EMPIRICAL EVALUATION OF METHODS OF FILLING THE MISSING DATA IN LEARNING PROBABILISTIC MODELS

Adrian Adam Falkowski, Anna Łupińska–Dubicka

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Missing data is a common problem in statistical analysis and most practical databases contain missing values of some of their attributes. Missing data can appear for many reasons. However, regardless of the reason for the missing values, even a small percent of missing data can cause serious problems with analysis reducing the statistical power of a study and leading to draw wrong conclusions. In this paper the results of handling missing observations in learning probabilistic models were presented. Two data sets taken from UCI Machine Learning Repository were used to learn the quantitative part of the Bayesian networks. To provide the opportunity to compare selected data sets did not contain any missing values. For each model data sets with variety of levels of missing values were artificially generated. The main goal of this paper was to examine whether omitting observations has an influence on model's reliability. The accuracy was defined as the percentage of correctly classified records and has been compared to the results obtained in the data set not containing missing values.

Keywords: missing data, probabilistic models, Bayesian networks, classification

1. Introduction

Missing values (or missing data) are a common problem in statistical analysis and most practical databases contain missing values of some of their attributes. They can have a significant effect on the conclusions that can be drawn from the data. There are several reasons why data may be missing. Sometimes they result from malfunctioning equipment, sometimes the value of the attribute is not known, or the data were not entered correctly. However, regardless of the reason for the missing values, the fact that a measurement is missing is a complication for any algorithm that analyzes the data.

Advances in Computer Science Research, vol. 14, pp. 55-67, 2018. DOI: 10.24427/acsr-2018-vol14-0004

This paper presents the results of handling missing values in problem of learning quantitative part of probabilistic models, in particular one of their prominent members – Bayesian networks. One of the most important features of Bayesian networks is the fact that they provide an elegant mathematical structure for modeling complicated relationships among random variables while keeping a relatively simple visualization of these relationships.

The experiments involved learning the conditional probability distribution of models created on the basis of two data set taken from UCI Machine Learning Repository [13]: *Car Evaluation* and *Nursery*. Original data set contained no missing values and for each case several data sets with variety of levels of missing values were artificially generated. The main purpose of this article was to study whether method of filling missing values in given data set has an influence on model's reliability. The accuracy was defined as the percentage of correctly classified records and has been compared to the results obtained in the data set not containing missing values.

The remainder of this paper is structured as follows. Section 2. explains the basic concepts of Bayesian networks. Section 3. explains the problem of missing data and shortly outlines the methods dealing with it. Section 4. introduces selected data sets and presents created Bayesian network models. Section 5. presents the results of experiments conducted on data sets with several levels of missing data. Section 6. concludes the paper.

2. Bayesian Networks

Bayesian networks (also knows as belief networks or causal networks, BNs) [8], belonging to to the family of probabilistic graphical models, are widely used to represent knowledge about an uncertain domain. In particular, each node of a network represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. Very often, the structure of the graph is given a causal interpretation, convenient from the point of view of knowledge engineering and user interfaces. Bayesian networks allow for computing probability distributions over subsets of their variables conditional on other subsets of observed variables. BNs are widely applied in decision support systems, where they typically form the central inferential engine.

Formally, a Bayesian network is a pair $<\mathcal{G},\Theta>$, where $\mathcal{G}$ is an acyclic directed graph which nodes represent random variables $X_1, X_2, \ldots, X_n$ and edges represent direct dependencies between these variables. The second component of a Bayesian network, Θ, denotes the set of parameters that describes a conditional distribution for each node X_i in $\mathcal{G}$, given its parents in $\mathcal{G}$, i.e., $P(X_i|Pa(X_i))$. Bayesian network

defines a unique joint probability distribution (JPD) over set of its variables, namely:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{N} P(X_i | Pa(X_i)) \tag{1}$$

where $Pa(X_i)$ represents set of parents of X_i.

Knowing only local conditional probabilities of network variables the occurrence of a specific state can be determined using Equation 1. And since each variable in the network depends on them either directly or indirectly, the expected value of the any variable can be calculated knowing the values of the variable that do not have parents in the graph (the root cause).

Note that in the Equation 1, probability of a random variable X_i depends only on the states of its parents. The graph $\mathcal{G}$ encodes conditional independence assumptions, by which each variable X_i is independent of its nondescendents given its parents in $\mathcal{G}$. This simplification allows to represent the joint probability distribution more compactly and thus to reduce, sometimes significantly, the number of parameters that are required to characterize the JPD of the variables[5,8,10]. In case of network consisting of n binary nodes, the full joint probability distribution would require storing 2^n values. Using the factored form would require $n2^k$, where k is the maximum number of parents of a node.

3. Methods of handling missing data

According to Little and Rubin [7] three classes of possible mechanism of missing data can be distinguished. Each of these mechanisms has unique characteristics both in terms of reasons for the missing data, and the implications of the specific type of missingness. However, regardless of the reason for the missing values, the fact that a measurement is missing is a complication for any algorithm that analyzes the data.

In Missing Completely At Random (MCAR) class, the missing values are distributed completely randomly among all observations in the data set. They are not associated with any other values present in the data, or with themselves. The probability of an absence of value for the X attribute is independent of the other value of the variable attribute Y or the X itself, for example when survey participants accidentally skipped questions. In Missing At Random (MAR) class, the missingness is related only to another variable in the model. The probability of a missing value for the X attribute depends on a different value of the Y attribute variable, but not on the X attribute variable itself. For example, well educated people are less likely to reveal their income than those with lower education. The third class is Non–Ignorable (NI).

NI means that the missingness of the data is not random and the missing data mechanism is related to the missing values. It commonly occurs when people do not want to reveal something very personal or unpopular about themselves, e.g. in mental health survey people who have been diagnosed as depressed are less likely than others to report their mental status.

A key distinction is whether the mechanism is ignorable (i.e., MCAR or MAR) or non–ignorable. Various approaches for handling ignorable missing data have been developed. Non–ignorable missing data are more challenging and require a different approach. In general the methods for treatment methods of ignorable missing data can be divided into following categories [7]: (a) procedures based on completely recorded units, (b) imputation–based procedures, and (c) likelihood–based procedures.

Listwise deletion is an example of procedure based on completely recorded units. It is one of the easiest and very often applied methods to deal with missing values. In this method only full data records are taken into account. In the case when any of the attributes is unknown, the record is excluded from further calculations. It should be noted that any elimination of data affects the loss of significant data, even though the absence occurs even in one attribute. In the case when the ratio of the number of missing items to the number of records in the whole set is large, it may result in the removal of a larger number of samples from all data.

In imputation–based methods the missing values are filled in and the resultant completed data are analyzed by standard methods. Commonly used procedures for imputation include *replacing with random value*, *mean imputation*, and *hot–deck imputation*. *Replacing with random value* consists in filling the lack of value with a randomly chosen value from all values of a given attribute, given in the remaining samples. *Mean and Class–Mean Imputation* consist of replacing the missing data for a given feature (attribute) by the mean of all known values of that attribute in the whole data set or class respectively to which the instance with missing attribute belongs. *Hot Deck Imputation* looks for the most similar case in a data set and fills the missingness with value taken from the other record. As a measure of similarity the Euclidean distance or Manhattan metric can be used. Also *K nearest neighbors* method can be used to determine similar records.

Likelihood–based methods are more robust than the imputation methods described as they have good statistical proper ties. The most common methods employed are:
Expectation–Maximization algorithm (EM) [6] uses the fact that the missing data contain relevant information to be used in the estimation of the parameter of interest. In addition, the estimate of the parameter also helps in finding likely values of the missing data. The EM algorithm is an iterative procedure, which aims to estimate

the missing values and consists of two steps in each iteration, the Expectation step (E–step) and the Maximization step (M–step). During the E–step the distribution of the missing values based on the known values for the observed data and the current estimate of the parameters is found. In M–step it substitutes the expected values (typically means and covariances) for the missing data obtained from the E–step and then maximizes the likelihood function as if no data were missing to obtain new parameter estimates.
Raw Maximum Likelihood method [1,2] uses all of the available information about the observed data, including means and variances for each available covariate to generate estimates of the missing values using maximum-likelihood. Raw maximum likelihood method only produces variances and means for the covariates that have been measured and the statistical package then uses these as imputes for further analyses. This approach is similar to the EM algorithm, except that raw maximum likelihood has no E–step.

4. Data sets and Models

Accurate analysis and understanding of the data greatly facilitates subsequent analysis and interpretation. Before starting the research, it is worth looking at the collection and checking if it is suitable for a specific type of research. For the purpose of this work, the UCI Machine Learning Repository [13] has been searched and two data set containing no missing values were chosen: *Car Evaluation* [14] and *Nursery* [15]. Then, the probabilistic models were constructed. The graphical structure of a Bayesian network represent a set of domain variables and relationships among them. Therefore, constructing the qualitative part of a network should first focus on identification of variables of interest and then on specification of relationships between them.

Car Evaluation data set contains 1728 records and 7 attributes (the last attribute is a decision class): *buying* (price of a given car), *maint* (possible maintenance costs), *doors* (number of doors), *persons* (number of person who can travel in a given car), *lug_boot* (luggage capacity), and *safety* (level of a given car's safety) [3]. On the basis of attributes concerning each of the cars, it is possible to determine to which decision class the auto data can be assigned. The data set does not contain not many records, but it covers all cases. However, the distribution of decision class is very asymmetrical: 70% of records belong to class *unacc*, about 22% to class *acc*, and classes *good* and *vgood* contain about 4% of records each, while the distributions of each attribute in the data set are uniform. It is worth mentioning that Car Evaluation

data set was derived from a simple hierarchical decision model originally developed for the demonstration of DEX [4]. Besides the six input variables, the model included three intermediate concepts: *price*, *tech*, *comfort*. However, the Car Evaluation data set in final form contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes.

Based on this data set, a Bayesian network scheme was created. The nodes representing random variables are attribute of this set. The decision node is a random variable that is a decision class. The states that each node can receive are individual values from each attribute. Designing the Bayesian network for this data set authors followed the example model that can be found in [3], taking into account the accessibility of data. Figure 1 presents the BN used for the further experiments.

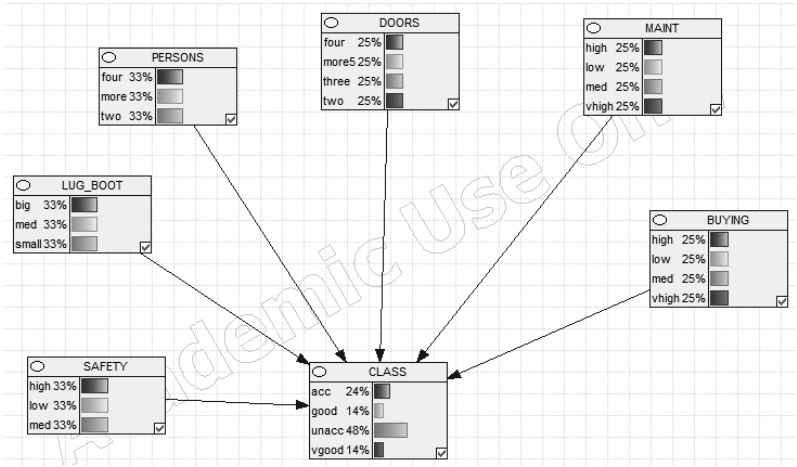

Fig. 1. A Bayesian network model of *Car Evaluation* data set.

Nursery data set contains 12,960 records consisting of 8 attributes plus the attribute of the decision class. The database presents information on the application of the child to kindergartens. The collection was created within a few years in the 1980s in Ljubljana, when many requests for admission were rejected. In the original research, the data set was used in the machine learning HINT evaluation (Hierarchy INduction Tool), which was able to completely recreate the original hierarchical model. The final decision depends on several factors of parental employment and financial situation, family structure or health status of the child. The attributes in data set are

as follows: *parents* (parents' occupation), *has_nurs* (level of childcare), *form* (family structure), *children* (number of children in a family), *housing* (family's living conditions), *finance* (family's financial conditions), *social* (social condition of a family), *health* (child's health condition) [9]. Like the previous data set, records in the distribution of decision–making class is asymmetrical, while the distributions of all attributes are uniform.

Designing the Bayesian network for this data set authors followed the example model that can be found in [11], taking into account the accessibility of data. Figure 2 presents the Bayesian network created on the basis of *Nursery* data set.

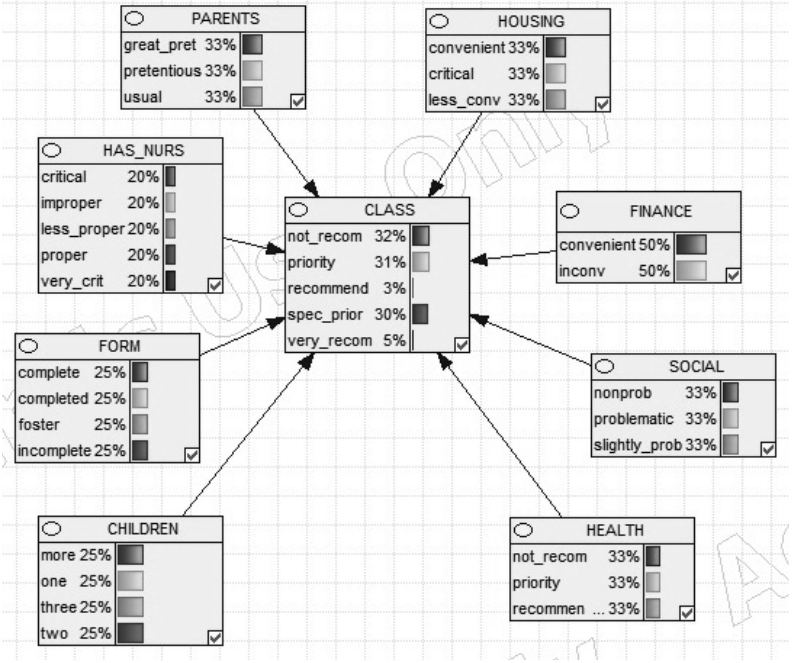

Fig. 2. A Bayesian network model of *Nursery* data set.

5. Experiments

The empirical part of the paper was performed using SMILE, an inference engine, and GeNIe Modeler, a development environment for reasoning in graphical probabilistic models, both developed at BayesFusion LLC, and available at [12].

5.1 Car Evaluation Experiment

The first experiment was conducted on the *Car Evaluation* data set. Due to the fact that this data set contained only nearly 1,700 records, the following methodology was used. Based on the original data, the parameters of a Bayesian network were learned. Next, using that network, two new data sets were generated: a training set (consisting of 10,000 records) and a test set (500 records). In the next step, on the basis of the training set six data sets were created containing 5%, 10%, 15%, 20%, 25%, and 30% of missing values. The missing values were generated randomly, therefore they were of the MCAR type. Data sets with an appropriate level of missing data have been saved. Saving files was an important step because each method must have been tested on a set with the same missing values. Further steps of the experiment consisted in filling the missing values using each of the methods described in Section 3., learning network parameters and performing classification using a test set. This procedure was performed ten times, and then the results of each experiment for each method were averaged. The averaged values are shown in Figure 3 and Table 1.

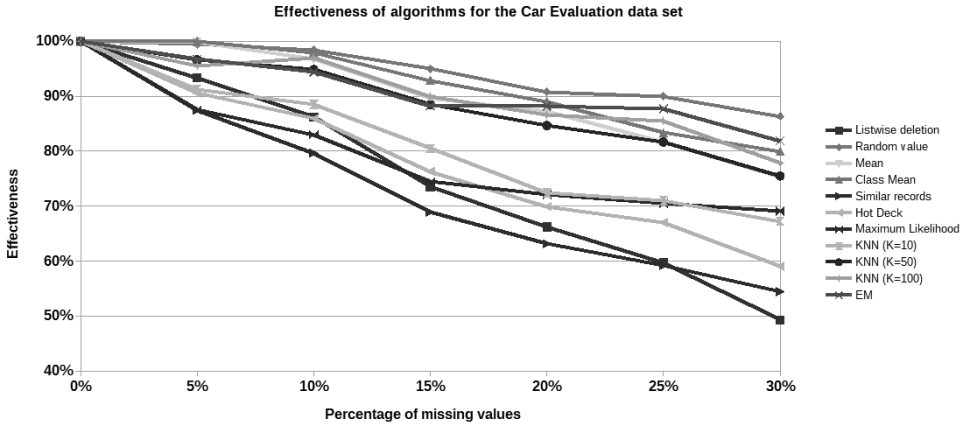

Fig. 3. Figure caption

As mentioned before, training sets used in experiment were generated on the basis of original data set. At first, the model's parameters were learned using data set without missing values and tested on a set of 500 elements. The network's accuracy was about 56%. Clearly, this is not high result – the possible explanation could be the fact that in the original data set the sizes of particular classes were highly un-

Table 1. Results for Car Evaluation data set

	Listwise deletion	Random value	Mean	Class Mean	Similar records	Hot Deck	ML	KNN (*k=10*)	KNN (*k=50*)	KNN (*k=100*)	EM
0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5%	93.36%	99.46%	99.86%	100.00%	87.46%	90.56%	87.49%	91.20%	96.65%	95.56%	96.76%
10%	86.21%	98.36%	96.79%	97.93%	79.64%	85.91%	82.99%	88.50%	94.89%	97.01%	94.40%
15%	73.57%	95.00%	89.57%	92.89%	68.86%	76.30%	74.43%	80.49%	88.48%	89.95%	88.25%
20%	66.21%	90.82%	87.22%	89.07%	63.14%	69.92%	72.12%	72.45%	84.67%	86.58%	88.23%
25%	59.72%	90.00%	81.64%	83.36%	59.21%	67.01%	70.50%	70.97%	81.67%	85.56%	87.69%
30%	49.29%	86.36%	75.25%	79.96%	54.46%	59.03%	69.05%	67.24%	75.49%	77.80%	81.90%

ML means Maximum Likelihood

even. The result obtained with the set with no missing values was a reference point for the results of the methods handling missing data, and the results obtained in the experiment were compared to it (it was treated as 100% effectiveness).

It is noticeable that the methods were divided into two groups due to their effectiveness. The less successful were the methods: listwise deletion, similar records, Hot Deck and KNN for *k=10*. The better–performing methods included: filling with random value and with mean (both global and class), KNN for *k=50* and *k=100*, as well as the EM algorithm.

In case of data set with up to 10% missing values the best algorithm was filling with class mean imputation. In the case of data sets containing more than 10% of missing values, the algorithm filling with a randomly selected value from a given attribute and the EM algorithm become the most beneficial. Algorithm giving the poorest results in the data sets containing up to 20% missing algorithm was one based on filling with similar records. When the level of missingness increased by more than 20%, the worst method was listwise deletion.

5.2 Nursery Experiment

In the case of the *Nursery* data set, the experiment was performed in a bit different way than in case the *Car Evaluation* data set. The *Car Evaluation* data set consisted of a small number of records, which is why the network learning methodology was used on the original data set and a training set consisting of 10,000 records was generated. In the case of the *Nursery* data set, there was no such need – the size of the data set equaled roughly 13,000 records. The further steps of the experiment were identical. The original data set was divided into a test set (size 1000 records) and a training set. Having a training set, again, missing values were generated at the

level of 5%, 10%, 15%, 20%, 25% and 30% respectively. Each of the created data sets was saved separately. Next stages consisted in filling the missing values by using particular methods, learning created Bayesian network and verifying on the basis of the test set. Similarly to the first experiment, this procedure was repeated ten times, and the average was drawn from the results (see Figure 4 and Table 2).

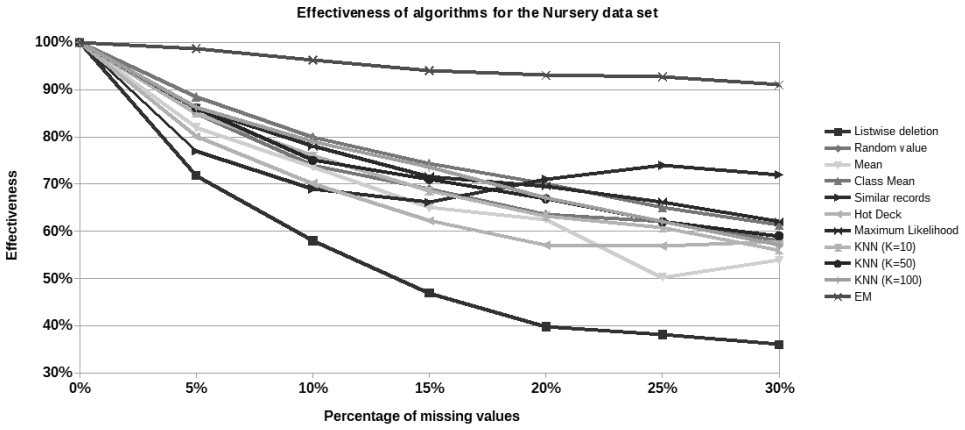

Fig. 4. Figure caption

Table 2. Results for Nursery data set

	Listwise deletion	Random value	Mean	Class Mean	Similar records	Hot Deck	ML	KNN (K=10)	KNN (K=50)	KNN (K=100)	EM
0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5%	71.78%	85.00%	82.00%	88.45%	77.01%	80.01%	86.02%	85.00%	86.11%	86.19%	98.65%
10%	58.00%	73.90%	73.68%	80.00%	68.96%	70.07%	78.00%	75.96%	75.00%	79.06%	96.25%
15%	46.90%	69.01%	65.00%	74.35%	66.08%	62.19%	71.54%	68.52%	71.00%	73.56%	94.00%
20%	39.80%	63.60%	62.50%	70.00%	71.01%	57.03%	69.59%	63.22%	66.88%	67.05%	93.04%
25%	38.10%	62.00%	50.26%	65.09%	73.98%	56.87%	66.25%	60.74%	62.00%	62.09%	92.69%
30%	36.00%	58.00%	53.86%	61.25%	72.00%	57.77%	62.03%	56.00%	59.00%	57.01%	91.00%

ML means Maximum Likelihood

The result of the classification on the set with no missing values was roughly 90%. And again, as in the previous experiment, it was the reference point to which

the classification results were compared on the sets in which the missing values were supplemented.

It can be unequivocally stated that the algorithm of missing value replenishment, giving the best results, was the EM algorithm. At every level of missingness it outclassed all other methods. It can be also noticed that the listwise deletion algorithm was the least effective algorithm in all cases. The difference in the accuracy of the classification between these two methods was roughly 25% for a set of 5% missing data and up almost 50% for data sets containing more than 20% missing data. Unlike in the previous experiment, all the imputation methods yielded similar results.

6. Conclusions

The conducted research confirmed the belief that there is no one universal method of handling the missing values. It all depends on the type of data, attributes with missing values, relationships between attributes, the number of records in the data set, the number of records with missingness and many other factors. Therefore, it is very important to carefully analyze the data on which the tests will be carried out. In order to choose the most effective method, it is worth conducting an experiment using several or even a dozen or so methods of dealing with the missing values in the collections. On the basis of such an experiment, the appropriate method should be chosen for the given set.

The conclusion that can be drawn from both experiments is that the most effective method was the EM algorithm. In the case of the *Car Evaluation* data set, the results of this algorithm were very similar to other methods. However, in the case of the *Nursery* data set, the EM algorithm outclassed the other methods. Very good compared to other algorithms, in both cases there were two methods: class mean imputation method and K nearest neighbors method. In both experiments, the poorest results were obtained by the listwise deletion method. This is not a surprising result. In the case of increasing the level of missing values in the set, the number of samples with at least one missing value increases. Since such data are deleted (or ignored) the greater part of the set is not used, a great amount of information is lost. The network is learned from data that often does not contain many relevant information and does not take into account most cases.

References

[1] James L. Arbuckle, Full information estimation in the presence of incomplete data, Marcoulides, G.A. and Schumacker, R.E. (eds.), Advanced Structural Equation Modeling: Issues and Techniques. Mahwah, NJ: Lawrence Erlbaum Associates, 1996.

[2] Paul D. Allison, Missing data techniques for structural equation models, Journal of Abnormal Psychology 112 (2003), pp. 545–557.

[3] Marko Bohanec and Rajkovic Vladislav, Knowledge acquisition and explanation for multi-attribute decision making, 8th Intl Workshop on Expert Systems and their Applications, pp. 59–78, 1988.

[4] Marko Bohanec and Rajkovic Vladislav, Expert system for decision making, Sistemica 1(1), pp. 145–157, 1990

[5] Nir Friedman, Dan Geiger and Moises Goldszmidt, Bayesian network classifiers, Machine Learning 29 (1997), 131–163.

[6] Steffen L. Lauritzen, The EM Algorithm for Graphical Association Models with Missing Data, Computational Statistics and Data Analysis, 19:191–201, February 1995.

[7] Roderick J. A. Little and Donald B. Rubin, Statistical Analysis with Missing Data, Second edition, Chichester: Wiley, 2002.

[8] Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann PUBLISHERs, Inc., San Mateo, CA, 1988..

[9] Olave, Manuel, Vladislav Rajkovic, and Marko Bohanec, An application for admission in public school systems, Expert Systems in Public Administration 1 (1989): 145-160.

[10] Peter Spirtes, Clark Glymour, and Richard Scheines, Causation Prediction and Search, Springer-Verlag, New York, 1993.

[11] Blaz Zupan and Marko Bohanec and Ivan Bratko and Janez Demsar Machine Learning by Function Decomposition, ICML, 1997

[12] BayesFusion, LLC, [https://www.bayesfusion.com/], Accessed 15-03-2017.

[13] UCI Repository of machine learning databases, [http://archive.ics.uci.edu/ml/datasets.html], Accessed 05-04-2017,

[14] Marko Bohanec, Database Car Evaluation. June 1997, [http://archive.ics.uci.edu/ml/datasets/Car+Evaluation], Accessed 05-04-2017.

[15] Vladislav Rajkovic, Database Nursery, June 1997, [https://archive.ics.uci.edu/ml/datasets/Nursery], Accessed 01-06-2017.

PORÓWNANIE METOD UZUPEŁNIANIA DANYCH BRAKUJĄCYCH W UCZENIU MODELI PROBABILISTYCZNYCH

Streszczenie Brakujące dane są częstym problemem w analizie statystycznej, a większość baz danych zawiera brakujące wartości niektórych z ich atrybutów. Brakujące dane mogą pojawiać się z wielu powodów. Jednak bez względu na przyczynę brakujących wartości nawet ich niewielki procent może spowodować poważne problemy z analizą, zmniejszając siłę statystyczną badania i prowadząc do wyciągnięcia błędnych wniosków. W artykule przedstawiono wyniki uzupełniania danych brakujących w uczeniu modeli probabilistycznych. Dwa zestawy danych pobrane z repozytorium uczenia maszynowego UCI posłużyły do wytrenowania ilościowej części sieci bayesowskich. Aby zapewnić możliwość porównania wybrane zbiory danych nie zawierały żadnych brakujących wartości. Dla każdego modelu zbiory danych z różnymi poziomami brakujących wartości zostały sztucznie wygenerowane. Głównym celem tego artykułu było zbadanie, czy braki w obserwacjach mają wpływ na niezawodność modelu. Dokładność została zdefiniowana jako procent poprawnie zaklasyfikowanych rekordów i została porównana z wynikami uzyskanymi w zbiorze danych niezawierającym brakujących wartości.

Słowa kluczowe: dane brakujące, modele probablistyczne, sieci Bayesa, klasyfikacja

Artykuł częściowo zrealizowano w ramach pracy badawczej S/WI/2/2018.