Manual of FuzzyDataMiner 1.0

Content

1. Introduction

The *FuzzyDataMiner* program was created as a part of the implementation of the data analysis methods described in the post-doctoral dissertation of Dr. Urszula Bentkowska and is available in the INTERNET network. Necessary information as well as the software itself can be obtained from:

http://diagres.ur.edu.pl/~fuzzydataminer/

The main purpose of this software is to present two applications of practical methods described in the dissertation. The first application concerns the classification of test objects using a classifier constructed by the k-NN method in a situation where missing values appear in the test object. The second one is the classifier designed for microarray data, which is characterized by a very large number of attributes and a small number of objects.

2. Technical aspects of the program design

The program has been implemented in Java. Therefore, for proper operation it is necessary to install the *Java SE Development Kit 8* or at least *Java SE Runtime Environment 8* from Oracle. To test the *FuzzyDataMiner* program, from the above mentioned web page, it is necessary to download the zip archive called fuzzydataminer.zip

After unpacking this file, the FuzzyDataMiner directory will appear on the disk containing the following files:

- fuzzydataminer.jar – runnable archive jar of the *FuzzyDataMiner* program,
- biodeg.tab - an example of a data set to test a method with a parameter -S (please see below),
- biodeg_config.txt – an exemplary configuration of 3 experiments to test the method with the parameter -S (please, see below),
- leukemia.tab – an exemplary data set to test the method with the parameter -M (please, see below),
- leukemia_config.txt – an exemplary configuration of 2 experiments to test the method with the parameter -S (please, see below),
- start_s.bat – a batch file to run the experiment with the method -S (this file can be run from the window of the Windows file manager without using the terminal),
- start_m.bat – a batch file to run the experiment with the method -M this file can be run from the window of the Windows file manager without using the terminal).

In the fuzzydataminer directory there will also appear a *lib* subdirectory containing the WEKA API library version 3.8, which is used by the *FuzzyDataMiner* program.

In addition, from the above-mentioned page you can download two sets of data sets used for experiments described in the dissertation. The first collection named **SDataSets.zip** is a data set for experiments with the addition of missing values. The other one, called **MDataSets.zip** is a microarray data set.

3. Using the program

The program is used by running a jar archive called **fuzzydataminer.jar** using the Java virtual machine with the appropriate options. Since the use of the program requires the options, it is convenient to run the program directly in the text terminal of the system or using appropriate overlays for the operating system. For a simple test, you can also use the batch files start_s.bat and start_m.bat (described above).

In general, there are two ways to use the program for experiments.

1. Using the program to perform experiments in the testing of the k-NN classifier on test data with the missing values. The general way of calling the program for this case is as follows:

   java -jar fuzzydataminer.jar -S data.tab config.txt results.txt

The meaning of the parameters is as follows:

- java - fuzzydataminer.jar - running of the fuzzydataminer.jar,
- -S – an option describing the first way of using the program (testing objects with missing values),
- data.tab - the name of the data file that is divided into the training and test part during the experiment,
- config.txt – the file name with calculation configuration,
- results.txt – the file name with calculation results.

2. Using the program to perform experiments in the testing of the k-NN classifier on the microarray data. The general way of calling the program for this case is as follows:

   java -jar fuzzydataminer.jar -M data.tab config.txt results.txt

The meaning of the parameters is the same as described above:

- java - fuzzydataminer.jar - running fuzzydataminer.jar,
- -M – an option describing the second way of applying the program (the microarray data analysis),
- data.tab – the name of the data file which during the experiment is divided according to the leave-one-out method,
- config.txt - the file name with calculation configuration,
- results.txt - – the file name with calculation results.

4. Configuration files

As described in the previous section, for the correct use of the program, apart from providing the correct parameters of the call, you also need to prepare a configuration file whose name is the second parameter of the program. The configuration file is a text file with the parameters of a single experiment that are

described in each line of the file. During the calculations, the program performs successively the experiments and their results are saved to the file with the name being the last parameter of the program's call.

Each of the experiments is described by means of a parameter sequence, with each parameter being given in the following format:

parameter_name=parameter_value

The order of parameters is optional, however, it is necessary to provide all the required parameters for a given experiment. Parameter descriptions are separated by a space.

Parameter sets for both types of experiments (the option -S and the option -M) differ significantly. This is why we describe exactly both sets of parameters.

Calculations with the parameter -S

In the case of calling the program with the -S parameter, the configuration of each experiment should contain the following parameters:

- Partition – this parameter informs in what proportion during the experiment the data table should be divided into the training and test part, e.g. parameter: Partition=0.6 means that the training table will have 60% of objects, the test table will have 40% of objects.
- Missing – this parameter informs what percentage of the randomly selected attribute values in the test data will be substituted during the experiment for the value MISSING, e.g.: parameter Missing=0.3 means that 30% of the values will be substituted.
- Decision - this parameter informs about the main decision value of the decision attribute, e.g. parameter Decision=[RB] means that the main value of the decision is the value RB (square brackets are added for technical reasons to facilitate the parsing of this value).
- Repeat – this parameter informs how many times the experiment should be repeated to obtain the average value of the result and the standard deviation, e.g. parameter Repeat=10 means that 10 repetitions will be made.
- Method - this parameter informs if and what the classifier aggregation method is used; in particular, the value C means that it is a single classifier (without aggregation), the value M means aggregation by means of an arithmetic mean and the value F means aggregation of uncertainty intervals.
- k – this parameter informs about the number of neighbors in the k-NN method; this parameter is used only in the C method, because in the M and F method the information coming from classifiers from different k is aggregated.
- MonteCarlo - this parameter informs about the number of randomly selected objects used to calculate the uncertainty interval in the method F.
- FMeasure - this parameter informs about the way of aggregation in the method F (there are six values numbered from 1 to 6).

This is an exemplary configuration in the case of calling the program with the parameter -S:

Partition=0.6 Missing=0.1 Decision= [RB] Repeat=10 Method=F MonteCarlo=10 FMeasure=3

It is an experiment, with the distribution of input data in the proportion of 60% to 40%, filling the value of Missing is 10% of the attribute values in the test table, the main value the RB decision attribute, ten times

repetition of the experiment, the F method of aggregation of uncertainty intervals based on the aggregation number 3 in the method F.

After performing the experiment with the parameter –S, the results will be saved in the result file as follows. In the first line of the file separate configuration parameter labels are saved along with the labels of the calculated results, there are two result labels: AUC and STDDEV (AUC means an average accuracy and STDDEV means the standard deviation of the accuracy). Then the results of individual experiments are saved in the following lines. For example, for the parameters formulated above, the following text will be printed in the output file.

Partition;Missing;Decision;Repeat;Method;k;MonteCarlo;FMeasure;AUC;STDDEV
0.6;0.1;RB;10;F;0;5;2;0.807;0.014

The last two parameters (0.807 and 0.014) mean the average accuracy and standard deviation, respectively. Note that k has a value of 0 here and has no meaning for the experiment. It occurs only to unify the file to CSV format, which allows to load the results, for example, into a spreadsheet for further analysis.

Calculations with the parameter -M

In the case of calling the program with the parameter -M, the configuration of each experiment should contain the following parameters:

- Decision1 - this parameter informs about the main value of the decision attribute, e.g. parameter Decision=[Cancer] means that the main value of the decision attribute is Cancer.
- Decision2 - this parameter informs about the subordinate value of the decision attribute, e.g. parameter Decision=[Normal] means that the subordinate value of the decision is Normal.
- Repeat – this parameter informs how many times the experiment should be repeated in order to obtain the average value and standard deviation, e.g. parameter Repeat=10 means that 10 repetitions will be performed.
- NoClassHoriz - this parameter informs how many uncertainty intervals are aggregated.
- NoClassVert - this parameter informs how many k-NN classifiers are used to determine the given uncertainty interval.
- Method – this parameter informs whether the aggregation method is used and what is the type of the aggregation method; in particular the value M means the aggregation with the use of the arithmetic mean and the value F means aggregation of the uncertainty intervals.
- k - this parameter informs about the number of neighbors in the k-NN method.
- FMeasure - this parameter informs about the way of aggregation in the method F (there are six values numbered from 1 to 6).

Here is an exemplary configuration when calling the program with the parameter -M:

Decision1=[Cancer] Decision2=[Normal] Repeat=20 NoClassHoriz=5 NoClassVert=10 Method=F k=5 FMeasure=6

This is an experiment where the main value of the decision attribute is Cancer and the subordinate Normal, 20 times the experiment is repeated, where 5 uncertainty intervals are aggregated, each interval is generated with the use of 10 k-NN classifiers, where k = 5, uncertainty intervals are aggregated using the method F, aggregation operator number 3.

After performing the experiment with the parameter -M the results in the result file will be written in the following way. In the first line of the file separate configuration parameter labels are saved along with the

labels of the calculated results, there are 6 labels of the results: ACC;SD_ACC;SEN;SD_SEN;SPEC;SD_SPEC (ACC means the average accuracy along with its standard deviation SD_ACC, SEN means the average sensitivity along with its standard deviation SD_SEN, SPEC means the average specificity along with its standard deviation SD_SPEC).

Then the results of individual experiments are saved in the following lines. For example, for the parameters formulated above, the following text will be printed in the output file.

Decision1;Decision2;Repeat;NoClassHoriz;NoClassVert;Method;k;FMeasure;ACC;SD_ACC;SEN;SD_SEN;SPEC;SD_SPEC [Tumor];[Normal];20;5;10;F;5;6;0.574;0.056;0.583;0.064;0.559;0.057

The last six parameters (**0.574, 0.056, 0.583, 0.064, 0.559, 0.057**) mean the average accuracy and its standard deviation, the average sensitivity and its standard deviation, the average specificity and its standard deviation, respectively.

In the case when the set of parameters does not comply with the above description or if there are any technical problems with the execution of the program (e.g. incorrect file names), the operation of the program is interrupted with an appropriate message. Then the parameters should be corrected and the program should be run again.

5. Format of the file with data

In the FuzzyDataMiner program, the data sets from text files are loaded in the following format. In the first line of the file, after the word ATTRIBUTES, the number of attributes (columns) in the given data set is provided. The following lines give a description of the types of values that can appear in the data set. The information is provided as follows. First, the name of the attribute, next (after the space) the type of attribute value in the form of one of the following three words: **double** (floating point type, otherwise a decimal fraction), **integer** (integer number) or **nominal** (text type). After giving the attribute types, the number of objects is given after the word OBJECTS. Finally, in the following lines, objects are given by specifying the values of all attributes in the same order as the presentation of their types was provided.

Here is an example of a simple data file:

ATTRIBUTES 3
weight double
growth integer
overweight nominal
OBJECTS 3
95 176 yes
65 180 no
85 160 yes

In the data set we have three attributes. The first one is called **weight,** this is the attribute with floating point values. The second one is called **growth**, this is the attribute with integer values. Finally, the third one is called **overweight**, this is the attribute with the text values. After describing the attributes, a description of three objects is made by providing the attribute values.