A Tensor-Based Distributed Discovery of Missing Association Rules On The Cloud

I.Elayyadi[a], S. Benbernou[a], M. Ouiziri[a], M. Younas[b]

[a] *Université Paris Descartes, Sorbonne Paris Cité, France*
[b] *Department of Computing and Communication Technologies, Oxford Brookes University, Oxford, UK*

Abstract

An increasing number of data applications such as monitoring weather data, data streaming, data web logs, and cloud data, are going online and are playing vital in our every-day life. The underlying data of such applications change very frequently, especially in the cloud environment. Many interesting events can be detected by discovering such data from different distributed sources and analyzing it for specific purposes (e.g., car accident detection or market analysis). However, several isolated events could be erroneous due to the fact that important data sets are either discarded or improperly analysed as they contain missing data. Such events therefore need to be monitored globally and be detected jointly in order to understand their patterns and correlated relationships. In the context of current cloud computing infrastructure, no solutions exist for enabling the correlations between multi-source events in the presence of missing data. This paper addresses the problem of capturing the underlying latent structure of the data with missing entries based on association rules.This necessitate to factorize the data set with missing data.

The paper proposes a novel model to handle high amount of data in cloud environment. It is a model of aggregated data that are confidences of associations rules. We first propose a method to discover the association rules locally on each node of a cloud in presence of missing rules. Afterward, we provide a tensor based model to perform a global correlation between all the local models of each node of the network.

The proposed approach based on tensor decomposition, deals with a multi modal network where missing association rules are detected and their confidences are approximated. The approach is scalable in terms of factorizing multi-way arrays (i.e. tensor) in the presence of missing association rules. It is validated through experimental results which show its significance and viability in terms of detecting missing rules.

Keywords: Distributed mining, association rules, cloud computing, tensor.

1. Introduction

Today every organization is facing the issue of handling a potentially large volume of data that come from multiple and distributed data sources. Data applications such as weather data, data streaming, sensor data, web logs and publish/subscribe applications, produce and consume large volume data. The underlying data of such applications is very dynamic and changes very frequently. Many interesting events can be detected by discovering such data from different distributed sources and analyzing it for specific purposes[1,2,3]. For instance, in the vehicle industry, the future driver assistance systems will need to discover, collect and analyze dynamic information about cars environment and the drivers state. It will need to collect information from various different sources which are distributed across different locations. For instance, real time information from road sensors, radars, GPS, video and eye-tracking systems need to be instantly discovered, collected and analyzed. In addition, such systems require that the correlation of information from different sources is necessary in order to get a consistent view of the real world and help drivers in making appropriate decisions and taking appropriate actions. However, the appropriate discovery, collection and analysis of such data are affected by various factors. For instance, reliable analysis of data collected can be affected by missing data. The loss of information and errors in the data collection process are the

two main contributing factors to the missing data. The consequence of erroneous and missing data is that some important data sets may be discarded or improperly analyzed giving incorrect information. Further, several isolated events may also have to be monitored globally and jointly detected in order to understand their patterns and correlation relationships, leading to adapt the system behavior and take appropriate actions considering a particular conjunction of events. Moreover, in a large network of computers or sensors, each of the components has some data about the global state of the system and much of the systems functionality relies on modeling the global state of the system which is constantly changing. It is necessary to keep the models up-to-date and seek for incomplete information. Computing global data mining models e.g. decision trees, k-means clustering in large distributed systems may be very costly due to the scale of the system and due to communication cost, which may be high. The cost further increases in a dynamic scenario when the data changes rapidly[1].

The aforementioned issues will be studied in this paper in a new type of distributed environment i.e. the cloud computing[4]by handling specific data that are the "association rules"[5]. Cloud computing is the most hyped trends for the last few years. However, to our knowledge, currently there exist no solution that enables the correlations between multi-source events in the cloud computing.

The paper proposes a novel model to handle high amount of data in cloud environment. The proposed model takes into account aggregated data that are confidences of associations rules. This paper addresses the issue of discovering and predicting the missing association rules from incomplete data on a cloud node, and by correlating them with data coming from other nodes. The proposed approach is distributed and is based on tensor decomposition[6]. The decompositions are applied to data arrays for extracting and explaining their properties. The proposed model deals with a multi modal network where missing association rules are detected and their confidences are approximated. For that, the association rules i.e. their confidences will be represented as arrays in each node, where the obtained arrays are incomplete and the results of correlation between the association rules with other nodes are represented by a tensor. In other words, our goal at first attempt is to capture the latent structure of the data via higher-order factorization in the presence of missing association rules. The second attempt is to recover the missing entries toward distributed correlation of association rules over the cloud network.

The paper proposes a novel approach in order to discover the association rules locally on each node of a cloud and globally correlates the local results (over the cloud) to predict missing association rules.

Our salient contributions in this paper are:

- *Global correlation model.* To handle and analyze efficiently the high amount of data on the cloud network, we propose an aggregation data model related to confidences of association rules in presence of missing data.
- *A scalable algorithm.* We developed a scalable algorithms for tensor factorization to correlate the association rules in presence of missing rules over the cloud network and recover these missing entries.
- *Experiments.* To validate the obtained results, the distributed approach is evaluated with numerical experiments on simulated data sets in presence of incomplete and missing data.

The remainder of the paper is organized as follows: section 2 presents a set of definitions and background that are used in the design of the proposed approach. Section 3 gives an overall picture of the proposed approach. Section 4 describes the local mining step with data representation and the applied algorithm. Section 5 presents the second step of distributed mining based on tensor concept. Section 6 gives experimental results of the approach. Finally, we provide and outlook on future work and conclusion of the paper in section 7.

2. Background

Notation

In data mining, association rule is a popular and well research method for discovering interesting relations between variables in large databases[7]. Agrawal introduced associations for discovering regularities between products in large scale transaction data recorded in supermarkets.The deduced information can be helpful for decisions about marketing activities

This section describes some basic definitions and concepts which are used in the design of the proposed model.

Table 1: Notation Table

i_k	:	The k^{eme} item
I	:	Itemset
T	:	A set of transaction
$Freq(I)$	:	Frequency
R	:	Association rule
$conf(R)$	:	confidence
$(\mathcal{X}; \mathcal{R})$	:	Tensors of order $N \geq 3$
$(\mathbf{A}; \mathbf{B}; \mathbf{C})$	:	Matrices
$(\boldsymbol{a}; \boldsymbol{b}; \boldsymbol{c})$	:	Vectors
$\mathcal{N}_i$	:	The i^{eme} node

2.1. Itemsets and association rules

2.1.1. Definitions

Following Agrawal'definition, the problem of association rule mining is defined based on itemsets.

Definition 1 (Itemset). Let $I = \{i_1; i_2 \cdots; i_k\}$ be a set of k binary attributes called *items* and let $T = \{t_1; t_2 \cdots; t_n\}$ be a set of transactions, an item i_l is an attribute and $I \subset T$.

An itemset is characterized by the following concepts:

- *Support.* A support of an itemset I denotes $supp(I)$ is defined as the proportion of transactions in the data set which contains the itemset and is equal to the number of object containers.
- *Frequency.* The frequency of an itemset I is the probability that I occurs in set of transactions T, which is denoted by $Freq(I)$ and is equal to $\frac{supp(I)}{card(T)}$ where $card(T)$ means the total number of transactions in T.

It is known that itemset is frequent if its support is greater than or equal to a minimum threshold.

Theorem 1. *All itemsets form an ideal order in* $(2M, \subseteq)$ *(compared to the frequency constraint)*

We deduce that any subset of a frequent itemset is frequent, and any superset of an infrequent itemset is infrequent.

Definition 2 (Association rule). An association rule is expressed as: $R : X \longrightarrow Y$,with $X \in T$, $Y \in T$ and $X \cap Y = \emptyset$. The concepts related to a rule are:

- *Support* The support of a rule is expressed by the amount of objects in T containing $X \cup Y$, $(supp(R) = P(X \cup Y))$ We measure the strength of an association rule by the confidence which is equal to the proportion of transactions containing X that also contain Y,
- *Confidence.* $conf(R) = \frac{P(X \cup Y)}{P(X)}$.

  Two types of rules emerge from the confidence measure: Exact rule if $Conf(R) = 1$ and rules of thumb if $Conf(R) < 1$
- The "$lift$" rule measures the improvement provided by the association rule in relation to a set of random transactions (where X and Y are independent). It is defined by $\frac{P(X \cup Y)}{P(X)P(Y)}$. A "lift" greater than 1 indicates a positive correlation between X and Y, and thus the significance of the association.

2.2. Algorithms

A set of algorithms have been provided to discover the associations rules. Among them one can cite the Apriori algorithm developped by Agrawall[8]which uses a bottom-up method in which, at each stage, subsets are expanded to a common item. After the Apriori was proposed, many new algorithms or improvements to existing algorithms have been published. But finding all the frequent itemsets remains a difficult task because the search space is exponential function the number of items in the database. Among the most significant changes to Apriori that have been proposed include[9] and Toivonen sampling algorithm[10] . The first algorithm to generate all candidates by a depth-first approach (Approach type Depth-First), Eclat, was published in 2000[11]. In this paper the local data correlation will be based on Apriori algorithm. Details of the algorithm will be presented in section 4

2.3. tensor

The notion of tensor science is not new. But in recent decades, their use have greatly developed in psychometrics[12] chemometrics[13] . In these areas, data can depend on several factors. It may be, for example, to measure different individuals in different situations. Tensor decomposition is used to

analyze the data according to all modes, to summarize with few components and describe their interactions. The methods for decomposing a tensor allow it to capture the underlying information that could not find a matrix analysis[6,14]. Tensors can also be used in social networks network analysis or Internet (web mining)[15]. The data can represent several types of similarities or several types of relationships between nodes in a graph. They can also take into account the evolution time of a relationship between nodes.

Definition 3 (Tensor). A tensor is a multidimensional array. The order of a tensor is the number of dimensions (or modes). More formally, an N-way or Nth-order tensor is an element of the tensor product of Nvector spaces, each of which has its own coordinate system.

Notations

Tensors of order $N \geq 3$ are denoted by Euler script letters $(\mathcal{X}; \mathcal{R})$, matrices are denoted by boldface capital letters $(\mathbf{A}; \mathbf{B}; \mathbf{C})$, vectors are denoted by boldface lowercase letters $(\boldsymbol{a}; \boldsymbol{b}; \boldsymbol{c})$, and scalars are denoted by lowercase letters (a, b, c). Columns of a matrix are denoted by boldface lower letters with a subscript $(\boldsymbol{a}_1; \boldsymbol{a}_2; \boldsymbol{a}_3)$ are first three columns of $\mathbf{A}$. Entries of a matrix or a tensor are denoted by lowercase letters with subscripts, i.e., the $(i_1; i_2; \ldots; i_N)$ entry of an N-way tensor $\mathcal{X}$ is denoted by $x_{i_1;i_2;\ldots;i_N}$.

Example 1. A third-order tensor has three indexes as shown in Figure 1. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors.

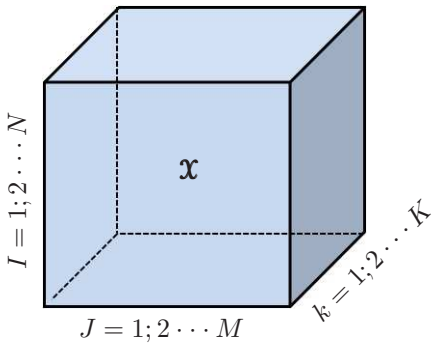

Figure 1: Tensor $\mathcal{X} \in \mathbb{R}^{N \times M \times K}$

Mathematically, a tensor is defined more rigorous as an intrinsic part of multi-linear algebra.

This notion of tensors is not to be confused with tensors in physics and engineering (such as stress tensors)[6], which are generally referred to as tensor fields in mathematics.

Definition 4 (Tensor Slice/Fiber). The tensor is referenced with tensor slice and fibre:

- A tensor slice is a two-dimensional fragment of a tensor, obtained by fixing all indices but two. For example, the horizontal, lateral, and frontal slides of a third-order tensor $\mathcal{X}$ are denoted by $\mathcal{X}_{i::}$, $\mathcal{X}_{:j:}$, and $\mathcal{X}_{::k}$, respectively.
- A tensor fiber is a one-dimensional fragment of a tensor, obtained by fixing all indices but one. Tensor fibers are the higher-order analogue of matrix rows and columns. Third-order tensors have column, row, and tube fibers, denoted by $\mathcal{X}_{:jk}$, $\mathcal{X}_{i:k}$, and $\mathcal{X}_{ij:}$, respectively. A tensor coupe is depicted in Figure 2.

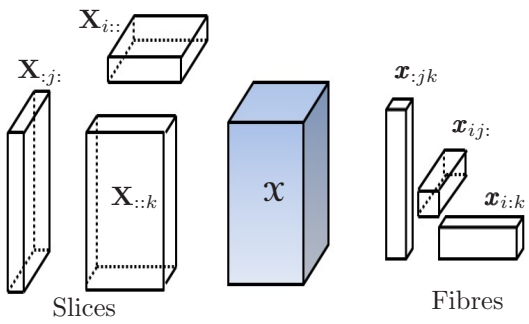

Figure 2: Tensor Coupe

3. Overall picture of the distributed data correlation framework on the cloud

This section presents the proposed framework. The objective of the framework is to predict association rules in the cloud computing environment. In cloud, data is distributed across different nodes (or computer systems) which are connected through networks such as Internet. The framework deals with the situation where data, discovered and retrieved from different nodes, is incomplete and may prone to errors.

There exist various approaches that deal with issues related to incomplete and erroneous data in

classical databases such as discovery of association rules discovery and prediction of missing data[16]. However, applying traditional approaches to cloud is problematic for the following reasons:

- *Centralization:* Data in traditional approaches is generally centralized. Thus it is relatively straightforward to discover and predict data when data is centrally stored at one location. However, this is not the case with cloud computing in which data is stored at different locations and in different systems. This significantly complicates the process of discovery and prediction of data.
- *Missing data:* Traditional algorithms designed for discovering associations rules do not deal with missing data. However, given the nature of cloud computing it is more likely that some data (that need to be discovered and analysed) might be missing. This will also result in creating situation wherein the resulted association rules could be incomplete.

The proposed approach is:

- Distributed: the approach does not collect all data from nodes on a centralized data warehouse. It performs local discovering of association rules on each node.
- Discovering missing association rules: discovering association rules on incomplete data of a given node leads to missing association rules. The proposed approach makes correlation of all the local association rules to predict missing one in each node.

The above issues provide the rationale for the design and development of a new framework that deals with the discovery of distribution of data across different nodes of the cloud as well as tackle the issues created by the missing data and incomplete association rules. In this paper we design and develop a framework that deals with these issues. The fundamental principles of the proposed framework are described as follows:

1. Distributed data: The framework discovers and collects data from different nodes that are distributed across the cloud. Unlike traditional approaches, it does not collect all data from nodes on a centralized data warehouse. It performs local discovery of association rules on each node. This then feeds into the discovery of missing association rules at the global level (or cloud level).
2. Discovering missing association rules: Discovering association rules on incomplete data of a given node leads to missing association rules. The proposed approach makes correlation of all the local association rules to predict missing ones in each node.

Based on the above the proposed framework is designed by following the following major stages, which are also depicted in Figure3.

1. *Local Data Correlation*: In this stage, the local data correlation process discovers local association rules on each node. This task is done separately on each node. It therefore does not require centralizing data of all nodes. The discovered association rules and their respective confidences are represented in a matrix, called local confidence matrix. A local confidence matrix is computed for each node. The discovery of association rules is done by applying the A Priori algorithm locally on each node. Further details are provided in section 4.
2. *Global Data Correlation:* In this stage, the resulted local association rules are correlated in order to discover missing local association rules. Discovering missing association rules is performed for each node in the following two steps:
   (a) Identifying missing association rules: In this step, the missing association rules are identified by correlating the local discovering results obtained from all the nodes of the cloud. Our intuition is that data of different nodes of the cloud are related (for a particular application, e.g., market analysis). Given this, the missing association rules at a given node N_i are all the local association rules discovered on either all or at least one other node. Thus if A_k denotes the set of local association rules, discovered on node N_k, then the missing association rules on N_k is the set $M_k = (\bigcup_{j=1:R, j\neq k} A_j) - A_k$.
   (b) Discovering missing association rules: This step discovers the missing association rules. At the global correlation, the confidence of missing association rules M_k is unknown in the confidence matrix of

the node N_k. The global data correlation aims to predict the unknown confidences of missing association rules for each node. This prediction is made by correlating results of local correlations, namely the computed confidence matrices. Note that the global correlation is not a centralized process. Global correlation does not aggregate data of nodes but only their summaries, which consist of confidence matrices of locally discovered association rules. The confidence matrices are aggregated into a tensor model and the prediction of missing confidences is done using the conjugate gradient algorithm. The details are provided in the section 5.

4. Local Data Correlation

In this step, data are correlated locally at each node for discovering association rules.

The association rules discovering consists of identifying the frequent itemsets and then, forming conditional implication rules among them.

On a node N_i, data is set of transactions, also called transactional basis.

The classic problem of discovering association rules can be described as follows: let $I = \{i_1; i_2 \cdots i_k\}$ a $k - items$. Let $\mathcal{D} = \{t_1; \ldots; t_n\}$ be a set of transactions stored at the node N_i of the cloud. Each transaction t_j is a set of items, uniquely identified by an identifier tid.

The extraction is done in two steps:

1. Step 1: frequent itemsets $I_1; \ldots; I_m$ are extracted using the Apriori algorithm (see Algorithm 1).
2. Step 2: association rules are generated from the frequent itemsets $I_1; \ldots; I_m$. The association rules $X \rightarrow Y$ is generated from the frequent itemset I_k if:

$$X = \{i_1; \ldots; i_l\} \subset I_j \text{ and } Y = I_k - X, \text{ and}$$
$$conf(X \rightarrow Y) \geq minconf$$

Let us consider a database on the node N_i depicted in Table 2,

The extracted frequent itemsets using the Apriori algorithm are presented in Table 3.

From the itemsets of Table 3, association rules can be generated only from the frequent itemset $\{ac\}$ and given in Table 4.

input : $\mathcal{D}$, $minsup$
output: $\bigcup_k L_k$

L_1 : { *set of 1-item that appear in at least three transactions*};
$k = 2$;
while $L_{k-1} \neq \emptyset$ **do**
  $C_k = generate(L_{k-1})$ /Generates the candidate set/ ;
  for $t \in T$ **do**
    $C_t = \text{subset}(C_k, t)$ /Selection of candidates C_k present in t/;
    for $c \in C_t$ **do**
      $count[c] = count[c] + 1$
    end
  end
  $L_k = \{c \in C_k / count[c] \geq minsup\}$;
  $k = k + 1$
end

Algorithm 1: The Apriori algorithm

Table 2: Example of a database on node $\mathcal{N}_i$

$\mathcal{N}_i$	
tid	transaction
1	*a b c*
2	*a c*
3	*a d*
4	*b e f*
5	*a b c f*

Table 3: Frequent itemsets of the database on N_i with $minsupp = 50\%$

$\mathcal{N}_i$	
Frequent itemsets	support
$\{a\}$	80%
$\{b\}$	60%
$\{c\}$	60%
$\{ac\}$	60%

5. Global Data Correlation

In this section we present the tensor-based approach for the approximation of missing confidences. The approximation problem is the minimization of tensor-based cost function after decomposition. We first present the model and discuss the approximation method and the proposed algo-

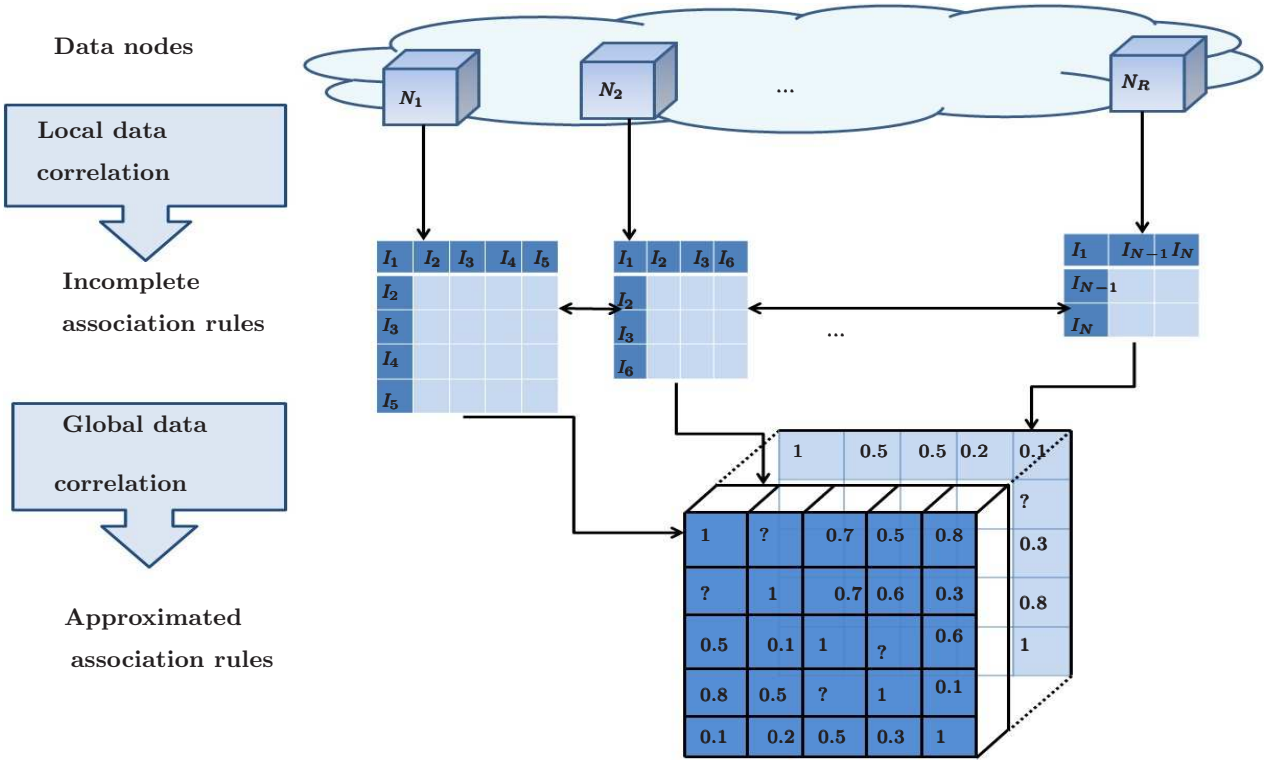

Figure 3: Framework of distributed data correlation on the cloud

Table 4: Association rules of the database on N_i with $minconf = 70\%$

$\mathcal{N}_i$	
Association rules	confidence
$\{a \rightarrow c\}$	$O.75$
$\{c \rightarrow a\}$	1

rithm.

5.1. The tensor-based model for global correlation

The correlation model is represented by the *confidence tensor*.

Definition 5 (Confidence tensor). Confidence tensor $\mathfrak{R}$ is defined by the uplet $(R, G, \mathcal{W})$ where:

- R is a network of nodes: $N_1, N_2 ... N_R$.
- G_i is the set of frequent itemsets in the node N_i $G = \bigcup_{i=1:R} G_i$.
- $\mathcal{R}$ is a tensor defined on the space $\mathbb{R}^{N \times N \times R}$ with values in $[0, 1]$, the fiber $\boldsymbol{r}_{ij:}$ represents all confidence related to association rules between itemsets i and j in all nodes.
- $\mathcal{W}_{ij}$ is a weight matrix, defined by:

$$\mathcal{W}_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{r}_{ijk} \text{ is known} \\ 0 & \text{if } \boldsymbol{r}_{ijk} \text{ is missing} \end{cases}$$

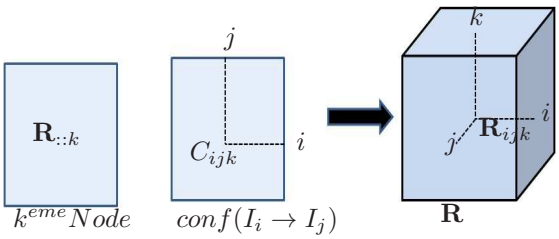

Figure 4: Confidence Tensor

The task of predicting relational patterns i.e., the missing association rule confidences will be to try to find all relational patterns for all zero values of the matrix $\mathcal{W}$. For this, we find a decomposition **CP** as close as possible to the original tensor $\mathcal{R}$ for relations known.

In order to perform this task, the following properties are defined.

Definition 6 (Properties on tensor). The properties comprise a set of definitions:

- the scalar product of two N way tensors $\mathcal{X}, \mathcal{R}$ is :
$$\mathcal{X} \cdot \mathcal{R} = \sum_{ijk} x_{ijk} x_{ijk}$$
- *Norm of tensor.* The norm of an N-way tensor $\mathcal{X} \in R^{I_1 \times I_2 \ldots \times I_N}$ is
$$\|\mathcal{X}\|^2 = \mathcal{X} \cdot \mathcal{X} = \sum_{ijk} x_{ijk}^2$$
- *Mode-n matrix product.* The mode matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$ and is of size$I_1 \times \ldots \times I_{n1} \times J \times I_{n+1} \times \ldots \times I_N$. Element-wise, we have :
$$\mathcal{X} \times_n \mathbf{U}_{i_1 \ldots i_{n1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}.$$
Multiple mode-n matrix products can be performed in any order $(\mathcal{X} \times_n \mathbf{U}) \times_m \mathbf{B} = (\mathcal{X} \times_m \mathbf{B}) \times_n \mathbf{A}$
- *exterior product.*The exterior product of three vector $u_1 \in \mathbb{R}^{I_1}$, $u_2 \in \mathbb{R}^{I_2}$ and $u_3 \in \mathbb{R}^{I_3}$ denoted by $u_1 \circ u_2 \circ u_3$ is a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as :
$$\forall (i;j;k) \in \mathbb{R}^{I_1 \times I_2 \times I_3} : \mathcal{X}_{ijk} = u_{1i} \times u_{2j} \times u_{3k}$$

5.2. Approximating confidence of missing association rules

This problem of approximating the confidences of association rules in presence of missing rules between nodes can be rewritten as a problem of approximating $\mathcal{R}$ by finding a tensor $\mathcal{X}$ of the same size that minimizes a cost of $\mathcal{R}$ defined as follows:

$$\mathtt{L}(\mathcal{X}) = \sum_{i,j=1}^{N} \mathcal{W}_{ij} \|\mathcal{R}_{ij:} - \mathcal{X}_{ij:}\|^2$$

The approximated tenor $\mathcal{X}$ is calculated by minimization $\mathtt{L}(\mathcal{X})$ using an iterative algorithm. For this aim, we use the conjugate gradient algorithm depicted is Algorithm 2.

Before using the gradient algorithm, the tensor $\mathcal{X}$ has to be decomposed into vectors or matrices. We use the CANDECOMP/PARAFAC decomposition method (CP)[6] depicted as follows:

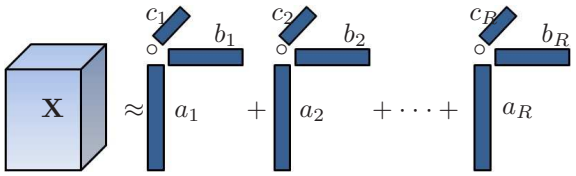

Figure 5: CANDECOMP/PARAFAC tensor decomposition

Then, the tensor $\mathcal{X}$ is decomposed as follows:

$$\mathcal{X} \approx \sum_{k=1}^{K} \boldsymbol{x}_k^{(1)} \circ \ldots \circ \boldsymbol{x}_k^{(N)}$$

We denote for the study of our model CP decomposition given by the three matrices 5.2 : $\mathcal{X} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$

By setting $\mathcal{X} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$, we obtain:

$$\mathtt{L}(\mathcal{X}) = \sum_{i,j=1}^{N} \sum_{r=1}^{R} \mathcal{W}_{ij} (\mathcal{R}_{ijr} - \sum_{k=1}^{K} A_{ik} B_{jk} C_{rk})^2$$

input : x_0 ;
We set $d_0 = -\nabla f(x_0)$
output: x minimum of f
while *Not convergence* **do**
  $x_{k+1} = x_k + \rho_k d_k$ with ρ_k optimal ;
  $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$ with :
  $\beta_k = \dfrac{\| \nabla f(x_{k+1}) \|^2}{\| \nabla f(x_k) \|^2}$;
end

Algorithm 2: The gradient algorithm

To establish the norm of the gradient of the form $\mathtt{L}(\mathcal{X})$ we determine the partial derivatives for the three directions X = A, Y = B and Z = C:

$$\frac{\partial \mathtt{L}}{\partial X_{ik}} = \sum_{j=1}^{N}\sum_{r=1}^{R} \mathcal{W}_{ij}(\sum_{k=1}^{K} X_{ik}Y_{jk}Z_{rk})Y_{jk}Z_{rk} - \sum_{j=1}^{N}\sum_{r=1}^{R} \mathcal{W}_{ij}\mathcal{R}_{i,j,r}Y_{jk}Z_{rk}$$

$$\frac{\partial \mathtt{L}}{\partial Y_{jk}} = \sum_{i=1}^{N}\sum_{r=1}^{R} \mathcal{W}_{ij}(\sum_{k=1}^{K} X_{ik}Y_{jk}Z_{rk})X_{ik}Z_{rk} - \sum_{i=1}^{N}\sum_{r=1}^{R} \mathcal{W}_{ij}\mathcal{R}_{i,j,r}X_{ik}Z_{rk}$$

$$\frac{\partial \mathtt{L}}{\partial Z_{rk}} = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathcal{W}_{ij}(\sum_{k=1}^{K} X_{ik}Y_{jk}Z_{rk})X_{ik}Y_{jk} - \sum_{i=1}^{N}\sum_{j=1}^{N} \mathcal{W}_{ij}\mathcal{R}_{i,j,r}X_{ik}Y_{jk}$$

5.3. Distributed data correlation algorithm

The entire approach combining the local and global correlations is processed by the distributed algorithm depicted in Algorithm 3.

The algorithm follows three steps:

1. First, the algorithm applies Apriori algorithm on each node of the cloud to calculate the association rules in presence of missing data. The obtained results are a set of confidence matrices. The combination of them will provide a confidence tensor $\mathcal{R}$ in presence of missing association rules. Confidence of missing association rules is unknown in the tensor.
2. Second, the algorithm performs the decomposition of the confidence tensor using the CP decomposition.
3. Finally, the algorithm approximates the unknown confidences in the tensor by means of the conjugated gradient.

input : $\{\mathcal{D}_1; \mathcal{D}_2; \cdots \mathcal{D}_R\}$ A set of database in all nodes;
ϵ; *minsup*
output: $\mathcal{X}$: A confidence tensor approaching missing value in $\mathcal{R}$

for $r \leftarrow 1$ **to** R **do**
  Apriori($\mathcal{D}_i$;*minsup*)
  for $i \leftarrow 1$ **to** N **do**
    for $j \leftarrow 1$ **to** N **do**
      $\mathcal{R}_{ijk} = conf(I_i \rightarrow I_j)$ in node r
    end
  end
end
for $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** N **do**
    for $r \leftarrow 1$ **to** R **do**
      $[[\mathbf{A}; \mathbf{B}; \mathbf{C}]] = CP(\mathcal{R})$
      $\mathcal{X} = GC(L_W; \epsilon)$
    end
  end
end

Algorithm 3: Distributed data correlation algorithm

6. Experimental results

In this section we simulated three-way data in order to assess the performance of the proposed algorithm in terms of its ability to find the underlying factors in the presence of missing confidences of association rules.

For simulation purpose, we generate a database D. For each node it has removed a random part of the database. The resulting databases are: $\{D_1 \cdots D_{10}\}$. The Apriori algorithm was applied to each node $\{N_i | i = 1 \cdots 10\}$,The confidence matrix were generated in all nodes. Table 5 gives details on the statistics of our database.

Table 5: Statistics application

Nodes	10
Total number of frequent itemset	55
Total number of missing confidence	2360
Confidence tensor $\mathcal{X} \in \mathbb{R}^{55\times55\times10}$	30250

The number of frequent itemsets for missing 10 nodes is given in Figure 6.

Several ranks of the tensor have been tested numerically. The weight matrix was filled with a focus on how we want to determine the missing association rules. We defined the relative error in the node N_i by $max(\frac{\text{experimental confidence}- \text{ theorical confidence}}{\text{theorical confidence}})$, the relative errors are shown in the following figure 6. We note that the error does not depend on the number of unknown association rules in the node i.

From this experiment 6, it was verified that the approximation error depends on the number of

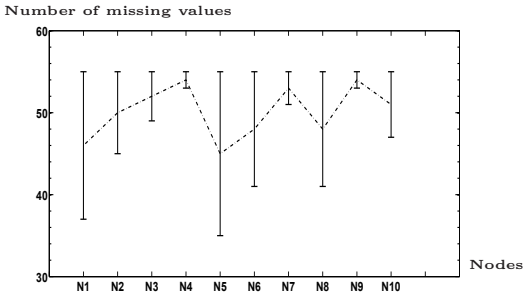

Figure 6: Missing frequents itemsets

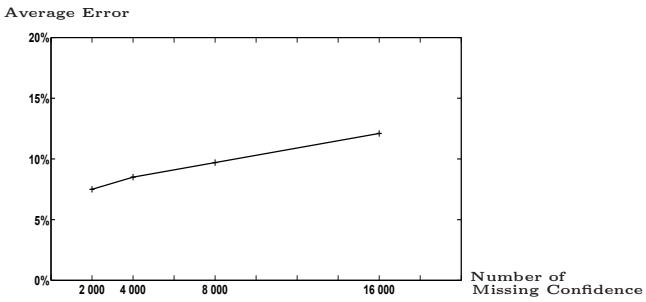

Figure 8: The Error evolution

Table 6: Model Results

	N_1	N_2	N_3	N_4
Nb of frequents Itemsets	46	50	52	54
Relative error ≈	6%	7%	3%	5%

N_5	N_6	N_7	N_8	N_9	N_{10}
45	48	53	48	54	51
6%	7%	8%	7%	1.5%	5%

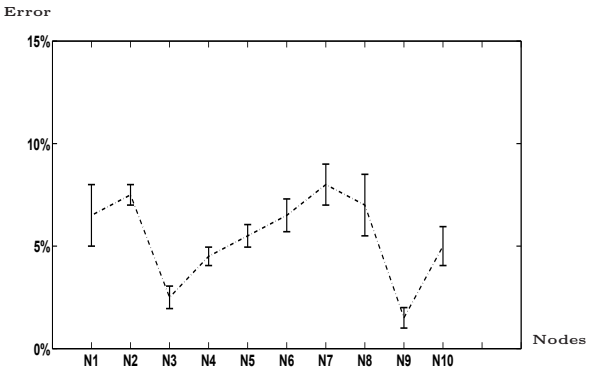

Figure 7: The relative error

missing values in effect in the nodes 9 and 3 was removed or less, the error was 2%. Nevertheless, the error is generally random but stable demonstrating the mutual impact between nodes.
In Figure 6 shows the evolution of the average error as a function of the amount of missing values. This curve shows some stability in our model, the evolution of the error effect is linear, with a maximum value that does not exceed 15%.

7. Related work

Mining data sets in a distributed way has been studied in the past time. Some of the works may choose to download the data set in a single site and performs the data mining operations at the central location. However, the decision should be based on the properties of computing, storage, a,d communication capabilities. A survey of such works is presented in[16]. For instance in[17]considers collective data mining to address data analysis for heterogeneous environment, where instead of combining incomplete local data models, it seeks to find globally meaningful pieces of information from each local sites. In[18] discusses how to find the right tradeoff between the abstraction levels of the local data sources and the global model accuracy is crucial for getting the optimal abstraction, especially when the local data are inter-correlated to different extents, and it is proposed the optimal abstraction task as a game and compute the Nash equilibrium as a solution.

Furthermore, current literature contains significant work on other aspects of cloud computing. For instance,[19] investigates into the performance analysis of high performance computing applications in the cloud environment. The work in[20] concerns Service Level Agreements (SLA). It presents an architecture for detecting SLA violation through sophisticated resource monitoring. Further,[21] proposes a business-oriented federated Cloud computing model in order to facilitate cooperation between multiple independent infrastructure providers so that they seamlessly provide IT infrastructure while taking into account the QoS aspects. Though these approaches have interesting contributions, their focus is different than the work presented in this paper.

In a new application such as vehicle telematics products innovative distributed data mining techniques are a necessity as mentioned in[22]. Additionally, few works addresses the data mining problems on the cloud, but some discussions rises where does cloud computing platform help to perform data analysis on big data?for instance[23] discusses a strategy and a model based on the use of services for the design of distributed knowledge discovery services and discuss how Grid frameworks can be developed as a collection of services and how they can be used to develop distributed data analysis tasks and knowledge discovery processes using the SOA model. In[24] describe the design and implementation of a distributed file system called Sector and an associated programming framework called Sphere that processes the data managed by Sector in parallel. A recent vision paper highlights the distributed data mining in big data in[25].

8. Conclusions and future work

This paper investigated into the critical issues related to cloud data discovery and analysis. In particular it has addressed the problem of discovering missing association rules in circumstances where data is distributed across different nodes of the cloud and some data is missing or erroneous.

A tensor-based framework has been designed and developed in order to model the association rules, the missing data, and to approximate their confidences. Accordingly, we developed a distributed and scalable algorithm to correlate the association rules on a local cloud node with the rules of the global cloud nodes in the presence of missing rules. The algorithm effectively provided an approximation of the confidences of missing rules. Various experiments have been conducted and numerical results are obtained.

In the current approach, the confidence tensor considers the missing data having zero values, which is based relatively on simple hypothesis. The future includes extending the framework to handle (1) more complex hypothesis (2) to discover and analyse data of streaming applications.

References

1. Bhaduri K, Das K, Borne KD, Giannella C, Mahule T, Kargupta H. Scalable, asynchronous, distributed eigen monitoring of astronomy data streams. *Statistical Analysis and Data Mining* 2011;**4**(3):336–352.
2. Gaber MM, Zaslavsky AB, Krishnaswamy S. Data stream mining. In: *Data Mining and Knowledge Discovery Handbook*. 2010, p. 759–787.
3. Gaber MM, Zaslavsky AB, Krishnaswamy S. Mining data streams: a review. *SIGMOD Record* 2005; **34**(2):18–26.
4. Bossche R, Vanmechelen K, Broeckhove J. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Generation Comp Syst* 2013; **29**(4):973–985.
5. Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: *VLDB*. 1994, p. 487–499.
6. Kolda TG, Bader BW. Tensor decompositions and applications. *SIAM Review* 2009;**51**(3):455–500.
7. Fayyad U.M. PSGS. From data mining to knowledge discovery : An overview. *n Advances in Knowledge Discovery and Data Mining* 1996;:1–34.
8. Agrawal R. Srikant R. Fast algorithms for mining association rules. *Proc 20th Int Conf Very Large Data Bases, VLDB* 1994;:487–499.
9. Park J.S. CMYP. An efective hash-based algorithm for mining association rules. *Proceedings of the 1995 ACM SIGMOD international conference on Management of data* 1995;(95):175–186.
10. Toivonen H. Sampling large databases for association rules. *Proceedings of the 22th International Conference on Very Large Data Bases* 1996;(VLDB 96):134–145.
11. Zaki M. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 2000;(12):372–390.
12. Herman. K, Mechelen. IV. Three-way component analysis : Principles and illustrative application. *Psychological Methods* 2001;**6**:84110.
13. Appellof CJ, Davidson. ER. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluent. *Anal Chem* 1981;(53):20532056.
14. Acar E, Kolda TG, Dunlavy DM, Mørup M. Scalable tensor factorizations for incomplete data. *CoRR* 2010; **abs/1005.2197**.
15. Daniel M. Dunlavy TGK, Kegelmeyer. WP. Tensor decompositions for analyzing similarity graphs with multiple linkages. *J Kepner and J Gilbert, editors, Graph Algorithms in the Language of Linear Algebra, Fundamentals of Algorithms SIAM,* 2000;.
16. Park BH, Kargupta H. Distributed data mining: Algorithms, systems, and applications. 2002, p. 341–358.
17. Park BH, Kargupta H, Johnson EL, Sanseverino E, Hershberger DE, Silvestre L. Distributed, collaborative data analysis from heterogeneous sites using a scalable evolutionary technique. *Appl Intell* 2002;**16**(1):19–42.
18. Zhang X, Cheung WK. A game theoretic approach to active distributed data mining. In: *IAT*. 2007, p. 109–115.
19. Expósito RR, Taboada GL, Ramos S, Touriño J, Doallo R. Performance analysis of hpc applications in the cloud. *Future Generation Comp Syst* 2013;**29**(1):218–229.
20. Emeakaroha VC, Netto MAS, Calheiros RN, Brandic I, Buyya R, Rose CAFD. Towards autonomic detection of sla violations in cloud infrastructures. *Future Generation Comp Syst* 2012;**28**(7):1017–1029.
21. Yang X, Nasser BI, Surridge M, Middleton SE. A business-oriented cloud federation model for real-time applications. *Future Generation Comp Syst* 2012;

28(8):1158–1167.
22. Kargupta H. Connected cars: How distributed data mining is changing the next generation of vehicle telematics products. In: *S-CUBE*. 2012, p. 73–74.
23. Talia D, Trunfio P. How distributed data mining tasks can thrive as knowledge services. *Commun ACM* 2010; **53**(7):132–137.
24. Gu Y, Grossman RL. Toward efficient and simplified distributed data intensive computing. *IEEE Trans Parallel Distrib Syst* 2011;**22**(6):974–984.
25. Intel's perspective on data at the edge, distributed data mining and big data, a vision paper. August 2012, .