

# A tool to manage low quality datasets

José M. Cadenas      M. Carmen Garrido      Raquel Martínez

Department of Engineering Information and Communications

Faculty of Informatic, University of Murcia

Campus of Espinardo, Murcia, Spain

Email: jcadenas, carmengarrido, raquel.m.e@um.es

**Abstract**—Nowadays in the world of Soft Computing there is a new challenge which consists of working with low quality data. To test the techniques that are designed in this area, there is the need for repositories of low quality datasets. Currently we can find various data mining techniques that are designed to handle some kind of low quality data. But, as far as we get our knowledge, it has not yet designed a software tool focused on the creation/management of low quality datasets that will help us to create repositories to facilitate the testing and comparison of the above techniques. We present in this paper a software tool which can create/manage low quality data. Even if a technique doesn't manage low quality data, the tool permits to transform the low quality data by other data that technique can manage.

**Index Terms**—Soft Computing, low quality data, software tool for Soft Computing

## I. INTRODUCTION

A new challenge in the area of Soft Computing is to deal with the inherent imperfection of the real world information. Although many data mining techniques, [14], are based on the assumption of perfect information, in the reality data are never as good as the engineers would like. Very often, data suffer damage which affect their interpretation. If the data mining techniques do not take into account this imperfection in the information, the models generated are low-quality defective or unnecessarily complex models. Finally, this affects the interpretation and decision-making that we make basing on these data.

There is a wide range of data mining techniques based on different theoretical proposals, [13], [18], [21], [25]. Unfortunately, most of the conventional techniques have not considered the sources of imperfection. As a result, incomplete and imprecise data have usually been discarded or ignored in their learning phases and in their subsequent inference processes. However, these data are inevitable when dealing with real-world applications.

In this situation, we can analyze different alternatives. We could transform low quality data, which a technique can not manage, by other kinds of values trying not to lose too much information in this change. In this alternative, we need a software tool to be able to make this transformation. Another alternative is to modify the technique in order that it can work with low quality data, but in this case the problem is to find public datasets to check the technique, so that we know how a technique behaves when it deals with different levels of imperfection. Besides we could need to introduce expert information expressed by linguistic labels or to discretize

certain values by a fuzzy partition to make interpretable the dataset. In these latter situations, we also need a software tool to be able to create datasets with the desired features.

In the previous paragraph we have discussed some of the reasons why is necessary to have a software tool for this purpose. In version 1.5, we designed a software tool, called NIP, intended to start with the management of low quality datasets. In this work, we present a new version, NiP<sub>2.0</sub>, of that software tool with new functionalities. Moreover, this software tool allows to define different formats for input/output datasets both predefined and custom by user.

This paper is organized as following. In Section II we present different kinds of low quality data that this software tool can manage. In Section III we expose a brief study about some software tools that can process datasets. Also in Section IV we explain the new functionalities of the software tool NiP<sub>2.0</sub>. Finally, in Section V we present an example to show the transforming of a dataset and the conclusions of this paper in Section VI.

## II. LOW QUALITY DATA

Low quality information inevitably appears in realistic domains and situations. Instrument errors or corruption from noise during experiments may give rise to information with incomplete data when measuring a specific attribute. In other cases, the extraction of exact information may be excessively costly or unfeasible. Moreover, it might be useful to complement the available data with additional information from an expert, which is usually elicited by imperfect data (interval data, fuzzy concepts, etc). In most real-world problems, data have a certain degree of imprecision. Due to this situation, datasets from which to extract knowledge or model systems will contain low quality information. In this section we develop a brief study about some kind of low quality data and how techniques handle with them.

### A. Missing Values

An important factor to consider when working with data mining techniques is the treatment that they perform on missing values. Namely:

- Allow missing values in the dataset when the technique is robust to the existence of such values.
- Remove attributes with missing values of the dataset.
- Remove samples with missing values of the dataset.

- Replace missing values manually (if not many) or automatically by a value that preserves the mean or variance (global or classes/groups), in the case of numeric values, or by the mode in the case of nominal values. Another way to estimate a value is to predict it from other examples (missing value imputation) using any predictive data mining technique.
- Replace missing values by an interval that covers the entire domain of the corresponding attributes.

### B. Noise

Noise is a broad term that has been interpreted in different ways. For example, it has been defined as a random error in a measured attribute. Another definition states that noise is any property in the detected pattern that is not due to real underlying model but the randomness in the world or the sensors [15]. In these definitions, the noise is considered an error occurred randomly. The noise in the data can happen for several reasons:

- First, due to problems with measuring instruments or equipment.
- Second, it is due to the fact that large datasets are obtained by automated methods.

Depending on the data mining technique with which we work, it maybe:

- If it is possible to detect the values with noise, the treatment may be similar to the case of missing values.
- It is sometimes possible to know the errors of the measuring instrument (mean and standard deviation). This allows us to incorporate this information in the set of values of attribute with noise by means of some transformation.

### C. Uncertainty

Other two kinds of low quality information can be found in our environment and our thinking process: imprecision and uncertainty. Imprecision and uncertainty can be considered as two complementary aspects of imperfect information [4], [8], [11]. From a practical viewpoint, an item of information can be represented as a four-tuple (attribute, object, value, confidence). The attribute is a function which assigns a value (or a set of values) to the object. The value is a subset of the reference domain associated to the attribute. Confidence indicates the likelihood of the item of information. In this context, imprecision is related to the value of the item of information, while uncertainty is related to the confidence in the item. Thus, an item of information will be precise when its value cannot be subdivided. Otherwise, we will talk about imprecision. Furthermore, when there are no crisp constraints on the set of values that an imprecise item can take, we will talk about fuzzy imprecision. On the other hand, uncertainty is a property of belief. We say that we are certain of an event if we assign it a maximum belief value. We can define uncertainty as the absence of certainty, and this may arise from the randomness of some experimenting (objective uncertainty),

or from subjective judgments by human reasoning (subjective uncertainty).

In [12] the concepts of imprecision and uncertainty are described in terms of stochastic and epistemic uncertainties. Stochastic uncertainty arises from random variability related to natural processes. Epistemic uncertainty arises from the incomplete/imprecise nature of available information. While stochastic uncertainty is adequately addressed using classical probability theory, several uncertainty theories have been developed in order to explicitly handle incomplete/imprecise information that basically are convex probability sets, random sets and possibility theory.

Some simple representations of uncertain/imprecise information based on intervals and its generalization are:

- Using an interval  $[a, b]$ , so we assume that the attribute value is within it.
- Using a set or fuzzy interval that assigns a degree of possibility between 0 and 1 to each value of the interval as a possible attribute value.

Depending on carrying out the treatment of uncertainty/imprecision in different data mining techniques, we can:

- Allow those values in the dataset when the technique is robust to the existence of such kind of values.
- Remove attributes with uncertainty/imprecision of the dataset.
- Remove samples with uncertainty/imprecision of the dataset.
- Replace/impute values with uncertainty/imprecision.

## III. SOFTWARE TO PRE-PROCESS DATA

In this section we are going to revise the main characteristics of some software tools of data mining and learning from the viewpoint of pre-processing of data. We only focus on exposing the main features of free software, although some private software tools which perform pre-processing of data are DataPreparator [24], Simulink Design Optimization [16], etc.

- Sodas2 [10]: is a tool that supports symbolic analysis of data. It tries to generalize the data mining and statistical process in a higher level, described by symbolic data. In this way, data are transformed in more manageable and more complex data. The running of Sodas2 allows to build a set of symbolic data that summarizes the information of initial dataset, and after this, to perform the symbolic analysis. The use of symbolic data allows to introduce different types of low quality data: multi-valued attributes, intervals and multi-valued attributes with weights. These types of attributes represent other types of low quality data, such as fuzzy, imprecise and uncertain data. However, Sodas2 does not allow: a) the direct use of fuzzy technology; b) the introduction of missing values and noise; c) the modification of data to introduce any kind of imperfection.

TABLE I  
COMPARATIVE OF CHARACTERISTICS OF SOFTWARE TOOL WITH REGARD TO DATA MANAGEMENT

(Y: yes; N: no; I: Intermediate)	Sodas2	WEKA	KEEL	Rapid Miner	MiningMart	NiP <sub>2.0</sub>
Format: Standard	I	Y	Y	Y	I	Y
Custom	N	N	N	N	N	Y
Low quality data: missing value	N	Y	Y	I	N	Y
interval values	N	N	N	N	N	Y
fuzzy values	N	N	N	N	N	Y
crisp partition	Y	Y	Y	Y	Y	Y
fuzzy partition	N	N	N	N	N	Y
Imputation/Replacement: missing value	Y	Y	Y	I	I	I
interval value	N	N	N	N	N	I
fuzzy value	N	N	N	N	N	I
Noise in Attributes: Nominal	N	Y	N	Y	N	Y
Numerical	N	N	N	Y	N	Y
Feature Selection	Y	Y	Y	Y	Y	N
Instance Selection	Y	Y	Y	Y	Y	N

- WEKA [25]: Provides a set of tools for data pre-processing, classification, regression, clustering, association rules and visualization. For the pre-processing of data, WEKA offers a wide variety of techniques to pre-process attributes and instances. In the case of low quality data, reduces the number of techniques, offering less opportunities. WEKA is only able to handle missing values, and provides tools to replace these values by the mode/media. It also allows the addition of noise in nominal attributes.
- KEEL [2]: It is a software tool for data mining. It contains a large collection of classical techniques for knowledge extraction, pre-processing and learning based on computational intelligence, hybrid models, and a module of statistical tests for comparison. KEEL allows the use of different input and output formats such as CSV, XML or ARFF. The data management module includes a wide variety of techniques for selection of instances, feature selection, discretization, etc. Once again, there are few techniques to management low quality data, as it only allows the use of missing values by various imputation methods (clearing of instances with nearest k-neighbor imputation, k-means imputation, etc).
- Rapid miner [19]: Formerly called YALE is an environment for computational learning and data mining paradigm intended to support rapid prototyping. It offers the possibility to use different formats for input/output. The number of pre-processing techniques offered by the tool is great but, again, decreases if we focus on techniques that manage low quality data. Rapid miner can manage missing values using multiple imputation methods: replacement with the mean, mode, maximum, minimum or the value predicted by data mining technique defined by the user. It also allows the introduction of noise in numerical attributes and/or nominal.
- MiningMart [20]: It is developed with the purpose of reuse techniques successfully in the pre-processing of large datasets. MiningMart is not focused on the knowledge discovery process, it is only the pre-processor. It offers various aggregation techniques, discretization, data

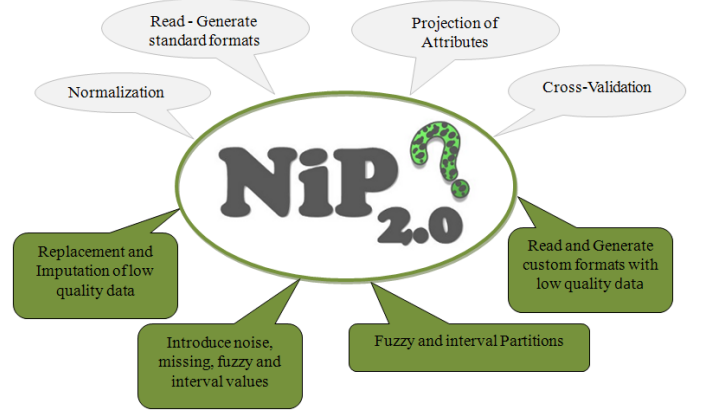


Fig. 1. Functionality of NiP<sub>2.0</sub>

cleaning, treatment of null values and selection of relevant attributes. The only allowed kind of low quality data is missing values, allowing deleting the samples containing them or replacing them by the mode, media or by an imputation method.

There are other tools that focus on the process of knowledge extraction, Adam [23], D2K [17], KNIME [3], Orange [9] or Tanagra [22], among others, but do not put too much attention to the treatment of low quality data.

Table I shows a summary where we can see the main features of the software tools that we have previously presented. In this table we write: “Y” when the tool has the feature which is described in the corresponding row, “I” when feature is only developed intermediate way, and “N” when the tool does not have the feature.

#### IV. NiP<sub>2.0</sub>: A TOOL TO MANAGE LOW QUALITY DATA

NiP<sub>2.0</sub>, [5], is a tool that allows us to generate and manage datasets with low quality data, created with the main purpose of being used in investigation due to absence of similar software that allows to establish a common framework for this kind of data. There is an earlier version of the tool, NIP 1.5. The new version, in general, improves most of the functionalities of the latest version, allowing: work with low

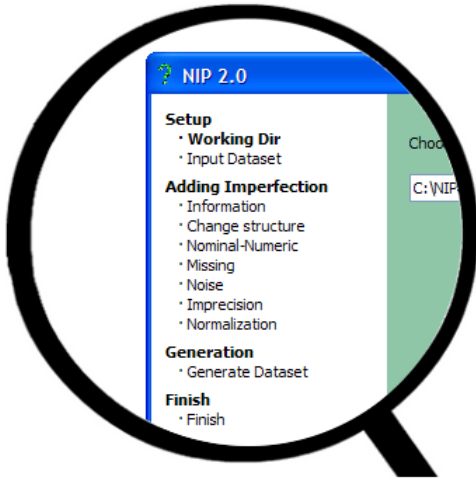


Fig. 2. Tracking process

quality data from the input to the output, the replacement of values (note that in the NiP<sub>2.0</sub> tool, this last functionality is shown as replacement/imputation – soon, the tool will incorporate some imputation algorithms), introduce expert information from partition file and to discretize attributes using several algorithms of partitioning crisp and fuzzy (Figure 1).

In this section we show new functionalities of the system, describing the new options of each one. Figure 1 shows, in a general way, the functionalities of NiP<sub>2.0</sub> where we draw in shading options from the previous version, and highlighted the new features.

In the tool, when we are processing a dataset, we can see in what part of the process we are, because in the left side we can see a menu, shown in Figure 2. This menu shows different options presented for the tool.

#### A. Input Format

As we have commented throughout this paper, NiP<sub>2.0</sub> has been updated to allow to work with low quality data, specifically, with values: 1) missing, 2) described by means of a membership function or 3) described by an interval. In the Figure 3, we show the screen where users can configure their own input format. To show better options we have activated the check box about delimiters, heading and foot. Also, users can select a predefined format as UCI, KEEL, WEKA or CSV.

On the other hand, once the input format is chosen, NiP<sub>2.0</sub> automatically detects all information and features of the dataset including low quality data such as: missing, interval or fuzzy values.

#### B. Adding low quality data in datasets

One of the more important aspects of this tool is to be able to add percentages of low quality values to the datasets. This functionality allows the generation of synthetic experiments which enable to measure the robustness of different data

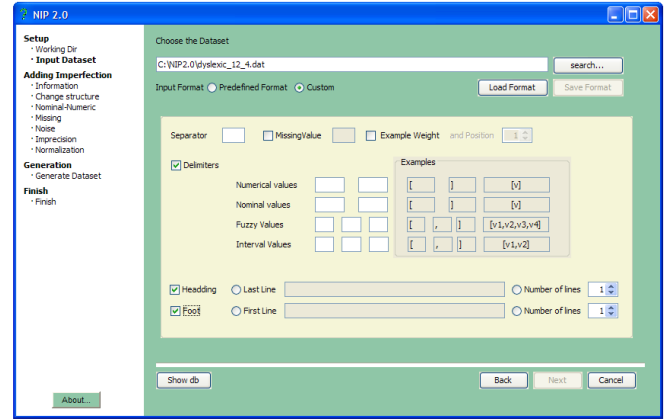


Fig. 3. Custom input format

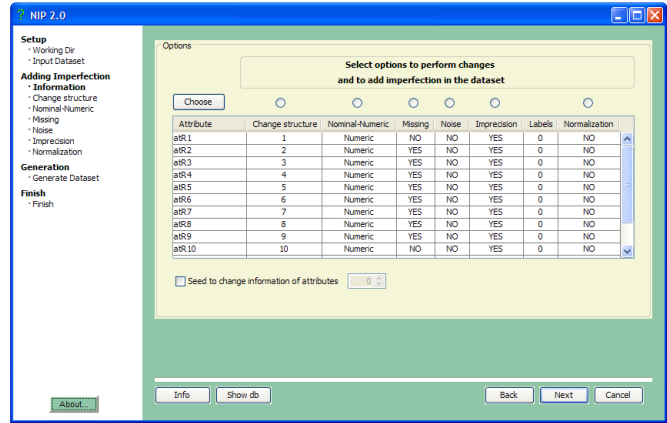


Fig. 4. Screen of summarized information

mining techniques from low quality data. With NiP<sub>2.0</sub> we try to generate a repository of datasets that helps in this area of research.

In this subsection we describe the new functionalities that NiP<sub>2.0</sub> offers to add low quality data in datasets.

Once we have set the input format, we can add into the dataset missing, noise and imprecise values. For this, we must select one of the options shown in Figure 4.

In the table in Figure 4 we can view all information about the dataset and when we make changes in it, this information is updated in the table. In this way, we can know the kind of values that each attribute has. We focus in options referring to the adding of noise and imprecise values, because this is the part where there has been a major expansion of its functionality:

- Imprecise values: As we have explained in previous sections, the Possibility and Interval Theories are used to represent imprecise information. As Figure 5 shows we can add imprecise information to datasets changing values of one numeric attribute for a fuzzy or interval value which contains the value of the attribute. Furthermore,

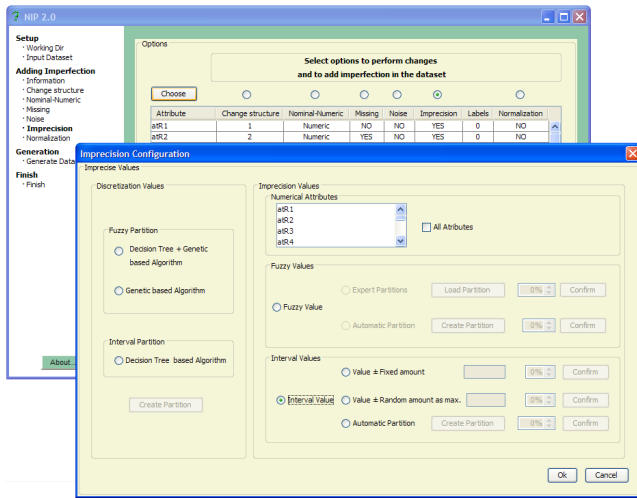


Fig. 5. Adding imprecision to dataset

users can decide which attribute has imprecision and which does not. There are two ways to add imprecision. The first way is introducing fuzzy values. In this case, we need to have a fuzzy partition of the attribute that we want to change. Once we have a fuzzy partition, with a determined percentage, the crisp values will be changed for the fuzzy value of the partition that they belong with the greatest degree. Regarding creating a fuzzy partition of the attributes, we have two options. In the first option we can create an automatic fuzzy partition from one of two algorithms that are available in the tool. Algorithms are *OFP\_CLASS*, [6], which carry out a fuzzy partition using a fuzzy decision tree and a genetic algorithm, and a version of the Yoon-Seok Choie's algorithm, [7], which constructs a fuzzy partition only using a genetic algorithm. In the second option we can indicate a file with a partition which can be contributed through an expert (in this case we introduce expert information in the dataset) or by mean of another algorithm of automatic partitioning.

The second way is introducing interval values. In this case we have three possibilities. First of one, we could generate an interval by adding and subtracting a fixed amount to the value that we want to modify. Another possibility is to generate an interval by adding and subtracting a random amount under the maximum value specified in the option. In both cases, the values that form the intervals obtained will be adjusted to the limits of the attribute domain. The latter option is to add interval values from an automatic crisp partition.

Also, in the imprecision option of NiP<sub>2.0</sub>, we can create a fuzzy or crisp partition of the numerical attributes without introduce imprecision in the dataset.

- Values with noise: We can introduce values with noise, both in nominal and numerical attributes. Figure 6 shows the screen to add noise to the several attributes with different percentages in each attribute. In the case of the

numerical attributes, the noise that we add is gaussian noise. In this kind of noise, it's necessary to indicate in the attribute that we want to modify the mean and the standard deviation and percentages of noise. In the case of the nominal attributes it's only necessary to express the percentage of noise to indicate the number of values of that attribute that will be changed randomly by another value of the domain. Figure 6 does not show attributes in the nominal part, because the dataset only have numerical attributes.

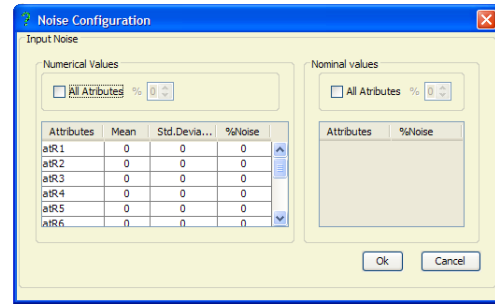


Fig. 6. Adding noise to dataset

### C. Output Format

In the same way at input, users can define output formats (Figure 7). We can choose to configure the output of the dataset in standard formats such as: WEKA, KEEL, UCI or CSV, or we can define output formats or even if we have already used NiP<sub>2.0</sub> and we kept old formats, we can load them.

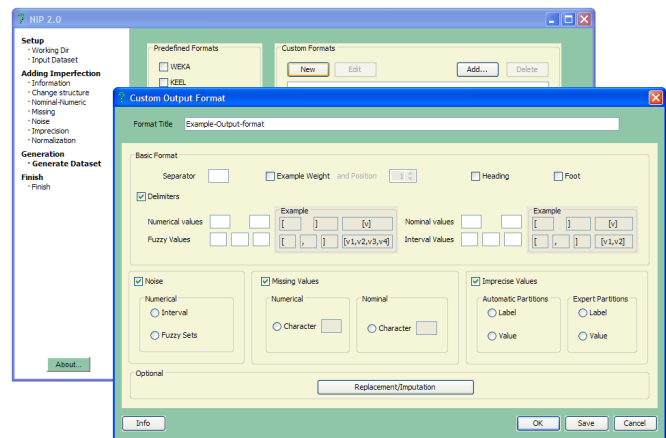


Fig. 7. Custom output format

In the background of Figure 7 we show the options to select standard formats like an output format, and in the foreground, we show the window that gives us the possibility to configure an own output format to work with low quality data. This way, we can define how we prefer to express the fuzzy values, either the name of a linguistic label or the values of a membership function. Also we can configure different options for a fuzzy

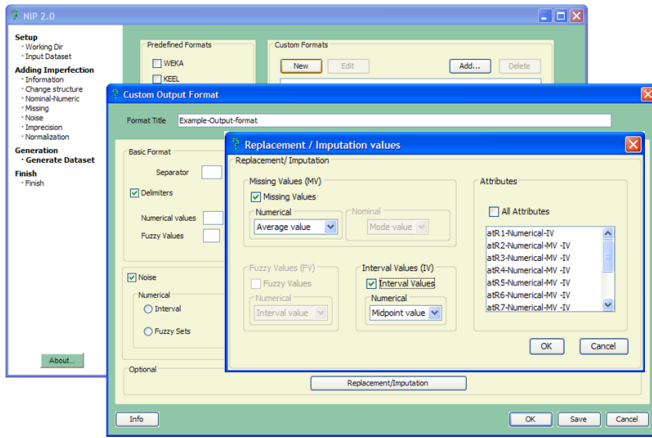


Fig. 8. Replacement/Imputation values

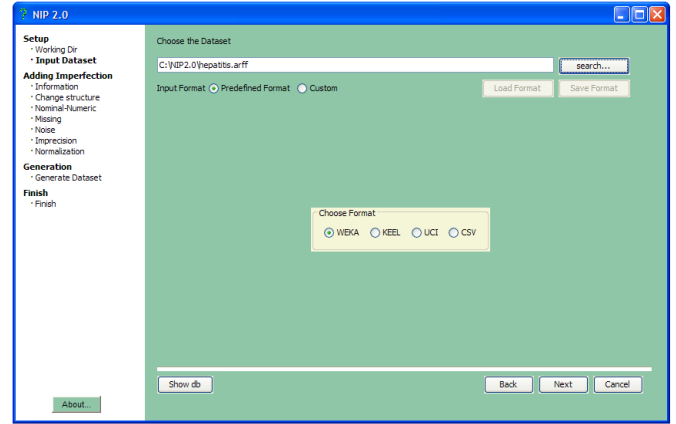


Fig. 10. Selecting input dataset with WEKA format

value of the automatic partition or a fuzzy value of the expert partition.

Moreover, we can replace/impute some types of values. This option is useful when we want to test a technique with low quality data with the same data but replaced/imputed or when a technique can't work with fuzzy, missing or interval values allowing to generate a crisp dataset.

In Figure 8, we show the different options to make a replacement of values. In this case  $NiP_{2.0}$  has automatically detected that the dataset has low quality data: missing and interval values. For this reason, the screen only shows activated the options of missing and interval values allowing to modify these values. It's important to note that in the right side of this foreground screen we can choose which attribute we want to transform, or in otherwise, we can choose to modify all.

## V. BUILDING A LOW QUALITY DATASET USING $NiP_{2.0}$

In this section, we show an example to explain the functionality of the tool  $NiP_{2.0}$ . To carry out this example, we have selected the dataset "hepatitis.arff" in WEKA format. This dataset is composed of 20 attributes including the class attribute and 155 instances. These attributes are nominal and numerical. In Figure 9, we show the first instances of the dataset. In those instances, we can see that "hepatitis.arff" has missing values in some attributes and the class attribute is the last.

```
30,male,no,no,no,no,no,no,no,no,no,no,1,85,18,4,?,no,LIVE
50,female,no,no,yes,no,no,no,no,no,no,no,0,9,135,42,3,5,?,no,LIVE
78,female,yes,no,yes,no,no,yes,no,no,no,no,0,7,96,32,4,?,no,LIVE
31,female,?,yes,no,no,yes,no,no,no,no,0,7,46,52,4,80,no,LIVE
34,female,yes,no,no,no,no,yes,no,no,no,no,1,?,200,4,?,no,LIVE
34,female,yes,no,no,no,no,yes,no,no,no,0,9,95,28,4,75,no,LIVE
51,female,no,no,yes,no,yes,yes,no,yes,yes,no,no,?,?,?,no,DIE
23,female,yes,no,no,no,no,yes,no,no,no,1,?,?,no,LIVE
39,female,yes,no,yes,no,no,yes,yes,no,no,0,7,?,48,4,?,no,LIVE
30,female,yes,no,no,no,no,yes,no,no,no,1,?,120,3,9,?,no,LIVE
39,female,no,yes,no,no,no,yes,no,no,no,1,3,78,30,4,4,85,no,LIVE
32,female,yes,yes,yes,no,no,yes,yes,no,yes,no,1,59,249,3,7,54,no,LIVE
41,female,yes,yes,yes,no,no,yes,yes,no,no,0,9,81,60,3,9,52,no,LIVE
```

Fig. 9. First instances of "hepatitis.arff" dataset

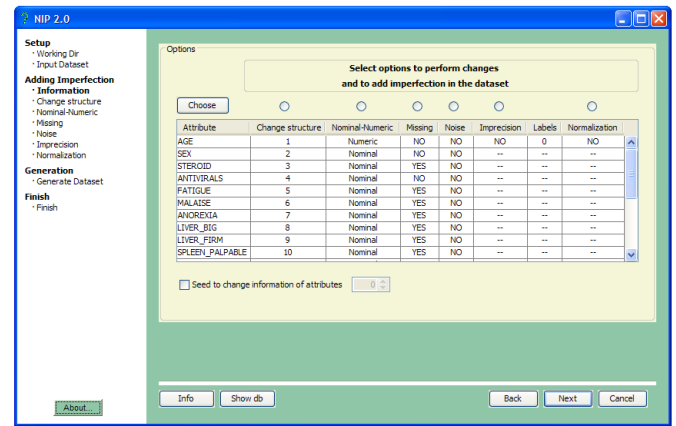


Fig. 11. Screen with summarized information about "hepatitis.arff" dataset

Once we have selected the working directory and the dataset, we must select the input format of this dataset. In this example, the input format is a predefined format, WEKA format, where the attributes of each instance are separated by commas, the missing values are represented by question marks, and attributes are of the specified type at the typical header of this format (Figure 10). If the dataset has another format, mark the appropriate format or define an own format using the option of custom format, as shown in Figure 3. In the cases of UCI or CSV formats, where we can find a specification file attached to the dataset (in the UCI format, the .name file), we must change the features of the dataset read by  $NiP_{2.0}$  using the column "nominal/numerical" (Figure 11) to adapt them to such specification.

We want to note that it may be that the file .data can include a particular attribute that takes values "1" and "2" to represent the values "YES" or "NOT". When reading the dataset and the values "1" and "2",  $NiP_{2.0}$  understands that this attribute is numeric. Therefore, according to the specification attached to the dataset, we must change this attribute to indicate to  $NiP_{2.0}$  that is a nominal attribute.

When the dataset is read (and adjusted the types of attributes, if necessary), we have the table with a summarized



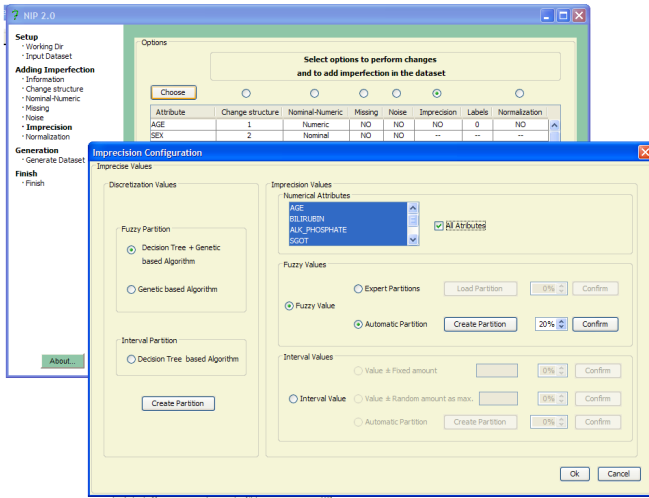


Fig. 12. Adding imprecision to “hepatitis.arff” dataset

information about the dataset. This is the moment when we can add or modify the features of attribute values to create a new dataset with low quality data.

Suppose we want to change the missing values for interval values and we want to add imprecision into some attributes.

To add imprecise information we must select the option imprecision and press the button “choose”. After pressing “choose”, a screen as in Figure 5 appears. In this screen we can configure different options such as: to create an automatic fuzzy or interval partition or to add imprecise information by fuzzy or interval values with different percentages in the attributes we want. In this example we choose the option to add imprecise information from an automatic partition in all numerical attributes. These are the options that are selected in Figure 12. Also we have selected 20% of imprecision for all attributes.

The partition generated is saved in a file shown in Figure 13. In this file, to each attribute the first line shows the name of attribute following of the number of fuzzy sets generated for such attribute. In the next lines, the different fuzzy sets generated are shown. In Figure 13, we only show the fuzzy partition of some numerical attributes.

```
AGE 6
AGE1,7.0,7.0,24.8849,28.1154
AGE2,24.8849,28.1154,37.4022,37.593903
AGE3,37.4022,37.593903,40.3984,40.6043
AGE4,40.3984,40.6043,41.9675,44.0265
AGE5,41.9675,44.0265,52.9370,70.062195
AGE6,52.937,70.062195,78.0,78.0
BILIRUBIN 1
BILIRUBIN0,0.3,0.3,8.0,8.0
ALK_PHOSPHATE 2
ALK_PHOSPHATE1,26.0,26.0,37.3249,201.657
ALK_PHOSPHATE2,37.3249,201.657,295.0,295.0
SGOT 1
SGOT0.14.0.14.0.648.0.648.0
```

Fig. 13. Fuzzy partition of some numerical attributes

When we have chosen the fuzzy partition option and press the “Confirm” and “Ok” buttons respectively, we return to the screen where summarized information is shown in the

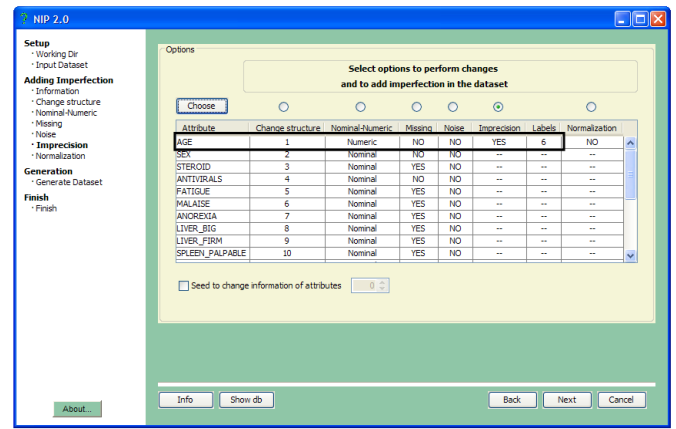


Fig. 14. Updated and summarized information of “hepatitis.arff” dataset

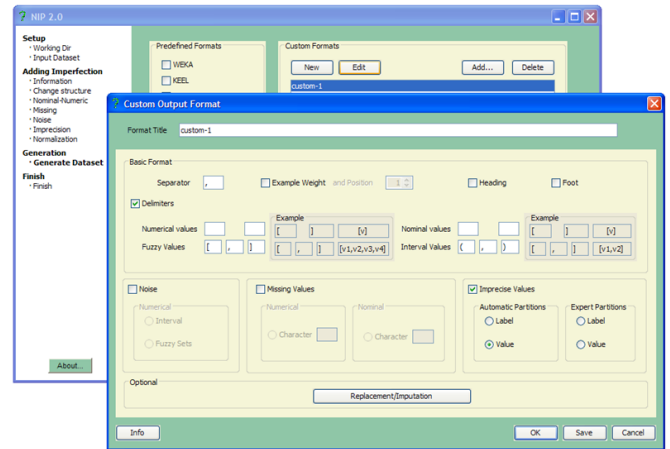


Fig. 15. Configuring the output format

same way as Figure 14, where now is indicated that numerical attributes have imprecision.

Afterwards, as we do not want to add any information to the dataset, we press the button “Next” and we get the screen to configure the output format. In this case, we want to configure an own output format, that’s why we press the button “new” and we have a screen similar to Figure 15. In this screen, we indicate how we want to write the values of the dataset, and besides it is in this screen where we must indicate if we want replace/impute some values. First, we define that interval values are represented by a format “( $n_1, n_2$ )”, fuzzy values are represented by a format “[ $v_1, v_2, v_3, v_4$ ]” and attributes are separated by commas. Second, using the option “Replacement/Imputation”, we replace missing values by intervals in numerical attributes and by mode in nominal ones (Figure 16).

Finally, we have obtained the configuration of the dataset as we want and, in the file that we indicate, we have the transformed dataset. In Figure 17 we show the first instances of the transformed dataset where the missing values are interval values and a percentage of the numerical attributes have been transformed for fuzzy values corresponding to a fuzzy

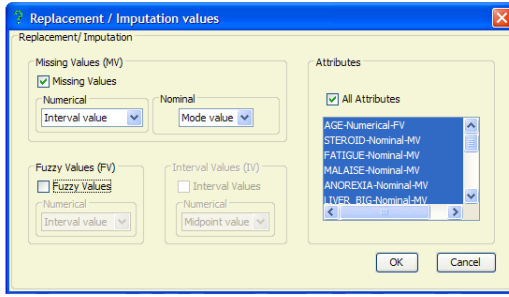


Fig. 16. Replacing missing values by interval and mode values

```
30.0,male,no,no,no,no,no,no,no,no,no,1.0,85.0,[14.0,14.0,648.0,648.0],4.0,(0.
50.0,female,no,no,yes,no,yes,no,yes,no,no,no,0.9,135.0,[14.0,14.0,648.0,648.0],3.
78.0,female,yes,no,yes,no,yes,no,yes,no,no,no,0.0,3.0,3.8,0.8,0],96.0,[14.0,14.0,64
31.0,female,yes,yes,no,no,no,yes,no,no,no,no,0.7,46.0,52.0,13.0464299,3.353449
34.0,female,yes,no,no,no,yes,no,yes,no,no,no,1.0,[26.0,295.0],200.0,4.0,(0.0,100.0
34.0,female,yes,no,no,no,yes,no,yes,no,no,no,0.9,95.0,28.0,4.0,75.0,no,LIVE
51.0,female,no,yes,no,yes,yes,yes,yes,yes,no,0.0,3.8,0.0],(26.0,295.0),(14.0,648.0
23.0,female,yes,no,no,no,yes,no,yes,no,no,no,1.0,(26.0,295.0),(14.0,648.0),(2.1,6.
39.0,female,yes,no,yes,no,no,yes,no,no,no,no,0.7,[26.0,295.0],48.0,(2.1,6.4),(0.0
30.0,female,yes,no,no,no,yes,no,no,no,no,0.0,3.8,0.0),(26.0,295.0),120.0,3.9,(0.0
[37.4022,37.593903,40.3984,40.6043],female,no,yes,no,no,no,yes,no,no,no,1.
[24.8849,28.1154,37.4022,37.593903],female,yes,yes,yes,yes,yes,yes,yes,yes,no,nc
[40.3984,40.6043,41.9675,44.0265],female,yes,yes,yes,yes,yes,yes,yes,yes,yes,no,nc
```

Fig. 17. First instances of file with transformed “hepatitis.arff” dataset

partition. In this figure, we highlight some of the transformed values with respect to the original dataset in Figure 9.

The output of Nip<sub>2.0</sub> is the following: a file (with extension \*.arff (to WEKA format), \*.dat (to KEEL format), \*.data (to UCI format), \*.csv (to CSV format) or \*.custom (to custom format) that contains the low quality dataset, a specification file with the types of attributes and the types of values that contains if the dataset is in UCI, CSV or custom formats (\*.data.spec, \*.csv.spec or \*.custom.spec respectively) and a file with the partition of the numerical attributes.

## VI. CONCLUSIONS

The lack of datasets representing the inherent imperfection in any data collection makes it difficult to develop data mining techniques to manage low quality data. Moreover, it is possible to improve some datasets generated automatically by introducing expert knowledge. With the main objective to improve the creation and management of low quality datasets, in this paper we have presented the improvements that have been made to a tool developed in Java, which we have called Nip<sub>2.0</sub>. This way, we think that the tool can be used as a common framework to create low quality datasets and so, for testing and comparison of data mining techniques and algorithms.

When we compare it with other similar software tools, we can conclude that this tool is able to handle more types of low quality data allowing custom formats in both input and output and so it becomes a more flexible tool.

## ACKNOWLEDGMENT

Supported by the project TIN2011-27696-C02-02 of the MICINN of Spain. Thanks also to the Funding Program for Research Groups of Excellence with code 04552/GERM/06

granted by the “Fundación Séneca” of Spain. R. Martínez is supported by the scholarship program FPI from the “Fundación Séneca”, Murcia, Spain.

## REFERENCES

- [1] A. Asuncion, and D.J. Newman, UCI Machine Learning Repository, <http://www.ics.uci.edu/~mlearn/MLRepository.html>, Irvine, University of California, School of Inf. and Computer Science, 2007.
- [2] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez and F. Herrera, KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Multiple-Valued Logic and Soft Computing*, 17(2–3):255–287, 2011.
- [3] M.R Berthold, N. Cebon, F. Dill, T.R. Gabriel, T. Kttr, T. Meinel, P. Ohl, C. Sieb, K. Thiel and B. Wiswedel, KNIME: The Konstanz Information Miner, In C. Preisach, H. Burkhardt, L. Schmidt-Thieme and R. Decker (eds.), *Data Analysis, Machine Learning and Applications*, pp. 319–326, Berlin: Springer, 2008.
- [4] P.P Bonissone, Approximate Reasoning Systems: Handling Uncertainty and Imprecision in Information Systems, In A. Motro and Ph. Smets (eds.), *Uncertainty Management in Information Systems: From Needs to Solutions*, pp.369–395, Kluwer Academic Publishers, 1997.
- [5] J.M. Cadenas, J.V. Carrillo, M.C. Garrido, R. Martínez-España and E. Muñoz, <http://heurimind.inf.um.es/NIP/index.htm>, 2008.
- [6] J.M. Cadenas, M.C. Garrido, R. Martínez-España, P.P. Bonissone, OFF\_CLASS: A hybrid method to generate Optimized Fuzzy Partitions for Classification, *Soft Computing*, (in press) doi:10.1007/s00500-011-0778-0, 2011.
- [7] Y-S. Choi, B.R. Moon, Feature Selection in Genetic Fuzzy Discretization for the Pattern Classification Problems, *IEICE Transac.*, 90(7):1047–1054, 2007.
- [8] R. Coppi, M.A. Gil and H.A.L. Kiers, The fuzzy approach to statistical analysis, *Computational Statistics & Data Analysis*, 51:1–14, 2006.
- [9] J. Demar, B. Zupan, G. Leban and T. Curk, Orange: From experimental machine learning to interactive data mining, *Lecture Notes in Computer Science*, 3202:537–539, 2004.
- [10] E. Diday and M. Noirhomme-Fraiture, *Symbolic Data Analysis and the SODAS Software*, New York: Wiley Interscience, 2008.
- [11] D. Dubois and H. Prade, *Possibility Theory: An approach to Computerized Processing of Uncertainty*, New York: Plenum Press, 1988.
- [12] D. Dubois and D. Guyonnet, Risk-informed decision-making in the presence of epistemic uncertainty, *International Journal of General Systems*, 40(2):145–167, 2011.
- [13] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern classification*, New York: John Wiley and Sons, 2001.
- [14] J. Han and M. Kamber, *Data mining: Concepts and Techniques*, Morgan Kaufmann Pub., 3rd Edition, 2011.
- [15] R. Hickey, Noise Modeling and Evaluating Learning from Examples, *Artificial Intelligence*, 82(1–2):157–179, 1996.
- [16] J. Little and C. Moler, The Mathworks. Simulink Design Optimization, 1.1 edition, Natic, MA, 2009.
- [17] X. Llorà, E2K: Evolution to knowledge, *SIGEVolution*, 1(3):10-16, 2006.
- [18] D.J.C. Mackay, *Information theory, inference and learning algorithms*, Cambridge: Cambridge University Press, 2003.
- [19] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz and T. Euler, YALE: Rapid Prototyping for Complex Data Mining Tasks, In *Proc. 12th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining*, pp. 935–940, 2006.
- [20] K. Morik and M. Scholz, The MiningMart Approach to Knowledge Discovery in Databases, In eds. N. Zhong, J. Liu (eds.), *Intelligent Tech. for Information Analysis*, pp. 47–65, Berlin: Springer, 2004.
- [21] J.R. Quinlan, *C4.5: Programs for machine learning*, California: Morgan Kaufmann Pub., 1993.
- [22] R. Rakotomalala, TANAGRA: a free software for research and academic purposer, In *Proceedings of EGC’2005, RNTI-E-3*, 2:697–702, 2005.
- [23] J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch and H. Lin, ADA-M: a data mining toolkit for scientists and engineers, *Computers & geosciences*, 31(5):607–618, 2005.
- [24] B. Stewart, DataPreparator: tool for data preparation, preprocessing and exploration for data mining and data analysis. <http://www.datapreparator.com>, 2010.
- [25] I.H. Witten and E. Frank, *Data mining: Practical Machine Learning Tools and Techniques.*, San Francisco: Morgan Kaufmann Pub., 2005.