Generation of Incomplete Test-Data using Bayesian Networks

Olivier François and Philippe Leray

***Abstract*— We introduce a new method based on Bayesian Network formalism for automatically generating incomplete datasets. This method can either be configured randomly to generate various datasets with respect to a global percentage of missing data or manually in order to handle many parameters. [1] proposed three types of missing data : MCAR (*missing completly at random*), MAR (*missing at random*) and NMAR (*not missing at random*). The proposed approach can successfully generate all MCAR data mechanisms and most of MAR data mechanisms. NMAR data generation is very difficult to manage automatically but we propose some hints in order to cover some of the NMAR data situations.**

I. Introduction

Software testing is a time-expensive component of software development. A particulary labor-intensive component of this testing process is the generation of test data to satisfy testing requirements. This is the primary method to establish confidences in the performances of softwares. These confidences are ordinarily established by executing the algorithm on test-data chosen by some systematic testing procedure.

Through the years, various methods for generating test-data have been proposed.
These methods have been divided into three classes [2] :

- Random test-data generation [3], [4],
- Structural or path-oriented test-data generation [5], [6] (for instance with mutation analysis and constraint satifaction [7]) and
- Goal oriented test-data generation [8] (for instance with genetic algorithms [9]).

Our approach could be classify in the random test-data generation class as it selects inputs randomly from the underlying probability distribution given by a specific Bayesian Network.

Another field of application of our work concerns Machine Learning : learning algorithms also have to be compared in various contexts. Some of them deal with incomplete databases but available datasets with incomplete data do not cover all the missing data processes with various missing data rate.

Our method aims at modelling missing data processes using Bayesian Network formalism, in order to automatically create incomplete datasets with different characteristics.

We first introduce Bayesian Network formalism and its use for data generation. The next section is devoted to missing data mechanisms. We then propose to model these mechanisms with Bayesian Networks and give some hints in order to constraint the global percentage of missing data.

INSA Rouen, LITIS – Information Processing and Computer Science Lab, BP 08, 76801 Saint-Etienne-Du-Rouvray Cedex, France
(email: Francois.Olivier.C.H@gmail.fr; Philippe.Leray@insa-rouen.fr).

II. Bayesian Network formalism

A. Preliminaries

Bayesian Networks aim at modelling complex systems [10], [11] by taking graphically into account conditional independences between variables (by means of a directed acyclic graph) and by giving a compact representation of the joint probability distribution as the product of local conditional probability distributions (one for each node in the graph).

In this paper, we are going to use $\mathfrak{Fractur}$ font for probability distributions, $\mathcal{SCRIPT}$ font for random variables and $\mathbf{Bold}$ font for instantiations. Moreover, we will use Normal font for nodes which represent random variables in Bayesian Networks and $Italic$ font for all other notations.
In a Bayesian Network, we have the following decomposition of the joint probability distribution:

$$\mathbb{P}(\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_n) = \prod_{i=1}^{n} \mathbb{P}(\mathcal{X}_i | \mathcal{P}a(\mathrm{X}_i)) \qquad (1)$$

where $\mathcal{P}a(\mathrm{X}_i)$ is the random vector built from the parent set of node X_i in the graph associated to the Bayesian Network.

Bayesian Networks, as other probabilistic models, can be used as generative models, that's why we choose this formalism to generate Test-Data.

B. Sampling in Bayesian Networks

The first idea of stochastic methods is using knowledge about a distribution (here conditionnal probabilities) to automatically generate samples following this distribution.

Probabilistic logic sampling [12] is the simplest and the first proposed sampling algorithm for Bayesian Networks. Rejection sampling could be used when $\mathbb{P}$ is not known and if a function $\mathbb{Q}$ that satisfy $1 \leqslant \frac{\mathbb{P}}{\mathbb{Q}} \leqslant M$ is known, where M is a known bound. The generated sample is accepted with the probability $\frac{\mathbb{P}(\mathcal{X})}{M \cdot \mathbb{Q}(\mathcal{X})}$. So if M is too large, we rarely accept samples.

Importance sampling [13], [14] is close to the logic sampling algorithm except that the importance function $\mathbb{Q}$ is updated to periodically revise the conditional probability tables in order to make the sampling distribution gradually approach the posterior distribution. The generated sample is accepted with the probability $\min\left(1, \frac{\mathbb{P}(x)\mathbb{Q}(x|x^{t-1})}{\mathbb{P}(x^{t-1})\mathbb{Q}(x^{t-1}|x)}\right)$.

Another family of stochastic sampling methods is formed by so-called Markov Chain Monte Carlo (MCMC) methods that are divided into Gibbs sampling, Metropolis sampling, and Hybrid Monte Carlo sampling [15], [16]. When applied to Bayesian Networks [17], [18], [19], those approaches determine the sampling distribution of a variable from its previous sample given its Markov blanket [20].

Gibbs sampling is a special case of Metropolis-Hastings algorithm which is applicable to state spaces in which we have a factored state space and access to the full conditionals. So it is perfect for Bayesian Networks.

The idea is to transit from one state (variable assignment) to another iteratively. The algorithm is simple and could be describe as follows :

1) pick a variable,
2) sample its value from the conditional distribution,
3) goto step 1) until all variables are not instanciated.

The importance ratio is $r = \dfrac{\mathbb{P}(x)\mathbb{Q}(x|x^{t-1})}{\mathbb{P}(x^{t-1})\mathbb{Q}(x^{t-1}|x)} = 1$ as $\mathbb{Q} = \mathbb{P}$ in the Gibbs sampler, so we always accept the new sample.

We propose to use the Gibbs sampler. Let choose the variable in step 1) in the set of root nodes as their *a priori* probability tables give the optimal importance function. If there is no more root nodes, let pick a node that have all its parent set instanciated.

C. Random generation of Bayesian Network structures

As the proposed approach is aimed to build datasets from various mechanisms of data generation and deletion, it should be judicious to randomly generate Bayesian Network structures. For instance, [21] or [22] have proposed methods for random generation of bayesian networks based on Markov chains.

III. Introduction to missing data

A. Notations

Let n and m be natural integers and let $\mathcal{X}_1^1, \ldots, \mathcal{X}_n^1, \mathcal{X}_1^2, \ldots, \mathcal{X}_n^m$ be $n \times m$ random variables that respectively follow distributions $\mathfrak{X}_i^j$, for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$. Suppose that we have a dataset

$$\mathbf{D} = [[\mathbf{x}_i^j]]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant m}}$$

This dataset $\mathbf{D}$ is an instantiation of the random vector $\mathcal{D} = (\mathcal{X}_1^1, \ldots, \mathcal{X}_n^1, \mathcal{X}_1^2, \ldots, \mathcal{X}_n^m)$. This vector $\mathcal{D}$ follows the distribution $\mathfrak{D} = (\mathfrak{X}_1^1, \ldots, \mathfrak{X}_n^1, \mathfrak{X}_1^2, \ldots, \mathfrak{X}_n^m)$. Let θ be the parameters of $\mathfrak{D}$. Let $\mathbf{x}_i^j$ be the instantiation of $\mathcal{X}_i^j$ in $\mathbf{D}$.

Let $\mathcal{R} = (\mathcal{R}_1^1, \ldots, \mathcal{R}_n^1, \mathcal{R}_1^2, \ldots, \mathcal{R}_n^m)$ be the random vector where the random variable $\mathcal{R}_i^j$ takes the value $\mathbf{1}$ if $\mathcal{X}_i^j$ is said to be missing and takes the value $\mathbf{0}$ if $\mathcal{X}_i^j$ is observed.

The random vector $\mathcal{R}$ follows a distribution named $\mathfrak{R} = (\mathfrak{R}_1^1, \ldots, \mathfrak{R}_n^1, \mathfrak{R}_1^2, \ldots, \mathfrak{R}_n^m)$. Let μ be the parameters of the distribution $\mathfrak{R}$.

Let $\mathbf{R} = [[\mathbf{r}_i^j]]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant m}}$ be the matrix where $\mathbf{r}_i^j = \mathbf{0}$ if $\mathcal{X}_i^j$ is observed and $\mathbf{1}$ if not.

Finally, let $\mathbf{O} = \{\mathbf{x}_i^j\}_{\{(i,j)|\mathbf{r}_i^j=\mathbf{0}\}}$ be the part of the dataset $\mathbf{D}$ that will be observed and let $\mathbf{H} = \{\mathbf{x}_i^j\}_{\{(i,j)|\mathbf{r}_i^j=\mathbf{1}\}}$ be the part of the dataset $\mathbf{D}$ that will not be observed and

$$\mathbf{D} = \{\mathbf{O}, \mathbf{H}\}$$

Notice that in these notations, the dataset $\mathbf{D}$ is a complete dataset. The variables $\mathcal{H}$ are measured in $\mathbf{D}$, but ”forgotten” because of $\mathcal{R}$ variables. $\mathbf{D}_{measured} = \{\mathbf{O}, \mathbf{H} = missing\}$ is the ”real” incomplete dataset containing only $\mathbf{O}$.

Our approach will sample $\mathbf{D}$ and $\mathbf{R}$ from distribution $\mathfrak{D}$ and $\mathfrak{R}$ that are modelled by a Bayesian Network. The output is $\mathbf{D}_{measured}$ which is built by taking only the values $\mathbf{x}_i^j$ in $\mathbf{D}$ where $\mathbf{r}_i^j = \mathbf{0}$.

As a lot of distributions are concerned, assumptions have to be made to simplify the model, but before, let us describe the different missing data mechanisms.

B. Missing Data mechanisms

Rubin [1] has highlighted that

$$\mathbb{P}(\mathcal{O}, \mathcal{H}, \mathcal{R}|\theta, \mu) = \mathbb{P}(\mathcal{O}, \mathcal{H}|\theta) \cdot \mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) \qquad (2)$$

as $\mathcal{R}$ does not depend on θ and $\mathcal{D}$ does not depend on μ.

we can distinguish three missing data mecanisms according to the distribution $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu)$.

a) MCAR: The data is said to be *Missing Completely At Random*. In this case, the missing data is independent either of the observed ones, and of the others missing data and $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mu)$ (for instance $\mathbb{P}(\mathcal{R}_i^j = 1) = \alpha$, for all i and j).

b) MAR: If we consider the situation where a data is not systematically measured for a special configuration of the other variables then this data is said to be *Missing At Random*. Here, the missing data is dependent of the observed ones, but is independent of the others missing data, i.e. $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mathcal{O}, \mu)$.

c) NMAR: In the last situation, a missing data can be a consequence of the actual state of any variable and we can not simplify $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu)$. The data is said to be *Not Missing At Random*, as it could exists deterministic mechanisms to empty the dataset. For instance, we can imagine that variable $\mathcal{X}$ can not be measured between times t and $t + T$ because of a sensor failure. Even if the fact that the sensor failure is non-deterministic, the fact that data is missing depends on time and then the dataset should not be *i.i.d* any longer.

C. Assumptions for MAR and NMAR situations

Usually, we assume that data are independent and identically distributed (*i.i.d*). Samples $\mathbf{x}^j = (\mathbf{x}_1^j, \cdots, \mathbf{x}_n^j)$ do not depend to each other This hypothesis leads to the following assumption :

A1: *”if $j \neq l$, random variables $\mathcal{X}_i^j$ and $\mathcal{X}_k^l$ are independent”*

Another hypothesis that is often made is the stationnarity. In other term, we suppose that the sampling distribution do not vary during the sampling phase. This claim can be reformulate as

A2: *"For all j, distributions $\mathfrak{X}_i^j$ are the same"*

In the following, we will forgot the 'j' by calling this unique distribution $\mathfrak{X}_i$ and will do the same for variables $\mathcal{X}_i$ if the context is clear.

To be coherent, the same assumptions will be made on distributions $\mathcal{R}_i^j$ as there is no reason that the missingness mechanism varies over time. This claim implies the two following assumptions:

A3: *"if $j \neq l$, random variables $\mathcal{R}_i^j$ and $\mathcal{R}_k^l$ are independent"*

A4: *"For all j, distributions $\mathfrak{R}_i^j$ are the same*

For the remainder of this paper, we will forgot the 'j' index by naming this unique distribution $\mathfrak{R}_i$ and will do the same for variables $\mathcal{R}_i$ if the context is clear.

As the dataset we want to create is *i.i.d*, the fact that a data is missing does not depend on the next of previous values in the dataset. This claim implies that

A5: *"if $j \neq l$, random variables $\mathcal{X}_i^j$ and $\mathcal{R}_k^l$ are independent"*

In section VI, some hints will be given to soften some of these assumptions in NMAR situations.

D. Our general approach

The approach we propose here stands for Bayesian Network formalism. We first assume that we have a generative model $\mathfrak{X}$ that can be use to generate a complete dataset. This model is the bayesian network in the top box of the figure 1. It only contains variables $\mathcal{X}_i$ that are represented by nodes named X_i using assumption A1 and A2.

We then have to add new variables $\mathcal{R}_i$ in this model in order to indicate if each variable $\mathcal{X}_i$ is measured or not. Using assumptions A3 and A4, we will represent those variables by nodes named R_i.

The output sample of our approach are the values of nodes M_i which are evaluated using only X_i's and R_i's values.

Not only the way we will connect nodes R_i, X_i and M_i together will lead us to MCAR or MAR situations but also the way we create Conditional Probability Tables as we could include independencies from these tables.

As, for the sample j, variable $\mathcal{M}_i$ takes either the value taken by $\mathcal{X}_i$ if $\mathcal{R}_i = \mathbf{0}$ or the value $missing$ if $\mathcal{R}_i = \mathbf{1}$, its Conditionnal Probability Table (CPT) is known and is

$\mathcal{X}_i, \mathcal{R}_i$ \ $\mathcal{M}_i$	v_1	v_2	$v_{...}$	v_{si}	$missing$
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_1\ , \mathrm{R}_i = 0)$	1	0	0	0	0
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_2\ , \mathrm{R}_i = 0)$	0	1	0	0	0
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_{...}, \mathrm{R}_i = 0)$	0	0	1	0	0
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_{s_i}, \mathrm{R}_i = 0)$	0	0	0	1	0
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_1\ , \mathrm{R}_i = 1)$	0	0	0	0	1
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_2\ , \mathrm{R}_i = 1)$	0	0	0	0	1
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_{...}, \mathrm{R}_i = 1)$	0	0	0	0	1
$\mathbb{P}(\mathrm{M}_i \vert \mathrm{X}_i = v_{s_i}, \mathrm{R}_i = 1)$	0	0	0	0	1

(3)

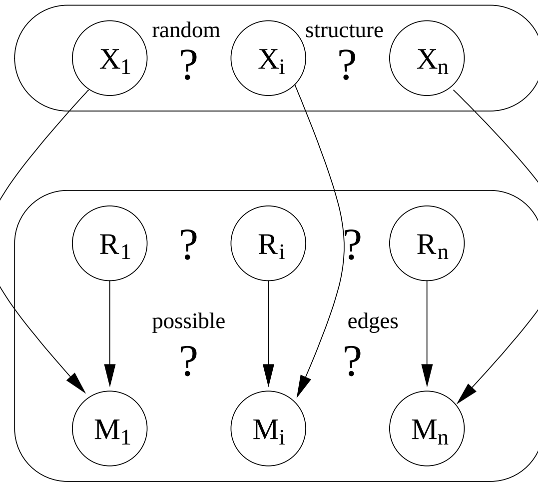

Fig. 1. Generic Bayesian Network for incomplete Test-Data generation.

where s_i denote the size of the variable $\mathcal{X}_i$. Notice that using zero probabilities in CPTs could introduce independencies. For instance, here, the fact that $\mathcal{M}_i = missing$ depend only on the value $\mathbf{1}$ for $\mathcal{R}_i$.

The two next sections will describe our modelisation for MCAR and MAR mechanisms. We will then adapt these models in order to take into account some NMAR situations.

IV. MCAR MECHANISMS MODELING

A. The model

Methods that have been proposed for MCAR dataset generation usually remove data for each variable with the same probability α. We propose here a more general method where a different "missing" probability is associated to each variable.

As $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mu)$ for MCAR mechanisms and with assumption A3, we have

$$\mathbb{P}(\mathcal{R}|\mu) = \prod_{j=1}^{m} \mathbb{P}(\mathcal{R}_1^j, \cdots, \mathcal{R}_n^j|\mu) \tag{4}$$

but we can't have a smaller decomposition as independences between $\mathcal{R}_i^j$, for a given j, are not known. But, with assumption A4, they do not depend on j.

Then in a MCAR process, we can imagine having rules such as : "if $\mathcal{X}_i$ is missing then $\mathcal{X}_k$ is missing too".

In a MCAR mechanism, $\mathcal{M}_i$ depend on $\mathcal{R}_i$ and $\mathcal{X}_i$, for fixed i, but $\mathcal{R}_i$'s could have interdependencies when i varies. The way we represent a generic MCAR mechanism is illustrated in figure 2.

In a first subsection we are going to highlight some parameters that have to be fixed and how to fix them. Then we are going to built a simple MCAR mechanism and explain how it works.

B. Identifying parameters

The only parameter users usually want to specify is the goal probability of missingness. Let α be that goal. With our

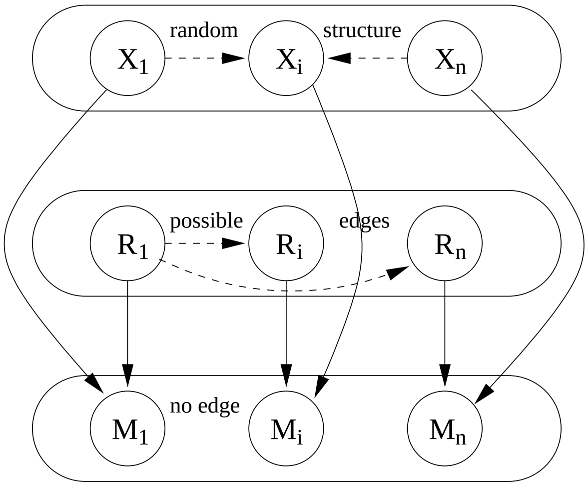

Fig. 2. Bayesian Network modelization for MCAR mechanism.

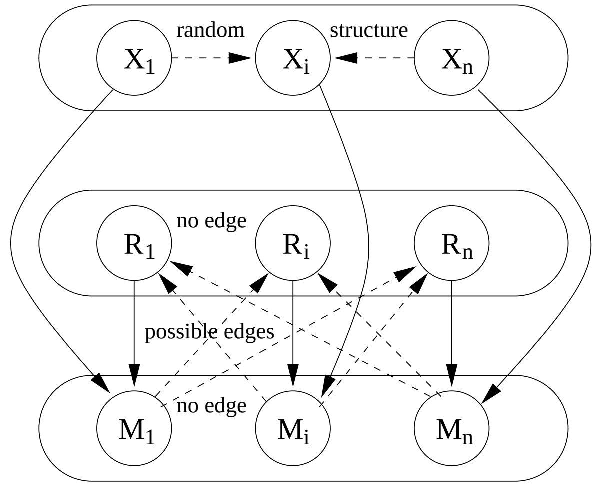

Fig. 3. Bayesian network modelization for MAR mechanism.

notations, we have

$$\mathbb{E}(\overline{\mathcal{R}}) = \alpha \tag{5}$$

where $\overline{\mathcal{R}} = \dfrac{1}{n \cdot m} \sum_{i,j} \mathcal{R}_i^j$.

Then assumptions A3 and A4 give $\mathbb{E}(\overline{\mathcal{R}}) = \dfrac{1}{n} \sum_{i=1}^{n} \mathbb{E}(\mathcal{R}_i^1)$.

As $\mathbb{E}(\mathcal{R}_i^1) = \sum_{\mathbf{r} \in \{0,1\}} \mathbf{r} \cdot \mathbb{P}(\mathcal{R}_i^1 = \mathbf{r}) = \mathbb{P}(\mathcal{R}_i^1 = \mathbf{1})$, we have

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \mathbb{P}(\mathcal{R}_i^1 = \mathbf{1}) \tag{6}$$

In the following, let β_i be the probability $\mathbb{P}(\mathcal{R}_i^1 = \mathbf{1})$.

To have an entirely automated (an randomized) method to compute an incomplete dataset generation process we have to build the β_i's randomly. To do so, we could gener a random vector $[\beta_1, \beta_{n-1}]$ of $n-1$ values in $[\alpha - \varepsilon, \alpha + \varepsilon] \subset [0, 1]$ uniformly around α an then choose $\beta_n = n \cdot \alpha - \sum_{i=1}^{n-1} \beta_i$.

C. A simple example

Suppose we want to create MCAR data and we have the following assumption (only for this example):
"Random variables $\mathcal{R}_i$ and $\mathcal{R}_k$ are marginally independent if $i \neq k$".
Then we can represent our dependencies by the Bayesian Network described figure 2 (but without any link between the R_i's).

To fill the *a priori* probability tables of nodes R_i, we have to generate the probability vector $\beta = (\beta_1, \cdots, \beta_n)$ with respect to the constraint given by equation 6: $\alpha = \dfrac{1}{n} \sum_{i=1}^{n} \beta_i$.
Then, we simply have to use the result to build tables

$$\mathbb{P}(\mathcal{R}_i) = [1 - \beta_i\ ,\ \beta_i] \tag{7}$$

D. General situation

In pratice, one could prefer to build more general MCAR mechanism. In this situation, edges between nodes R_i have to be created. The model that is schematized by the figure 2 is obtained. The previous method proposed to fill the probability tables does not work anymore as we have to specify values for Conditional Probability Tables (CPTs) instead of *a priori* probability tables. The methodology used to determine the CPTs is close to the one presented in section IV-B and will be presented in section V-B as we also use it in a more general way for MAR mechanisms.

V. MAR MECHANISMS MODELING

A. The model

For MAR processes, nodes representing the missingness of a variable can no longer be disconnected from observable nodes (remember that $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mathcal{O}, \mu)$) as we could see in figure 3.

Edges between M_i's and X_i's are allowed to take into account the fact that the probability of being missing is no more independent of the observable part of the system.

Here, we don't need edges between R_i's anymore as dependencies between R_i's comes from the fact that R_i is dependent of some M_i's and that M_i's are dependent of some other R_i's.

With this model, we have to adjust the probability of a node to be missing with respect to other variable values. Then we have to fix random values for $\mu = (\mu_{ibk})_{ibk}$ where

$$\mu_{ibk} = \mathbb{P}(\mathcal{R}_i^1 = \mathbf{b} | \mathcal{P}a(\mathrm{R}_i) = \mathbf{k}) \tag{8}$$

with $1 \leqslant i \leqslant n$, $\mathbf{b} \in \{\mathbf{0}, \mathbf{1}\}$ and $\mathbf{k} \in K_i$ where K_i is the set of possible configurations of $\mathcal{P}a(\mathrm{R}_i)$.

As the probability of a node to be missing does not depend on the unobserved part of the system, some other constraints on the way to fix the μ_{ibk}'s have to be made and we must have
$\mathbb{P}(\mathcal{R}_i = \mathbf{1} | \mathcal{P}a(\mathrm{R}_i) = completly\ missing) = \beta_i$ and

$\mathbb{P}(\mathcal{R}_i = \mathbf{1}|\mathcal{P}a(\mathrm{R}_i) = partially\ missing) =$
$= \mathbb{P}(\mathcal{R}_i = \mathbf{1}|\text{observed part of } \mathcal{P}a(\mathrm{R}_i))$
which is obtained by inference in the bayesian Network.

B. Identifying parameters

Remember that $\beta_i = \mathbb{P}(\mathcal{R}_i = \mathbf{1})$, then $\beta_i = \sum_{\mathbf{k}} \mathbb{P}(\mathcal{R}_i = \mathbf{1}, \mathcal{P}a(\mathrm{R}_i) = \mathbf{k})$ and the Bayes formula gives

$$\beta_i = \sum_{\mathbf{k}} \mu_{i\mathbf{1}k} \cdot \xi_{ik} \tag{9}$$

where $\xi_{ik} = \mathbb{P}(\mathcal{P}a(\mathrm{R}_i) = \mathbf{k})$. All the ξ values can be obtained by inference in the Bayesian Network.

Then we simply have to use a methodology close to the previous one used to gener β_i's to randomly generate parameters μ_{i1k} with additional verification tests to validate the supplement constraints explained in the previous paragraph.

Now suppose R_i has only one parent node $\mathrm{M}_{i'}$. The whole conditionnal probability table $\mathbb{P}(\mathcal{R}_i|\mathcal{P}a(\mathrm{R}_i))$ is then completly determined by

$$^t\begin{bmatrix} \mu_{i01}, \cdots, \mu_{i0k}, \cdots, \mu_{i0s_i}, 1-\beta_i \\ \mu_{i11}, \cdots, \mu_{i1k}, \cdots, \mu_{i1s_i}, \quad \beta_i \end{bmatrix}$$

where s_i is the size of the node X_i and $\mu_{i0k} = 1 - \mu_{i1k}$.

If R_i has more than one parents, we have to consider that some μ_{ibk} must be identical and then, we simply have to sum the corresponding γ_{ik}, to factorise, and to use the same kind of formula.

If the two last recommendations of section V-A are not checked, we need to model NMAR mechanisms. But there is a lot of ways to build such mechanisms. Let us give some hints to built some of them.

VI. NMAR MECHANISMS MODELING

The main problem of NMAR mechanisms is that, in practice, samples of the dataset are usually no longer *i.i.d*. The number of NMAR mechanisms that could lead to pseudo-real data is infinite as it depends on external factors, so we are free to imagine all the factors we want.

A first hint to generate NMAR data is modifying the conditional probability densities previously described in MAR or MCAR situations in order to lost the specific independences entailed in these densities.

Another solution will be the use of a dynamic bayesian network [23] to represent data samples that are time dependent, for instance by connecting $R_i(t)$ to $R_i(t+1)$. With this solution the fact that a variable is missing will become a Markov chain.

Another mean could be to introduce one or many new variables and to build dependencies between $\mathcal{R}_i$ and those variables by drawing new edges on the Bayesian Network.

We could also imagine to mix all this processes.

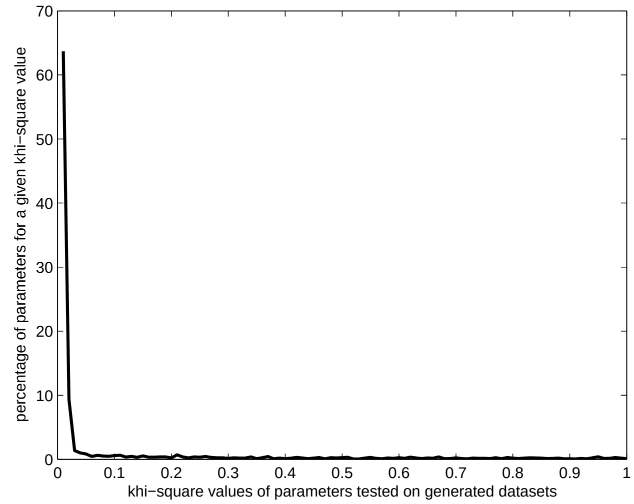

Fig. 4. Histogram of χ^2 value of parameters tested from generated samples.

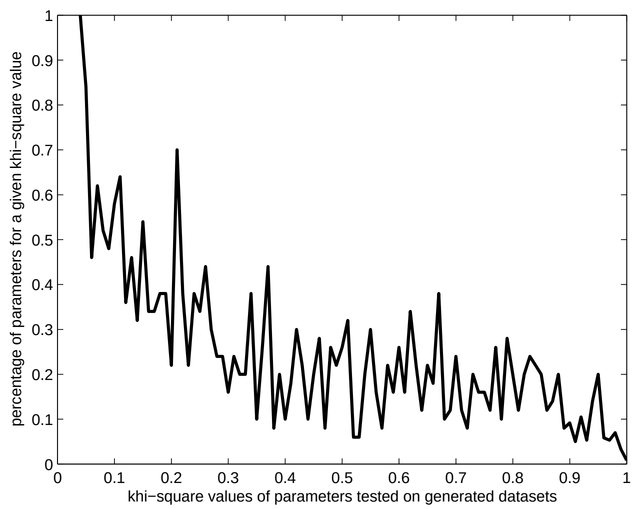

Fig. 5. Zoom of the flat part of Figure.4.

VII. EXPERIMENTATIONS

For the experimentation stage, we have used our formalism to generate datasets from randomly generated Bayesian networks (between 4 and 13 nodes). Those networks have been used to gener MAR incomplete datasets with 10000 samples with a percentage of missingness which is randomly chosen between 15% and 40% (results on MCAR datasets are similar). Then we pick up different parameters which model the percentage of missingness of an attribute in a specific context for each incomplete dataset generative Bayesian network. We then calculate the χ^2 critical value that this parameter has if we test it on the corresponding generated dataset.

In figure 4, an histogram of Chi-square values of parameters tested on generated datasets is shown.

As we could see on figures 4 and 5, the distribution of Chi-square values is high for small values (*i.e.* < 0.05) and arround 65% of the parameters tested have a Chi-square value smaller than 0.01.

On figure 5, we could see that arround 0.02% of tested parameters could have a fixed Chi-square value higher than 0.3. Those values are reach for parameters that lead to a small number of samples in the datasets. Then the tests are not reliable in this case as the number of corresponding samples is often smaller than 20 samples.

VIII. Conclusion and future work

The method we proposed here aims at modelling missing data processes using Bayesian Network formalism, in order to automatically create incomplete datasets with different characteristics. We first used a generative model $\mathfrak{X}$ to generate a complete dataset. We then added new variables $\mathcal{R}_i$ and $\mathcal{M}_i$ in this model in order to indicate if each variable $\mathcal{X}_i$ is measured or not and to give the value of the measured variables. The connectivity between variables $\mathcal{R}_i$, $\mathcal{M}_i$ and $\mathcal{X}_i$ and a specific form of the corresponding conditional probability densities lead to MCAR or MAR models. We also proposed some first hints to adapt these models in order to take into account some NMAR situations.

For each of the MCAR and MAR models proposed here, we also described how to randomly generate the parameters with respect to the global probability of missingness α we want to reach.

The methodology described here for discrete data generation can be extended to continuous data by using conditional gaussian models for $\mathcal{X}$ variables, and softmax functions for $\mathbb{P}(\mathcal{R}_i|\mathcal{X}_j, \mathcal{R}_k)$ conditional probability distributions.

An experimental phase has been done to show the reliability of such a method when we gener MAR incomplete datasets.

We have implemented all these models in *BNT Matlab toolbox* [24] and functions are freely available in the *Structure Learning Package* [25]. An application of this work could be seen in [26] with more than 4300 generated datasets.

We plan to identify some usual NMAR situations and model those situations with the method proposed here, by example using Dynamic Bayesian Network formalism to model a sensor that will have a given lifetime.

Acknowledgment

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors views.

References

[1] D. Rubin, “Inference and missing data,” *Biometrika*, vol. 63, pp. 581–592, 1976.

[2] R. Ferguson and B. Korel, “The chaining approach for software test data generation,” *ACM TOSEM*, vol. 5(1), pp. 63–86, 1996.

[3] P. Thévenod-Fosse and H. Waeselynck, “Statemate: Applied to statistical software testing,” in *ACM SIGSOFT, Proceedings of the 1993 International Symposium on Software Testing ans Analysis*, vol. Software Enginering Notes 23(2), 1993, pp. 78–81.

[4] W. Deason, D. Brown, K. Chang, and J. Cross II, “A rule-based software test data generator,” *IEEE transactions on Knowledge and Data Engineering*, vol. 3(1), pp. 108–117, 1991.

[5] C. Ramamoorthy, S. Ho, and C. W.T., “On the automated generation of program test data,” *IEEE Transactions on software Engineering*, vol. 2(4), pp. 193–300, 1976.

[6] B. Korel, “Automated software test data generation,” *IEEE transations on Software Engineering*, vol. 16(8), pp. 870–879, 1990.

[7] R. DeMillo and J. Offutt, “Constraint-based automatic test data generation,” *IEEE Transactions on Software Engineering*, vol. 17(9), pp. 900–910, 1991.

[8] B. Korel and A. Al-Yami, “Assertion-oriented automated test data generation,” in *Proceedings of the 18th International Conferance on Software Engineering (ICSE)*, vol. 16(8), 1996, pp. 71–80.

[9] R. Pargas, M. Harrold, and R. Peck, “Test-data generation using genetic algorithms,” *Journal of Software testing, Verification and reliability*, vol. 9, pp. 263–282, 1999.

[10] F. Jensen, *An introduction to Bayesian Networks*. Taylor and Francis, London, United Kingdom, 1996.

[11] J. Pearl, “Graphical models for probabilistic and causal reasoning,” in *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 1: Quantified Representation of Uncertainty and Imprecision*, D. M. Gabbay and P. Smets, Eds. Dordrecht: Kluwer Academic Publishers, 1998, pp. 367–389.

[12] M. Henrion, “Propagating uncertainty in Bayesian networks by probabilistic logic sampling,” in *Uncertainty in Artificial Intelligence 2*, J. F. Lemmer and L. M. Kanal, Eds. Amsterdam: Elsevier Science Publishing Company, 1988, pp. 149–163.

[13] R. Shachter and M. Peot, “Simulation approaches to general probabilistic inference on belief networks,” in *Proceedings of Uncertainty in Artificial Intelligence 5*. Elsevier Science Publishing Company, 1989, pp. 221–231.

[14] J. Cheng and M. Druzdzel, “Ais-bn: An adaptive importance sampling algorithm for evidential reasoning in large bayesian networks,” *Journal of Artificial Intelligence Research*, vol. 13, pp. 155–188, 2000.

[15] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[16] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics. Chapman & Hall, 1996.

[17] J. Pearl, “Evidential reasoning using stochastic simulation of causal models,” *Artificial Intelligence*, vol. 32, no. 2, pp. 245–258, 1987.

[18] M. Chavez and G. Cooper, “A randomized approximation algorithm for probabilistic inference on bayesian belief networks.” *Networks*, vol. 20, no. 5, pp. 661–685, 1990.

[19] J. York, “Use of the Gibbs sampler in expert systems,” *Artificial Intelligence*, vol. 56, pp. 115–130, 1992.

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, second edition in 1991, 1988.

[21] Y. Xiang and T. Miller, “A well-behaved algorithms for simulating dependence structure of bayesian networks,” *International Journal of Applied Mathematics*, vol. 1, pp. 923–932, 1999.

[22] J. Ide, F. Cozman, and F. Ramos, “Generating random bayesian networks with constraints on induced width,” in *European Conference on Artificial Intelligence (ECAI)*. IOS Press, Amsterdan, 2004, pp. 323–327.

[23] K. Murphy, “Dynamic bayesian networks: Representation, inference and learning,” Ph.D. dissertation, University of california, Berkeley, 2002.

[24] ——, “Bayes net toolbox v5 for matlab,” Cambridge, MA: MIT Computer Science and Artificial Intelligence Laboratory, 2004, http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html.

[25] P. Leray and O. François, “BNT structure learning package: Documentation and experiments,” Laboratoire PSI, INSA de Rouen, Tech. Rep. 2004/PhLOF, 2004, http://bnt.insa-rouen.fr/.

[26] O. François, “De l’identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d’informations complètes ou incomplètes,” Ph.D. dissertation, Institut National des Sciences Appliquées de Rouen (INSA), http://asi.insa-rouen.fr/~ofrancois/pdf/these.pdf, 2006.