New OLAP Operators for Missing Data

Maha Ben Kraiem*,** Kais Khrouf*, Jamel Feki***, Franck Ravat**, Olivier Teste**

*MIR@CL Laboratory, University of Sfax
Airport Road Km 4, P.O. Box. 1088, 3018 Sfax, Tunisia
Maha.BenKraiem@yahoo.com, Khrouf.Kais@isecs.rnu.tn
** IRIT, University of Toulouse
118, route de Narbonne, 31069 Toulouse Cedex 9, France
{Ravat, Teste}@irit.fr
*** University of Jeddah, FCIT, IS dept
Saudi Arabia
jfeki@uj.edu.sa

Abstract. Data analysis of social networks is often impeded by the problem of missing data. Recent studies highlight the negative effects of this problem mainly regarding querying process. The analysis of data social networks would be severely distorted when limited to filled fields (i.e., not null valued fields) whereas missing data are ignored. To overcome the missing data problem, we provide in this paper an extension of classical *Drilldown* and *Rollup* operators in order to support analyses on multidimensional datasets containing missing values of dimension members.

1 Introduction

In the last decade, many social networks such as Facebook, LinkedIn and Twitter have been developed, and they made users perceive the Web as a place where they exchange feelings and opinions as well as contents. However, despite these tools ease the sharing and collaboration between users; they may cause new challenges concerning the relevant exploitation of these User-Generated Contents (UGC)for decision making systems. Thus, new multidimensional models have been proposed for OLAP purposes. The multidimensional modeling comes with a set of specifics such as missing data. Missing data in social networks is a long standing but relatively poorly understood problem. The analysis of social networks is even thwarted by missing values. There are several ways in which researchers can cope with missing values, which are frequently found in data collected in empirical research. The easiest way is to simply ignore the missing data. However, restricting analyses to the observed responses (i.e., not null fields) results in serious loss of information and then decreases the power of statistical results. Some other missing data treatments include weighting procedures, model-based procedures, and imputation. Facing to great amount of missing data in large volumes of data sets, we set a twofold purpose, first increase the efficiency of analysis and, secondly, help the analysts. For this reason, we extend the two classical *Drilldown* and *Rollup* operators; this extension enables the analyst to handle missing data on dimension members. In this context, our previous work proposed integrating data extracted from tweets into a multidimensional model Ben Kraiem et al. (2014). The proposed model reflects on some specifics (e.g., recursive references between tweets) and, in particular, on missing data. We define in this paper, new versions for two popular OLAP operators that take into account the specificity of this model dealing with missing data. This paper is organized as follows. Section 2 reviews related works concerning the processing of missing data in the literature. In Section 3, we present our case study. Section 4 defines the extended versions to handling missing data for each of the two OLAP operators. For each of these operators, we propose a user-oriented definition along with an algorithmic pseudo code translation. Finally, this paper ends with a conclusion that focuses on perspectives for improvements.

2 Related work

To overcome the problems due to missing data, several methods are proposed in the literature. In Sadikov et al. (2011) the authors address the problem of missing data in information cascades [1] . The authors propose a numerical method that, given a cascade model and observed cascade C' , it can estimate properties of the complete cascade C. There are several ways to handle missing data. Popular approach is based on imputation. Imputation procedures replace missing values by plausible estimates. Huisman (2009) performed a simulation study to investigate the of non-response and missing data on the structural properties of social networks, and the ability of some simple imputation techniques to treat the missing network data. The simulations were based on an existing friendship network in school classes.
Adar and Ré (2007)argue that new methods for collecting social network structure, and the shift in scale of these networks, introduce a greater degree of imprecision that requires rethinking on how social network analysis techniques can be applied. The authors proposed a new area in data management, probabilistic databases, whose main research goal is to provide tools to manage and manipulate imprecise or uncertain data such as missing data. Furthermore, Collins.L et al. (2014) has proposed methods which aim at finding approximations to missing data in a dataset by using optimization algorithms to improve the network parameters after which prediction and classification tasks can be performed. The optimization methods that are considered are genetic algorithm (GA), simulated annealing (SA), particle swarm optimization (PSO), random forest (RF) and negative selection (NS). These methods are individually used in combination with auto-associative neural networks (AANN) for missing data estimation; the results obtained are compared. Other approaches have been proposed for the treatment of missing data. For instance, McClean.S.I et al. (2001) consider the problem of aggregation using an imprecise probability data model that allows representing imprecision by partial probabilities and uncertainty using probability distributions. Further to this study, we may conclude that missing data in social networks is a long standing but relatively poorly understood problem. Most of works do not offer tools for the decision maker to manipulate missing data analyses. None of the previous work fully supports carrying out analysis in the case of missing data. In terms of the analysis of missing data in prior work is practically nonexistent. To the best of our knowledge, our contribution in this paper is the first attempt to extend two algebraic OLAP operators in order to support analyses over missing data and most importantly, the first attempt

1. As information or actions spread from a node to node through the social network, a cascade is formed.

to increase the efficiency of analysis and facilitate the analysts' task in case of missing data. In the next section, we will describe a case study of multidimensional model dedicated to the OLAP of tweets that fulfills decision-makers' needs.

3 Case study

In this section, we recall our multidimensional model dedicated to the OLAP of tweets. Further details about this model can be found in (Ben Kraiem et al. (2014),Ben Kraiem et al. (2015a)). FIG 1 depicts the extended multidimensional model for tweets using graphical notations. Once the conceptual model is defined, the logical model can be derived by applying

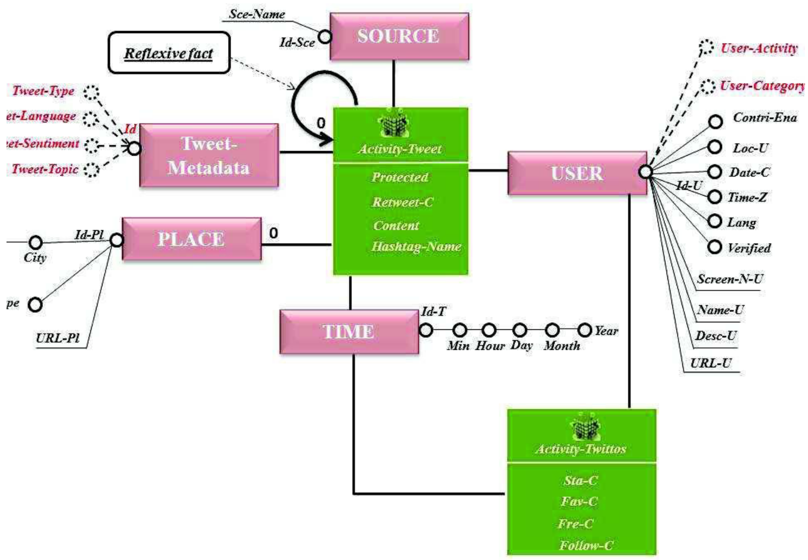

FIG. 1 – *Multidimensional constellation model dedicated for the OLAP of tweets.*

a set of transformation rules (Ben Kraiem et al. (2015b)). FIG 2 depicts the R-OLAP model resulted from the transformation process of the multidimensional constellation diagram (FIG. 1).

The multidimensional data model and implementations of social networks come with a set of further constraints, such as missing data. The analysis of social networks is even more thwarted by missing values. This is the case where there is simply no value provided at all. Technically, the loading process sees a NULL value (Hess (1998)). Existing OLAP operators cannot be successfully applied to handle the above-mentioned challenge. These operators have been defined in a classical context assuming that data are present all the time (Ravat et al. (2008)). So, a remarkable effort must be made to extend these operators to take into consideration the specificity of multidimensional modeling of tweets (missing data). Facing

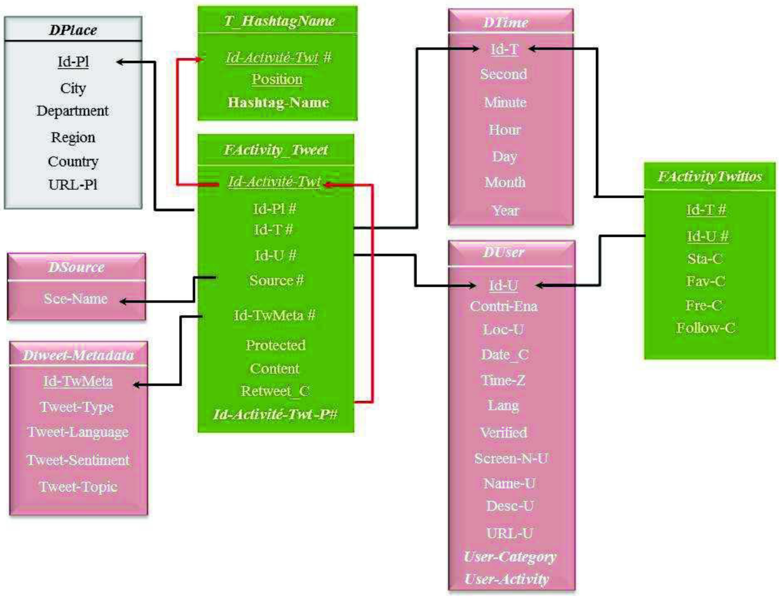

FIG. 2 – *R-OLAP Logical model for constellation of FIG 1.*

to this issue, we propose to extend two OLAP operators Drilldown and Rollup. We call the extended versions *Drilldown* *null-option* and *Rollup* *null-option*, in order to support missing data by offering new options. They both take a multidimensional table currently displayed, an analysis dimension, a parameter and a Null-Option as input. As output, a new multidimensional table is produced containing information at a lower or higher granularity level after executing the *Drilldown* *null-option* and *Rollup* *null-option* operator respectively. For each of these OLAP operators, we propose a user oriented definition along with an algorithmic translation for its implementation.

4 Extended OLAP operators

OLAP analysis results are usually presented in tabular format called *Multidimensional Table* (Gyssens and Lakshmanan (1997); Ravat et al. (2007)).

Definition. Analysis results are presented in forms of multidimensional table, denoted *MT*, which is defined by *(F, MES, Dim, Hier, Pred)* where:

- *F*: is the fact name analyzed in the table,
- *MES* = *{*$f_1(m_1)$,..., $f_p(m_p)$*}* is a set of p measures (m_1),..., (m_p) associated to aggregation functions f_1,..., f_p, f $\subseteq$ {SUM, AVG, MAX ...},
- *Dim* = *{*D_1, D_2*}* is the set of the two dimensions currently displayed in MT,
- *Hier* = *{*H^{D_1}, H^{D_2}*}* is the set of the two hierarchies currently displayed belonging respectively to the two dimensions D_1, D_2 in MT,

— *Pred* = {$pred_1 \wedge$,..., $\wedge$ $pred_s$} is a normalized conjunction of predicates (restrictions of dimension data and fact data).

4.1 Drilldown null-option operator

The *Drilldown null-option* operator allows displaying information at a finer granularity level on a currently displayed dimension. After executing the *Drilldown null-option*, the decision-maker obtains a new multidimensional table with one dimension unchanged whereas the other dimension displays information at a finer granularity level. Our proposed analysis operator should facilitate decision-makers' tasks by not requiring the involved missing data.

4.1.1 Conceptual definition

*Drilldown*null-option(MT_k , D_i , P_{inf} , *Null-Option, [S]) = MT*	

Input	— MT_k: A multidimensional table currently displayed
	— D_i: One among the two analysis axes displayed in MT_k
	— P_{inf}: A parameter of low level than the lowest parameter displayed in the current hierarchy of D_i.
	— *Null-Option*: {All | $All_{NullLast}$ | Flexible}: Indicates how null-values of parameters Pinf will be treated by the Drilldown:
	  — *All*: is the default option, it means that the Drilldown returns all rows including those containing null values of parameter P_{inf}.
	  — $All_{NullLast}$: The Drilldown returns all rows including those containing null-values of parameter P_{inf}. It moves to the end of the result multidimensional table MT all rows containing null values.
	  — *Flexible*: If the percentage of null values returned for P_{inf} exceeds the threshold *S*, the operator changes the granularity level of P_{inf} in order to find a parameter *p* of lower level than P_{inf} having a percentage of null values less than *S*. A message will be posted to the user; it contains the percentage of null values for each parameter *p*. So, the user will be guided to select the adequate parameter *p* instead of P_{inf}.
	— *S*: Optional threshold to indicate the highest acceptable *percentage of null values (Percentage_Null)* in all cells in the result.

Output	MT is the resulting multidimensional table.

TAB. 1 – *Formalization of the Drilldown*null-option *operator.*

Example 1. In order to test and assess our proposed operator, we have extracted and loaded a data set containing 25508 tweets issued from different geographical places. Note that the place field does not have values in all tweets. For instance, assume that a decision-maker starts the analysis by displaying the total number of tweets according to the Country parameter of the *PLACE* dimension and *User-Activity* on the *USER* dimension. FIG 3 shows the result for this analysis. After executing the previous query, the decision-maker continues

Number of Tweets	**PLACE**				
	Country	Belgium	Canada	France	Spain
User-Activity					
New And Active		77	76	1266	56
New And Passive		131	13	1230	87
Old And Active		85	2	9177	103
Old And Passive			20	13119	66

TAB. 2 – *Multidimensional table* MT_0.

her/his analysis by displaying the number of tweets at a finer granularity level (*Region*) on one currently displayed dimension (*PLACE* dimension in our case) and (without changing the granularity level User-Activity of the USER dimension. The result is shown in FIG. 4 (MT_1). According to this analysis, many missing data are encountered. To deal with this issue, we propose an extension of the classic operator *Drilldown*. The decision-maker may receive 3 versions of multidimensional tables according to the specified *Null-option* for the *Drilldown*.

— The decision-maker chooses the option *All* in order to keep the analysis granularity to *Region* level: *Drilldown* ^{all} *(*MT_0*, PLACE, REGION)* = MT_1 (FIG. 4).

	PLACE									
	Country	France						Spain	Canada	Belgium
	Region									
User-Activity		centre	Ile-de-France	Languedoc Roussillon	Midi-Pyrénées	Picardie	Null	Null	Null	Null
New And Active		2	12	57	38	87	1075	56	76	77
New And Passive		1	24	1	1	2	1241	87	13	131
Old And Active		1	51	35	77	101	9002	103	2	85
Old And Passive		7	342	30	1	11	12907	66	20	

TAB. 3 – *Multidimensional table* MT_1.

— $All_{NullLast}$: The Drilldown returns all rows includ parameter P_{inf}. It moves at the end of the resulted multidimensional table all rows containing null values.

— If the decision-maker chooses the *Flexible* option, a message containing a list of parameters of lower level than P_{inf} with the percentage of the missing values of each one will be posted to the analyst. The chosen parameter *Region* is replaced with *City* which is the parameter having the minimum of missing values. The involved analysis expression presents as follows:*Drilldown*[*flexible*]*(*MT_0*, Place, City)* = MT_2. After the execution of this analysis operator, the decision-maker obtains the new MT presented in FIG. 5. We note that the analysis of the data according to the *City* parameter has improved the returned results since most of the missing values due to two parameters *Department* and *Region* are not included in the multidimensional table MT_2.

	PLACE									
	Country	France					Spain		Canada	Belgium
	City									
User-Activity		Cergy	Nanterre	Paris	Toulouse		Gérone	Ripoll	St. Catharines	Tournai
New And Active		91	80	670	524		56		76	77
New And Passive		110	159	567	109			87	13	131
Old And Active		86	922	4039	362		103		2	85
Old And Passive		98	121	6788				66	20	

TAB. 4 – *Multidimensional table* MT_2.

4.1.2 Logical definition

The logical definition of the *Drilldown*[*null-option*] operator is given as an algorithm described hereafter.

Algorithm 1: *Drilldown*[*null-option*]*(*MT_k , D_i , P_{inf} *, Null-Option, [S]) = MT* To clarify the algorithm, we need two functions defined as follows:

— *Length(*H^D*, D)*: returns the number of aggregation level in hierarchy H^D

— *Level (p,* H^D*, D)*: returns the level of parameter *p*, in hierarchy H^D of dimension *D* such as the finest parameter has level 1

Input

— MT_k: Multidimensional table

— *D* ∈ *{*D_1*,* D_2 *}* One of the two dimensions of MT_k

— P_{inf}: parameter of H^D , to be reached by Drilldown

— *Null-option*: Indicates how null-values of parameters P_{inf} will be treated by the Drill-down

— S: Optional threshold to indicate the highest acceptable percentage of null values *(Percentage_ Null_ Values* in the result.

Output: New multidimensional table MT, with the same structure as MT_k

Begin

1. Let H^D be the actually displayed hierarchy of *D*

2. Let Par = $\{p_n, p_{n-1}, \ldots, p_c\}$ be the set of displayed parameters of H^D with *c* is the level of the finest displayed parameter of H^D, and *n* is the level of the least fine parameter of H^D (i.e.,*n= Length*(H^D, *D*)), ($c \leqslant n$)
3. If *Level* (p_c, H^D, *D*) $\leqslant$ *Level* (P_{inf}, H^D, *D*) then
4. Impossible operation, the parameter P_{inf} is of lower granularity level than the specified parameter p_c displayed.
5. Else
6. Translate Drilldown (MT_k; D ; P_{inf}) into query Q
7. Q = " **Select** " $\|$ p_n, $p_{n-1}, \ldots,$ P_{inf} $\|$ $f_1(m_1)$, $f_2(m_2), \ldots$ $\|$ " **From** "$\|$ D_1, D_2, *F* $\|$ " **Where** " $\|$ MT_k.*Pred*, *Join Condition* $\|$ " **Group by** " $\|$ p_n, $p_{n-1}, \ldots,$ P_{inf} $\|$ " **Order by** " $\|$ P_{inf}
8. MT = Results of query Q.
9. Percentage_ Null_ Values = Number of cells containing null values of P_{inf} in MT $/$ Card(MT)
10. If Percentage_ Null_ Values > S then
11. If Null-option = "Flexible" then
12. For each parameter $p_j \in H^D$ ($1 \leqslant$ j < *Level* (P_{inf}, H^D, *D*))
13. ContinuerForage = True
14. While ContinuerForage
15. Drop table MT
16. Translate Drilldown (MT_k; D ; p_j) into query Q
17. Q_j = " **Select** "$\|p_n$, $p_{n-1}, \ldots,$ p_j $\|$ $f_1(m_1)$, $f_2(m_2), \ldots$ $\|$ " **From** "$\|$ D_1, D_2, *F* $\|$ " **Where** "$\|MT_k$.*Pred*, *Join Condition* $\|$ " **Group by** " $\|p_n$, $p_{n-1}, \ldots, p_j\|$" **Order by** "$\|p_j$
18. MT = Results of query Q_j
19. Percentage_ Null_ Values = Number of cells containing null values of p_j in MT $/$ Card(MT)
20. If Percentage_ Null_ Values < S then
21. Display table MT
22. ContinuerForage = False
23. End If
24. j = j+1
25. End While
26. Else
27. Drop table MT
28. Translate Drilldown (MT_k; D ; P_{inf}) into query Q
29. Q = " **Select** " $\|$ p_n, $p_{n-1}, \ldots,$ P_{inf} $\|$ $f_1(m_1)$, $f_2(m_2), \ldots$ $\|$ " **From** "$\|$ D_1, D_2, *F* $\|$ " **Where** " $\|$ MT_k.*Pred*, *Join Condition* $\|$ " **Group by** " $\|$ p_n, $p_{n-1}, \ldots,$ P_{inf} $\|$ " **Order by** " $\|$ P_{inf}

MT = Results of query Q

If null-option = "$All_{NullLast}$" then

Q = **" Select "** $\|$ p_n, p_{n-1},..., P_{inf} $\|$ $f_1(m_1)$, $f_2(m_2)$,... $\|$ **" From "**$\|$ D_1, D_2, *F* $\|$ **" Where "** $\|$ MT_k*.Pred, Join Condition* $\|$ **" Group by "** $\|$ p_n, p_{n-1},..., P_{inf} $\|$ **" Order by "** $\|$ P_{inf} $\|$ **" DESC NULLS LAST "**;

MT = Results of query Q

End if

Display MT

End For

End if

End if

End if

End

Result Now we will illustrate how Null-option analysis operators are transformed into SQL queries. The first multidimensional table MT_0 is obtained by executing the following SQL code.

```
SELECT      COUNT(A.id_ activity_ TW), U.user-Activity, P.country
FROM        FACTIVITY_ TWEET A, DUSER U, DPlace P
WHERE       A.id-U = U.id-U AND A.id-Pl = P.id-Pl
GROUP BY    U.user-Activity,P.country
ORDER BY    P.country
```

During the execution of the *Drilldown* *null-option* operators, three types of queries are generated according to the used option.

— If the decision-maker chooses the option *All* in order to keep the analysis granularity to *Region* level: *Drilldown* *all* *(*MT_0*, Place, Region)* = MT_1. The generated query is as follows:

```
SELECT      COUNT(A.id_ activity_ TW),U.user-Activity, P.country, P.Region
FROM        FACTIVITY_ TWEET A, DUSER U, DPlace P
WHERE       A.id-U = U.id-U and A.id-Pl = P.id-Pl
GROUP BY    U.user-Activity,P.country, P.Region
ORDER BY    P.Region
```

— The query corresponding to the option $All_{NullLast}$ is transformed to the SQL code below.

```
SELECT      COUNT(A.id_ activity_ TW),U.user-Activity, P.country, P.Region
FROM        FACTIVITY_ TWEET A, DUSER U, DPlace P
WHERE       A.id-U = U.id-U and A.id-Pl = P.id-Pl
GROUP BY    U.user-Activity,P.country, P.Region
ORDER BY    P.Region
DESC NULLS Last;
```

— If the decision-maker chooses the option *Flexible*, the chosen parameter *Region* is replaced by *City* which is the parameter having the minimum of the missing values. The analysis operator involved is presented as follows: *Drilldown*^{flexible} (MT_0, *Place, City*) = MT_2. The null-option analysis framework generates the query applicable to our R-OLAP model.

```
SELECT      COUNT(A.id_ activity_ TW),U.user-Activity, P.country, P.City
FROM        FACTIVITY_ TWEET A, DUSER U, DPlace P
WHERE       A.id-U = U.id-U and A.id-Pl = P.id-Pl
GROUP BY    U.user-Activity,P.country, P.City
ORDER BY    P.City
```

4.2 Rollup $^{null-option}$ operator

The *Rollup* $^{null-option}$ operator consists in moving from finer granularity data to coarser granularity data on a currently displayed dimension. Table 2 shows its algebraic formalization.

4.2.1 Conceptual definition

Example 2.Suppose that the decision-maker continues his analysis by rolling up. The analysis operator involved is presented as follows:

— *Rollup* ^{all} (MT_2, *Place, City*) = MT_1

— *Rollup* ^{flexible} (MT_1, *Place, Country*) = MT_0

4.2.2 Logical definition

The logical definition of the *Rollup*$^{null-option}$ operator is given by the following algorithm.
Algorithm 2: *Rollup*$^{null-option}$(MT_k , D_i , P_{sup} , *Null-Option, [S]) = MT* **Input**

— MT_k: Multidimensional table

— *D* ∈ *{*D_1, D_2 *}* One of the two dimensions of MT_k

— P_{sup}: parameter of H^D

— *Null-option*: Indicates how null-values of parameters P_{sup} will be treated by the Rollup

— S: Optional threshold to indicate the highest acceptable percentage of null values *(Percentage_ Null_ Values* in the result.

Output: New multidimensional table MT, with the same structure as MT_k
Begin

1. Let H^D be the actually displayed hierarchy of *D*
2. Let Par = $\{p_n, p_{n-1}, \ldots, p_c\}$ be the set of displayed parameters of H^D with *c* the most general graduation displayed parameter of H^D, and *n* is the level of the least coarse parameter of H^D (i.e.,*n*= *Length(*H^D*, D)), (c* $\geqslant$ *n)*
3. If *Level (*p_c*,* H^D*, D)* $\geqslant$ *Level (*P_{sup}*,* H^D*, D)* then
4. Impossible operation, the parameter P_{sup} is of of higher granularity level than the most general parameterp_c displayed.
5. Else

Rollup *null-option*$(MT_k$, D_i , P_{sup} , *Null-Option, [S]) = MT*	

Input	— MT_k: A multidimensional table currently displayed
	— D_i: One among the two analysis axes displayed in MT_k
	— P_{sup}: Chosen parameter on dimension D_i.
	— *Null-Option*: {<u>All</u> | $All_{NullLast}$ | Flexible}: Indicates how null-values of parameters Pinf will be treated by the Drilldown:
	  — *All*: The Rollup operator will return all the values corresponding to the chosen parameter including the null-values.
	  — $All_{NullLast}$: The Rollup operator will return all the values corresponding to the chosen parameter including the null-values. The operator will be accompanied with a classification of the values, by putting at the end of the multidimensional table the null values.
	  — *Flexible*: If the percentage of null values returned for P_{sup} exceeds the threshold *S*, the operator changes the granularity level of P_{sup} in order to find a parameter *p* of higher level than P_{sup} having a percentage of null values less than *S*. A message will be posted to the user; it contains the percentage of null values for each parameter *p*. So, the user will be guided to select the adequate parameter *p* instead of P_{sup}.
	— *S*: Optional threshold to indicate the highest acceptable *percentage of null values (Percentage_Null)* in all cells in the result.

Output	MT is the resulting multidimensional table.

TAB. 5 – *Formalization of the Rollup* *null-option* *operator.*

6. Translate Rollup (MT_k; D ; P_{sup}) into query Q
7. Q = **" Select "** ‖ p_n, p_{n-1},..., P_{sup} ‖ $f_1(m_1)$, $f_2(m_2)$,... ‖ **" From "**‖ D_1, D_2, *F* ‖ **" Where "** ‖ MT_k*.Pred, Join Condition* ‖ **" Group by "** ‖ p_n, p_{n-1},..., P_{sup} ‖ **" Order by "** ‖ P_{sup}
8. MT = Results of query Q.
9. Percentage_ Null_ Values = Number of cells containing null values of P_{sup} in MT ⁄ Card(MT)
10. If Percentage_ Null_ Values > S then
11. If Null-option = "Flexible" then
12. For each parameter $p_j \in H^D$ (*Level* (P_{sup}, H^D, *D)* < j $\leqslant$ *Level* (p_c, H^D, *D)*)
13. ContinuerForage = True
14. While ContinuerForage
15. Drop table MT
16. Translate Rollup (MT_k; D ; p_j) into query Q

Q_j = **" Select "**$\|p_n, p_{n-1},\ldots, p_j \| f_1(m_1), f_2(m_2),\ldots\|$ **" From "**$\| D_1, D_2, F \|$ **" Where "**$\|MT_k$*.Pred*, *Join Condition* $\|$ **" Group by "** $\|p_n, p_{n-1},\ldots,p_j\|$**" Order by "**$\|p_j$

MT = Results of query Q_j

Percentage_ Null_ Values = Number of cells containing null values of p_j in MT ／ Card(MT)

If Percentage_ Null_ Values < S then

Display table MT

ContinuerForage = False

End If

j = j+1

End While

Else

Drop table MT

Translate Rollup (MT_k; D ; P_{sup}) into query Q

Q = **" Select "** $\| p_n, p_{n-1},\ldots, P_{sup} \| f_1(m_1), f_2(m_2),\ldots \|$ **" From "**$\| D_1, D_2, F \|$ **" Where "** $\| MT_k$*.Pred*, *Join Condition* $\|$ **" Group by "** $\| p_n, p_{n-1},\ldots, P_{sup} \|$ **" Order by "** $\| P_{sup}$

MT = Results of query Q

If null-option = "$All_{NullLast}$" then

Q = **" Select "** $\| p_n, p_{n-1},\ldots, P_{sup} \| f_1(m_1), f_2(m_2),\ldots \|$ **" From "**$\| D_1, D_2, F \|$ **" Where "** $\| MT_k$*.Pred*, *Join Condition* $\|$ **" Group by "** $\| p_n, p_{n-1},\ldots, P_{sup} \|$ **" Order by "** $\| P_{sup} \|$ **" DESC NULLS LAST "**;

MT = Results of query Q

End if

Display MT

End For

End if

End if

End if

End

Facing large volumes of data among which a great amount of missing data are found, our aim is to both increase the efficiency of analysis and facilitate the analysts task. To this end, we have proposed extensions of classical drilldown and rollup operators.

5 Conclusion

Data analysis in social networks is often hampered by missing data. For this reason, we have proposed an extended version for each of the two OLAP operators Drilldown and Rollup. We call the extended versions $Drilldown^{null\text{-}option}$ and $Rollup^{null\text{-}option}$, in order to support a way to process OLAP queries on data sets having missing data. For each of these OLAP operator, we have presented an algebraic formalization and a logical definition as a pseudo code algorithm. Then we have given illustrative examples showing results given when Null-option analysis is used. To the best of our knowledge, this is the first discussion about how OLAP analysis operators can be carried out in the case of missing data in multidimensional modeling. As perspective work, we intend to integrate more analysis operators that take into consideration the specificities of our multidimensional model, as Reflexive Fact and dynamic Data. These operators will help the interpretation of the results of multidimensional analyses on tweets and their metadata. It is also important to note that social networks data entries (e.g., user profile data, message status) evolve over time and therefore the occurring changes must be considering in the corresponding analysis. For this reason; it would be interesting to define an approach enabling OLAP to keep up with volatile data using the concepts of slowly changing dimensions to enable analysis of both the recent state of data and any of its previous states. Moreover, we plan to conduct experiments to measure the quality of the result extracted by our OLAP operators. Finally, the scalability of our approach merit to be proved.

References

Adar, E. and C. Ré (2007). *Managing Uncertainty in Social Networks*. In ieee computer society technical committee on data engineering, 15-22.

Ben Kraiem, M., J. Feki, K. Khrouf, F. Ravat, and O. Teste (2014). Olap of the tweets from modeling toward exploitation. *In 8th International Conference on Research Challenges in Information Science (IEEE RCIS'2014)*, 45–55.

Ben Kraiem, M., J. Feki, K. Khrouf, F. Ravat, and O. Teste (2015a). Modeling and olaping social media: the case of twitter. *In Social Netw. Analys. Mining 5(1)*, 47:1–47:15.

Ben Kraiem, M., J. Feki, K. Khrouf, F. Ravat, and O. Teste (2015b). Olap4tweets: Multidimensional modeling of tweets. *In European Conference on Advances in Databases and Information Systems, ADBIS*, 68–75.

Collins.L, C., B. Twala, and T. Marwala (2014). Missing data prediction and classification: The use of auto-associative neural networks and optimization algorithms. *In Computer Science: Neural and Evolutionary Computing*.

Gyssens, M. and L. V. S. Lakshmanan (1997). A foundation for multi-dimensional databases. *In Proceedings of the 23rd International Conference on Very Large Data Bases Athens, Greece*, 106–115.

Hess, J. (1998). Dealing with missing values in the data warehouse. *In A Report of Stonebridge Technologies, Inc*.

Huisman, M. (2009). Imputation of missing network data: Some simple procedures. *In Journal of Social Structure, Vol.10, No.1*.

McClean.S.I, B. W. Scotney, and M. Shapcott (2001). Aggregation of imprecise and uncertain information in databases. *In IEEE Transactions on Knowledge and Data Engineering TKDE*, 902–912.

Ravat, F., O. Teste, R. Tournier, and G. Zurfluh (2007). Graphical querying of multidimensional databases. advances in databases and information systems. *Advances in Databases and Information Systems Vol. 4690, Berlin, Heidelberg: Springer Berlin Heidelberg*, 298–313.

Ravat, F., O. Teste, R. Tournier, and G. Zurfluh (2008). Algebraic and graphic languages for olap manipulations. *In Algebraic and graphic languages for olap manipulations*, 17–46.

Sadikov, E. M. M., J. Leskove, and H. Garcia-Molina (2011). Correcting for missing data in information cascades. *In International Conference on Web Search and Data Mining, WSDM'11, February 9-12, 2011, Hong Kong, China*.

Résumé

L'analyse des données issues des réseaux sociaux est souvent entravée par le problème d'absence de données. Des études récentes montrent les effets négatifs des données manquantes (ou valeurs nulles). Les résultats de l'analyse des données des réseaux sociaux peuvent être gravement erronés si les analyses se limitent aux attributs renseignés et ignorent les valeurs nulles. Pour surmonter ce problème de données manquantes, plusieurs méthodes ont été proposées dans la littérature. Dans cet article, nous proposons des extensions d'opérateurs classiques de *Drilldown* et *Rollup* pour permettre des analyses en présence de données manquantes dans les membres de dimensions.