

# Techniques for Dealing with Missing Data in Knowledge Discovery Tasks

Matteo Magnani

matteo.magnani@cs.unibo.it

University of Bologna, department of Computer Science

Mura Anteo Zamboni 7,

40127 Bologna - ITALY

## 1 Introduction

Information plays a very important role in our life. Advances in many research fields depend on the ability of discovering knowledge in very large data bases. A lot of businesses base their success on the availability of marketing information. This kind of data is usually big, and not always easy to manage.

Scientists from different research areas have developed methods to analyze huge amounts of data and to extract useful information. These methods may extract different kinds of knowledge, depending on the data and on user requirements. In particular, one important knowledge discovery task is supervised learning. Today, there exist many methods to build classifiers, belonging to different fields, such as artificial intelligence, soft computing, statistics.

Unfortunately, traditional methods usually cannot deal directly with real-world data, because of missing or wrong items. This report concerns the former problem: the unavailability of some values. The majority of interesting data bases is incomplete, i.e., one or more values are missing inside some records, or some records are missing at all.

There exist many techniques to manage data with missing items, but no one is absolutely better than the others. Different situations require different solutions. As Allison says, “the only really good solution to the missing data problem is not to have any” [1].

This report reviews the main missing data techniques (MDTs), trying to highlight their advantages and disadvantages. Next section introduces some terminology and presents a taxonomy of MDTs. Section 3 describes these methods more in detail. Finally, some conclusions are reported.

Patient ID	Age	Test Result
pat1	23	1453
pat2	23	1354
pat3	23	2134
pat4	23	2043
pat5	75	1324
pat6	75	1324
pat7	75	2054
pat8	75	2056

Table 1: This complete example will be used to show different missing mechanisms. It represents a list of patients, with their age and the result of a medical test. *Patient ID* is a “bureaucratic” variable, which is not used during data analysis. *Age* is an independent variable, while *Test Result* is the dependent variable.

## 2 Taxonomies

### 2.1 Missing data mechanisms

The effectiveness of a MDT depends closely on the missing mechanism. For example, if we know why a value is missing, we can use this information to guess it. If we do not have such an information, we should hope that the missing mechanism is *ignorable*, so that we can apply methods which assume it is not relevant.

Statisticians have identified three classes of missing data. The easiest situation is when data are *missing completely at random (MCAR)*. This means that the fact that a value is missing is unrelated to its value or to the value of other variables. When data are MCAR, the probability that a variable is missing is the same for every record. If the probability that a value is missing depends only on the value of other variables, we say that it is *missing at random (MAR)*. In term of probabilities, if we have a variable  $Y$  with missing values, and another variable  $X$ , we may write that data are MAR if  $Pr(Y_{missing}|Y, X) = Pr(Y_{missing}|X)$ . If the missingness depends on the missing value, data are *not missing at random (NMAR)*, and this is a problem for many statistical MDTs. This happens, for instance, when we collect data with a sensor which is not able to detect values over a particular threshold. Tables 1, 2, 3 and 4 show an example of each missing mechanism.

### 2.2 Missing data techniques

If we want to build a classifier from a set of incomplete records, we may do it in two ways. Either we build a complete data set, then we apply traditional learning methods; or we use a procedure which can internally manage missing data. We are more interested in the first strategy, for the following reasons:

- Data collectors may have knowledge of missing data mechanisms, which can be used while building the complete data set.

Patient ID	Age	Test Result
pat1	23	1453
pat2	23	<i>null</i>
pat3	23	2134
pat4	23	<i>null</i>
pat5	75	1324
pat6	75	<i>null</i>
pat7	75	2054
pat8	75	<i>null</i>

Table 2: MCAR. Assume that the test is very expensive. Researchers can decide to perform the test only on some patients, randomly chosen. During analysis, they can consider only those records with a test result. Notice that  $Pr(\text{TestResult missing} | \text{Age} = 23) = Pr(\text{TestResult missing} | \text{Age} = 75)$  and  $Pr(\text{TestResult missing} | \text{TestResult}) = Pr(\text{TestResult missing})$ .

Patient ID	Age	Test Result
pat1	23	1453
pat2	23	<i>null</i>
pat3	23	2134
pat4	23	<i>null</i>
pat5	75	1324
pat6	75	1324
pat7	75	2054
pat8	75	2056

Table 3: MAR. In this example, the test is performed mainly on old people. Therefore, the fact that a test result is missing depends on the *age* variable. In particular,  $Pr(\text{TestResult missing} | \text{Age} = 23) = 0.5$ , while  $Pr(\text{TestResult missing} | \text{Age} = 75) = 0.0$ .

Patient ID	Age	Test Result
pat1	23	1453
pat2	23	1354
pat3	23	<i>null</i>
pat4	23	<i>null</i>
pat5	75	1324
pat6	75	1324
pat7	75	<i>null</i>
pat8	75	<i>null</i>

Table 4: NMAR. In this case the device used to perform the test does not work on high values. This means that the missingness of a test result depends on the missing value. More formally,  $Pr(\text{TestResult missing} | \text{TestResult} < 2000) = 0.0$ , while  $Pr(\text{TestResult missing} | \text{TestResult} > 2000) = 1.0$ . Notice that without external information this data is not statistically different from Table 2, because  $Pr(\text{TestResult missing} | \text{Age}) = Pr(\text{TestResult missing})$ . If we do not know the underlying dependency, which is usually the case, it may be very difficult to analyze this data.

- With a complete data set, every existing learning technique can be used.
- A common starting point allows comparisons of different learners.

Anyway, we present also a classifier which directly manages missing data, without preprocessing the input table: C4.5 [3]. This algorithm builds a decision tree which can be then used to classify new records, and it has shown to be very efficient if compared with other methods [4].

The easiest way to obtain a complete data set from an incomplete one is to erase missing items. These *conventional* methods are used because of their simplicity, and are usually the default option in statistical packages. In this report we present two variations of this strategy: *listwise deletion* and *pairwise deletion*.

If we do not want to lose data and perhaps information, we may try to guess missing items. This process is generally called *imputation*. In fact, usually missing values depend on other values, and if we find a correlation between two variables, we may use it to impute missing items.

Imputations may be *deterministic* or *random (stochastic)*. In the first case, imputations are determined by the incomplete data table, and are the same if the method is applied again. This is useful if we want to compare the work of different researchers, and if we want to repeat simulations. In the second case, imputations are randomly drawn.

We may use different kinds of information to impute missing data. We may analyze the behavior of all the other records on the missing variable (global imputation based on missing attribute). We may try to find a relationship between other variables and the missing one in all the other records, and use this relationship to impute the missing value (global imputation based on non-missing attributes). Or we may look for similar

records to see how they behave on the missing attribute (local imputation).

If we are interested in guessing statistics, i.e., summary values of the sample, we may also use the EM algorithm [5], based on the maximization of a likelihood function. This algorithm directly estimates statistics, without imputing values - even if it can be used also for imputation.

In next section, various techniques belonging to these classes are presented.

## 3 Techniques

### 3.1 Conventional methods

*Listwise deletion* is the easiest way to obtain a complete data set. Only complete records are retained. This method has two main drawbacks. Firstly, if we apply it in real situations we may lose a lot of data. For example, if we have a table with 40 features and a missing probability of 0.05% on every variable, the probability that a record is not missing is 0.13%<sup>1</sup>. This means that if we have collected 1000 records, we can use only 130 of them. Moreover, real situations may be even worse: in [6] the data base used to compare different MDTs has no complete records, so this method cannot be applied. When we want to build a classifier from a data table, records are usually organized into overlapping clusters, called concepts, or granules. In a rule-based system, every concept contributes to the discovered rules with some information which in general is not provided by other concepts. For this reason, losing all the records in a concept may cause the loss of some rules, and losing a part of the records may bias support and significance of the rules. The second problem of listwise deletion is that, if we want to calculate sample statistics, data must be MCAR. For instance, if data are MAR, the mean of non-missing variables can be biased. In Table 1 the mean of the ages is 49; if we only consider complete records of Table 3, it is 64.6. This happens because records with a low age are more likely to be incomplete.

We may also decide to discard all variables (columns) with missing data, but this is usually really a bad choice, because of evident reasons.

*Pairwise deletion* is a variant of listwise deletion, which uses an incomplete record during analysis only when the needed variable is not missing. For example, to calculate the means of the variables in Table 2, we can use records *pat1*, *pat3*, *pat5* and *pat7* for the mean of *TestResult*, and all the records for the mean of *Age*. This method has the advantage of using all available data, but is more complicated than listwise deletion, and sometimes it cannot be used (for example, for some data sets we cannot perform a linear regression [1]). In general, this is used less than listwise deletion, because either we want to deal easily with missing data (in this case the preferred solution is listwise deletion), or we decide to spend some time to tackle the problem, in which case we may use a more sophisticated method.

---

<sup>1</sup>If the missingness of a value does not depend on the missingness of other values

### 3.2 Global imputation based on missing attribute

If we look at the other values taken by a variable with a missing item, and we find that some of them are more frequent than the others, we may decide to use this fact to assign the most frequent value to a missing one. We may also choose a measure of central tendency, such as mean, median, mode, trimean or trimmed mean, and use it to fill the holes. In particular, *mean imputation* and *most common attribute value (mode) imputation* have been compared against other MDTs ([7], [4], [8]). They are usually regarded as bad methods by statisticians, because the standard deviation of the sample is underestimated even when data are MCAR. For example, look at Table 2. The real mean and standard deviations are  $M_r = 1717.75$  and  $S_r = 306.372$ . If we apply a mean imputation, we obtain  $M_{imp} = 1741.25$ , which is very similar to the real sample mean, while  $S_{imp} = 252.302$ . This happens because we add all values on the center of the sample, diminishing the importance of values on the tails. Obviously, this result can be adjusted, but in this case we do not obtain a reusable complete data, which was one of the motivations for using imputation. Moreover, it cannot be used by a rule-based system, because we are interested in contingent values, not in summaries. For instance, the mean of the second attribute in  $(\langle \text{Mark}, 10/10 \rangle, \langle \text{John}, 3/10 \rangle)$  and in  $(\langle \text{Mark}, 3/10 \rangle, \langle \text{John}, 10/10 \rangle)$  is the same, but the rules induced by a learner are very different. The *non-deterministic version* of the mean imputation method, which adds a random disturbance to the mean, works better from a statistical point of view, but it is still not satisfactory if we want to extract rules from the data.

Another variation of this approach consists in *assigning all possible values* to a missing item, i.e., all the values that the missing variable takes in other records. This produces an inconsistent data table, but inconsistencies may be managed subsequently. This idea is apparently interesting, but in general imputation should not add information to the data. This is not always easy to avoid, but this method does it systematically. Moreover, as shown in [4], it is computationally unfeasible.

### 3.3 Global imputation based on non-missing attributes

If there are correlations between missing and non-missing variables, we may learn these relationships and use them to predict missing values.

One such strategy is *imputation by regression*. Missing attributes are treated as dependent variables, and a regression is performed to impute missing values. Linear regression and logistic regression are typical choices.

This kind of strategy has two kinds of problems. The first one depends on the use of regression. In fact, usually values are missing on important variables - which are the most predictive. Therefore, we must evaluate different regression equations, depending on the available data. Moreover, regression is based on the assumption that the chosen model fits well the data. If we perform a linear regression, but the relationship is not linear, we may obtain wrong results. The second class of problems depends on the fact that we impute only one value for every missing item. This produces a complete data set, which does not represent correctly the uncertainty of imputed values.

To solve this second problem, *multiple imputation* [9] has been proposed. It consists in imputing several values for every missing item, producing an array of complete

data sets. These can then be analyzed by standard complete-data techniques, and the results are finally merged. For an overview of multiple imputation, see [10].

### 3.4 Local imputation

*Hot Deck imputation* is a procedure where the imputation comes from other records in the same data, and these are chosen on the basis of the incomplete record [11], [12]. This is an abstract procedure, which has been used in practice in very different ways and is not based on a strong theory. A Hot Deck imputation acts in two steps:

1. Records are subdivided into classes.
2. For every incomplete record, the imputing values are chosen on the basis of the records in the same class.

Changing the implementation of these steps leads to many variations of this method. As an example, we can use Hot Deck imputation to repair Table 3. We group records in two groups, using the complete variable “Age”. Then, when we have a missing value, we read the imputation on the preceding record in the same group. In this case, “pat2” takes the value for “Test result” by “pat1”, while “pat4” takes it by “pat3”.

In [12] the author says that a main attractive feature of Hot Deck procedures is that “no strong model assumptions need to be made in order to estimate the individual values”. In our opinion, Hot Deck procedures make a very strong assumption: that objects can be organized in classes with little variation inside a class. This is different from the assumption of other MDTs, where data are thought as being independent and identically distributed. The interesting point here is that other MDTs and Hot Deck were originally applied mainly to survey data, where the second assumption is more likely to be true. But in current complex tasks of knowledge discovery in data bases (KDD), and in particular when we want to extract a set of rules from a data table, probably the first assumption is verified in many situations. To check this statement, a precise analysis of the model building process should be done.

Recently, some variations of the Hot Deck imputations have been developed. In particular, [13], [14] and [15].

### 3.5 Parameter estimation

The EM (Expectation Maximization) [5] is a well-known algorithm to estimate the parameters of the probability density function of an incomplete sample.

It is a maximum likelihood procedure. When we have a sample (set of records), and we “suspect” that it has been drawn by a given distribution, it tries to find the parameters of the distribution that would have more likely produced the observed sample.

The main drawbacks of EM are that it is an iterative algorithm, which can take a long CPU time before converging, and to use it we have to specify the sample distribution. Unfortunately, in many Data Mining problems we do not know the probability density function in advance.

### 3.6 Direct management of missing data

Case deletion and imputation are not the only ways to manage missing data. A well-known approach to incomplete supervised classification is tree-based classification. From an incomplete data table, a decision tree is built, which is also able to classify unseen records with missing values. For a short overview of tree-based classification techniques see [16] and [17].

In general, we may build a decision tree following this procedure:

1. Select a test which can partition records in disjoint sets.
2. Partition the current set of records (initially, the training set).
3. For every partition, repeat this procedure.

The most famous algorithm of this kind is C4.5 [3], which has shown to be very effective [4]. It chooses the test which maximizes the *gain ratio*, a measure that expresses the improvement of classification power of the tree after the partition. This technique can also be used in presence of missing values.

1. The test is chosen using a measure similar to the *gain ratio*, adjusted by a factor that depends on the number of records complete on the tested attribute.
2. Every incomplete record is then assigned to all partitions, with a probability which depends on the size of the partition.
3. When an unseen record must be classified, but the testing attribute is unknown, all possible paths are examined, leading to more than one possible class.

The efficiency of C4.5 can be increased by the use of committee learning techniques, a sort of multiple imputation applied to decision trees. In [18] four of them have been compared, when applied to C4.5.

## 4 Final remarks

There exist many techniques to manage missing data. In particular, we are interested in preprocessing methods, which can be used before any further analysis.

Some of them are very easy to apply, but have too many drawbacks, which limit their use to uninteresting situations. This is the case for listwise deletion, and for naive imputation techniques, such as deterministic mode or mean imputation.

Statistical “intensive” methods have been mainly developed to manage survey data. They have proven to be very effective in many situations, in particular the EM algorithm and multiple imputation. The main problem of these techniques is the need for strong model assumptions, which are usually difficult to justify in KDD processes.

Hot Deck procedures do not assume particular distributions, so it is appealing to see if they can be applied to the KDD preprocessing task. These methods have also another interesting feature: for some missing values they can find no imputation information. This is very important, because they can be used together with other strategies (Cold



Deck, or statistical techniques). In this way, every MDT can repair the portion of data for which it is suitable. Recently, some methods have been proposed, which are in fact Hot Deck variations. We can cite various closest-fit strategies [4], k-nearest neighbor imputation [14], non-invasive imputation [15], and an aged but very interesting paper about clustering imputation [19].

Further research is needed in this direction, with the following high level objectives:

**Identify all possible variations of Hot Deck.** As we have already said, Hot Deck is not a precise method, but it is a general strategy. Every step can be implemented in several ways. A complete list of choices can be used as a taxonomy, and to identify some variations which are possibly better than the others.

**Classify recently proposed techniques.** The newly proposed Hot Deck techniques usually present an implementation, without discussing why it has been chosen between all the possible ones. Sometimes, they do not explicitly show their belonging to the Hot Deck philosophy. Performing a classification of these implementations, based on the above taxonomy, can be useful also to compare apparently different methods.

**Analyze Hot Deck assumptions.** Usually, Hot Deck methods are presented as not having *strong* model assumptions. In my opinion, this is correct only if mean *statistical* model, i.e., *distributional* model. But we obviously need some assumptions, because imputation should not add information to the data<sup>2</sup>. To impute a value, we have to look for it in the remaining data. And to claim that it is the right value, we need a model which tells us that this is the case. Therefore, if we want to justify the use of a Hot Deck procedure, we need to define the kind of data where this is reasonable.

## References

- [1] P. D. Allison, *Missing data*. Sage Publications, Inc, 2001.
- [2] J. Schafer, *Analysis of incomplete multivariate data*. Chapman Hall, 1997.
- [3] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [4] J. Grzymala-Busse and M. Hu, “A comparison of several approaches to missing values in data mining,” in *Rough Sets and Current Trends in Computing* (W. Ziarko and Y. Y. Yao, eds.), vol. 2005 of *Lecture Notes in Computer Science*, Springer, 2001.
- [5] D. A. P., L. N. M., and R. D. B., “Maximum-likelihood from incomplete data via the EM algorithm,” *Journal of The Royal Statistical Society Series B*, vol. 39, 1977.

---

<sup>2</sup>If we perform a predictive imputation on a data table, we add information to the table, not to the data. In this case the data is composed by the table, which is experimental knowledge, and the model, as domain knowledge.

- [6] K. Lakshminarayan, S. A. Harp, and T. Samad, "Imputation of missing data in industrial databases," *Applied Intelligence*, vol. 11, pp. 259–275, November 1999.
- [7] J. Scheffer, "Dealing with missing data," *Res. Lett. Inf. Math. Sci.*, vol. 3, pp. 153–160, 2002.
- [8] M. Hu, S. M. Salvucci, and M. P. Cohen, "Evaluation of some popular imputation algorithms," in *Section on Survey Research Methods*, American Statistical Association, 2000.
- [9] D. B. Rubin, "Multiple imputation for nonresponse in surveys." John Wiley and Sons, 1987.
- [10] D. B. Rubin, "An overview of multiple imputation," in *Survey Research Section*, pp. 79–84, American Statistical Association, 1988.
- [11] B. L. Ford, "An overview of hot-deck procedures," in *Incomplete data in sample surveys*, Academic Press, Inc., 1983.
- [12] I. G. Sande, "Hot-deck imputation procedures," in *Incomplete Data in Sample Surveys*, vol. 3, 1983.
- [13] J. W. Grzymala-Busse, W. J. Grzymala-Busse, and L. K. Goodwin, "A comparison of three closest fit approaches to missing attribute values in preterm burth data," *International journal of intelligent systems*, vol. 17, pp. 125–134, 2002.
- [14] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003.
- [15] G. Gediga and I. Düntsch, "Maximum consistency of incomplete data via non-invasive imputation," *Artificial intelligence Review*, vol. 19, pp. 93–107, 2003.
- [16] J. R. Quinlan, "Unknown attribute values in induction," in *Proceedings of the Sixth International Workshop on Machine Learning* (B. Spatz, ed.), pp. 164–168, Morgan Kaufmann, 1989.
- [17] W. Liu, A. White, S. Thompson, and M. Bramer, "Techniques for dealing with missing values in classification," in *International Symposium on Intelligent Data Analysis*, 1997.
- [18] Z. Zheng and B. T. Low, "Classifying unseen cases with many missing values," in *PAKDD* (N. Zhong and L. Zhou, eds.), vol. 1574 of *Lecture Notes in Computer Science*, pp. 370–374, Springer, 1999.
- [19] R. C. T. Lee, J. R. Slagle, and C. T. Mong, "Towards automatic auditing of records," *IEEE Transactions on Software Engineering*, vol. 4, no. 5, pp. 441–448, 1978.