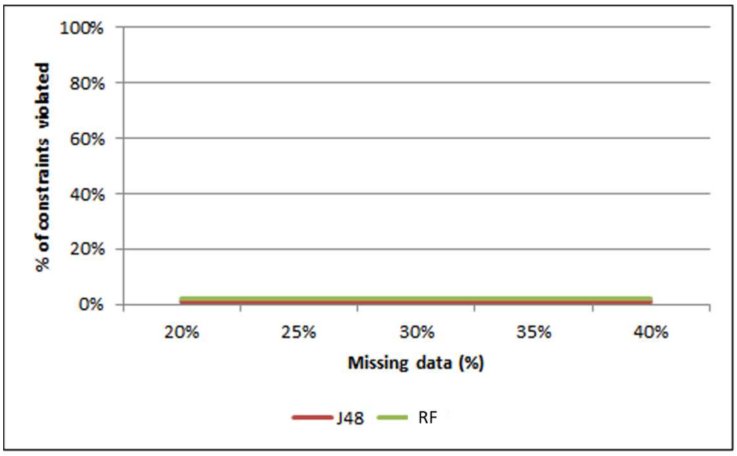


Figure 16. *FoodMart and distinct count as aggregation constraint: constraints violation of RF and J48 horizontal methods after our adjustment approach*

Concerning the quality of estimation after adjustment, figure 17 shows the curve of the two cases in terms of percentage of well-estimated elements using J48 and RF algorithms. The case of J48 is better than RF. However, both give bad results since the two algorithms are not adapted for this data. The result of the estimation after the adjustment depends on the case. The quality of the result degrades when the number of missing values increases.

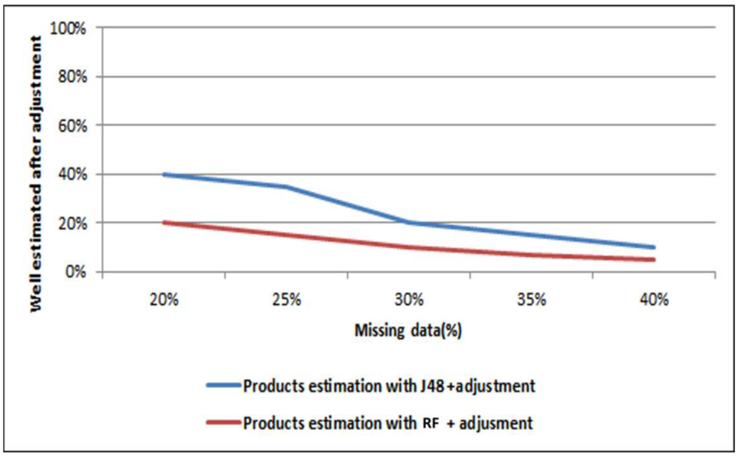


Figure 17. *FoodMart and distinct count as aggregation constraint: quality of estimation of missing value with J48 and RF with adjustment*

7.1.3.3 Discussion

In this section, we provide a discussion of previous described experiments. The performed tests highlight that the case of SUM and AVG of our proposal provides good results in terms of quality estimation. In other terms, the usage of simple estimation functions coupled with the adjustment approach allows to respect aggregation constraints coupled with a good estimation of missing values. The case of Distinct Count is more complicated since it is a holistic function. In all cases, horizontal function should be correctly chosen to have a good estimation quality. Indeed, for the FoodMart dataset the RF and J48 machine learning algorithms do not provide good estimation. Instead, the same functions for the Car Evaluation dataset seems to be effective.

Although, the choice of the horizontal estimation functions depends on the dataset used and the decision-makers skills, these experiments validate the generic aspect of our adjustment approach. Indeed, it can be applied to any aggregation function and horizontal estimation function.

7.2 Computational time

For the time performance evaluation, we have used FoodMart since it contains more data than Car Evaluation. We have tested three different sizes of the fact table: 5000, 10000 and 20000 tuples.

Figure 18 shows the execution time curves of the three cases related to aggregation constraint "sum". Execution time seems feasible for an off-line process.

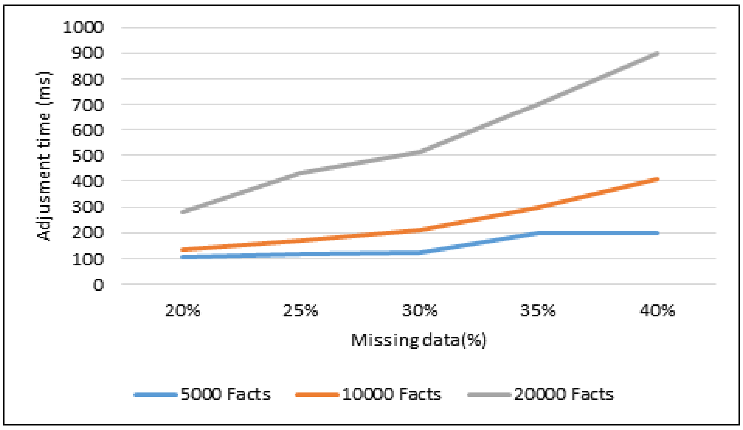


Figure 18. *Execution time of adjustment: cases of sum and average*

Figure 19 presents the execution time of the adjustment when the aggregation constraint is the "distinct count". Also for this case, the execution time rests feasible.

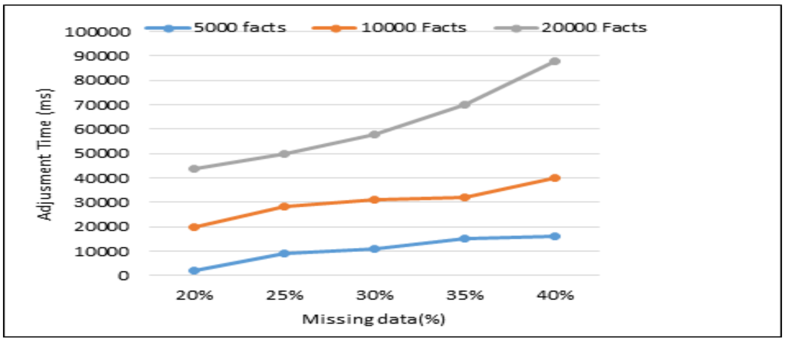


Figure 19. *Execution time of adjustment: case of distinct count*

8 Conclusion and Future work

Data Warehouse (DW) and OLAP systems are first citizens of Business Intelligence tools. They are widely used in the academic and industrial communities for numerous different application domains. In despite of the maturity of DW and OLAP systems, with the advent of Big Data more and more source of data are available and warehousing this data can lead to important quality issues. In this paper, we describe an approach of missing data estimation in multi-granular data warehouse. We first classify estimation methods in two approaches: horizontal approaches, which consist of using facts of the same level of granularity in the estimation and vertical approaches that estimate from known aggregated values. Our method first checks the possibility to use aggregated values, otherwise only horizontal functions can be applied. If an aggregated fact and same levels facts exist, then one horizontal method is used for estimation, and then our adjustment method is used. Otherwise, a vertical method is applied. For the adjustment, we consider aggregated values as constraints to be satisfied, by minimizing the adjustment of result of estimation issuing from horizontal method. We use linear programming techniques to address the constraints problem. For this, we have investigated distributive and algebraic aggregation functions, for holistic case, we investigated only "distinct count" which is a standard function. Experiments results done over FoodMart claim good quality of estimation when all constraints are satisfied.

Some aspects of this paper merit our future work. We plan to investigate the linearization of constraints for other common holistic and algebraic functions (such as median, rank, etc.), and investigate scalability issue for our adjustment approach concerning holistic constraints.

Moreover, we also plan to investigate spatial facts with measures resulting from spatial metric operators such as distance, area, etc. In order to tackle this issue, a formal and explicit representation of spatial data (i.e. geometrical objects) must be introduced in our formal multidimensional model, and then the linear programming algorithm must be adapted with the particular aggregation functions associated to those measures.

Finally, to validate our approach it is also necessary to test it over different real datasets, with advanced estimation functions defined by decision-makers, in order to evaluate the quality improvement provided by our adjustment approach over these functions.

Acknowledgement

This work was supported by the French National Research Agency as part of the "Investissements d'Avenir" through the IDEX-ISITE initiative CAP 20-25 (ANR-16-IDEX-0001), and the ANR project ANR-17-CE04-0012 VGI4bio.

References

Abdelbaki W.,Yahia, B. S., Messaoud, B. R.(2015). Modular Neural Networks for Extending OLAP to Prediction. *Trans. Large-Scale Data- and Knowledge-Centered Systems,* (pp. 73-93).

Ahmed T. et Miquel M. (2005). Multidimensional Structures Dedicated to Continuous Spatiotemporal Phenomena. In : JACKSON et al. *22th British National Conference on Databases, Sunderland, UK. Berlin Heidelberg : Springer*, (pp. 29-40), LNCS 3567.

Amanzougarene, F., Zeitouni, K., Chachoua, M., (2014). Predicting Missing Values in a Data Warehouse by Combining Constraint Programming and KNN, In *proceedings of EDA'14, (*pp.145-154).

Faten Atigui, Franck Ravat, Jiefu Song, Olivier Teste, Gilles Zurfluh: Facilitate Effective Decision-Making by Warehousing Reduced Data: Is It Feasible? IJDSST 7(3): 36-64 (2015)

Bimonte, S., Pradel, M., Boffety, D., Tailleur, A., Andre, G., Bzikaha, R. & Chanet, JP. (2013). A new sensor-based Spatial OLAP architecture centered on an agricultural farm energy-use diagnosis. *International Journal of Decision Support System Technology*, 5(4), 1-20.

Bimonte S. (2015). Spatial OLAP for agri-environmental data and analysis: Lessons learned. *38th International Convention on Information and Communication Technology, Electronics and Microelectronics,{MIPRO}, (pp. 1393-1398).*

Bohanec, M. and Rajkovic, V. (1990). Expert system for decision making. *Sistemica 1*, (pp. 145-157).

Boulil, K., Bimonte, S., Pinet, F. (2015). Conceptual model for spatial data cubes: A UML profile and its automatic implementation. *Computer Standards & Interfaces*, 38, 113-132.

Camossi E., Bertolotto M. et Bertino E. (2006). A multigranular object-oriented framework supporting spatio-temporal granularity conversions. *International Journal of Geographical Information Science*, 20(5), 511-534.

Chen, B. C., Chen, L., Lin, Yi., Ramakrishnan, R. (2005). Prediction cubes In VLDB Endowment, 31st international conference on Very large data bases (pp. 982-993).

E. B. Dagum and P. A. Cholette. Benchmarking, Temporal Distribution, and Reconciliation Methods for Time Series. Lecture Notes in Statistics. Springer-Verlag, New York, 2006.

Damiani, Maria Luisa, and Stefano Spaccapietra. "Spatial data warehouse modelling." Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications. IGI Global, 2008. 659-678.

F. T. Denton. Adjustment of monthly or quarterly series to annual totals: An approach based on quadratic minimization. Journal of the American Statistical Association, 66:99–102, Mar. 1971

Dyreson C. E., Pedersen T. B. & Jensen C. S. (2003). Incomplete information in multidimensional databases. In M. Rafanelli (ed.), *Multidimensional Databases* (pp. 282-309). Idea Group Publishing. Hershey, PA: Information Science Publishing.

Eekhout I., Boer M. R.,Twisk JosW. R., Vet Henrica C.W. & Heymans M. W. (2012). Missing data : a systematic review of how they are reported and handled. *Epidemiology*, 23(5), 729-32.

Gagnon, Patrick, et al. "The added value of stochastic spatial disaggregation for short-term rainfall forecasts currently available in Canada." Journal of hydrology 554 (2017): 507-516.

Gallego J. and Peedell S. 2001 Using CORINE land cover to map population density. In Towards agri-environmental indicators. Integrated statistical and administrative data with land cover information. Copenhagen: European Environment Agency, pp. 94-105.

Graham, J.W., Olchowski, A.E. & Gilreath, T.D. (2007). How Many Imputations are Really Needed? Some Practical Clarifications of Multiple Imputation Theory. *Prev Sci*, 8 , 206-213.

Gray, J., Chaudhuri, S., Bosworth, A. et al. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1, 29-53.

Gascueña, C. & Guadalupe, R. (2009). A Multidimensional Methodology with Support for Spatio-Temporal Multigranularity in the Conceptual and Logical Phases. In Progressive Methods in Data Warehousing and Business Intelligence: Concepts and Competitive Analytics (pp. 194-230). Hershey, PA, USA: IDEA Group Publishing.

Hegner, Stephen J., and M. Andrea Rodríguez. "A model for multigranular data and its integrity." Informatica 28.1 (2017): 45-78.

Iftikhar, N., Pedersen, B. T. (2010). Schema Design Alternatives for Multi-granular Data Warehousing. *Database and Expert Systems Applications, 21th International Conference DEXA* (pp. 111-125).

R.J.A Little and D.B Rubin. Statistical Analysis with Missing Data, Second Edition. John Wiley, New York, 2002, p. 491

Li, Tiebei, et al. "A comparison of spatial disaggregation techniques as applied to population estimation for South East Queensland (SEQ), Australia." Applied GIS 3.9 (2007): 1-16.

Malinowski, Elzbieta, and Esteban Zimányi. "Logical representation of a conceptual model for spatial data warehouses." GeoInformatica 11.4 (2007): 431-457.

M. McGuire, A., Gangopadhyay, A., Komlodi, C., & Swan (2008). A user-centered design for a spatial data warehouse for data exploration in environmental research. *Ecological Informatics* 3(4-5), 273-285

Monteiro, João, et al. "Spatial Disaggregation of Historical Census Data Leveraging Multiple Sources of Ancillary Information." ISPRS International Journal of Geo-Information 8.8 (2019): 327.

Odgers, Nathan P., et al. "Disaggregating and harmonising soil map units through resampled classification trees." Geoderma 214 (2014): 91-100.

Plumejeaud, C., Mathian, H., Gensel, J., Grasland, C., (2011). Spatio-temporal analysis of territorial changes from a multi-scale perspective. *International Journal of Geographical Information Science,* 25(10), 1597-1612.

Quinlan, J. R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings of the Tenth International Conference on International Conference on Machine Learning (pp. 236-243).

Rubin, D. B. (1976). Inference and Missing Data. *Biometrika*, 29, 159-183.

Sair, A., Erraha, B., Elkyal, M., Loudcher, S. (2012). Prediction in OLAP cube. *In International Journal of Computer Science,* 9(3).

Sarawagi, S., Agrawal, R., Megiddo, N. (1998). Discovery-driven Exploration of OLAP Data Cubes. In: *Proceedings of the 6th International Conference on Extending Database Technology (EDBT'1998)*, Valencia, Spain, Springer (pp. 168–182)

Palpanas, T., Koudas, T., Mendelzon, A., (2005). Using datacube aggregates for approximate querying and deviation detection. *IEEE Transactions on Knowledge and Data Engineering*, 17(11), 1465-1477.

Truong, T. M., Amblard, F., Gaudou, B. & Sibertin-Blanc, C. (2014). To calibrate & validate an agent-based simulation model, an application of the combination framework of BI solution & Multi-agent platform. In: *Proceedings of 6th International Conference on Agents and Artificial Intelligence (ICAART),* Angers, France.

Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G. & Cunningham, S.J. (1999). Weka: Practical machine learning tools and techniques with Java implementations. *(Working paper 99/11).* Hamilton, New Zealand: University of Waikato, Department of Computer Science.

Wohlrab L. et Furnkranz J. (2011). A review and comparison of strategies for handling missing values in separate-and-conquer rule learning. *J. of Intelligent Information Systems*, 36(1) :73-98.

Wu X. et Barbara D. (2002). Learning missing values from summary constraints, *SIGKDD Explorations*, 4(1).

Zaamoune, M., Bimonte, S., Pinet, F., Beaune, P. (2013). A New Relational Spatial OLAP Approach for Multi-resolution and Spatio-multidimensional Analysis of Incomplete Field Data. In: *Proceedings of the 15th International Conference on Enterprise Information Systems*, Angers, France.

[i]