

# Testing of Inductive Preprocessing Algorithm

Miroslav Čepěk<sup>1</sup>, Pavel Kordík, Miroslav Šnorek

<sup>1</sup>*Dept. of Computer Science and Engineering, Karlovo nám. 13, 121 35 Praha 2, Czech Republic*

cepekmir@fel.cvut.cz

**Abstract.** *The data preprocessing is very important part of the knowledge discovery process. Data mining systems contains tens of preprocessing methods (for example methods for missing data imputation, data reduction, discretization, data enrichment, etc...) and usually it is not clear which methods to use. The selection of preprocessing methods appropriate for particular dataset needs strong experience and a lot of experimenting.*

*In this paper we will test our extension of inductive approach to data preprocessing. We developed inductive preprocessing method which utilizes genetic algorithm to compose from scratch a sequence of preprocessing methods which fits to the data and allows successful model to be created.*

*To test our automatic preprocessing utilize several real-world datasets available from UCI Machine learning repository. To extend our experiments we selected three common problems with dataset – missing data, imbalanced classes and data with noise and introduce them into the data. In this paper we will demonstrate abilities of inductive preprocessing method.*

## Keywords

Inductive preprocessing, UCI

## 1 Introduction

The data preprocessing is very important part of the knowledge discovery process. According to [11] the data preprocessing takes about 80% of whole data mining process. More the preprocessing is corner stone of every data mining process and there is no good model without correctly preprocessed data. To make things even harder data mining systems contains tens preprocessing methods – for example methods for missing values imputation, data reduction, reduction of dimensionality (PCA), data enrichment or normalization. The selection of preprocessing methods and setup of their parameters which are appropriate for given dataset needs strong experience and a lot of experimenting.

To overcome this drawback of the knowledge discovery process we are developing a novel inductive method for automatic selection and configuration of preprocessing method. This inductive method starts with empty preprocessing sequence and adds preprocessing methods which fits to the data and the model trained with this sequence achieves the best accuracy. The aim of this method is to allow less experienced users to preprocess their data and therefore finish the knowledge discovery process with better results – in other words with better accuracy of resulting model.

In this work we will present testing of inductive preprocessing. We will test it with several real world datasets downloaded from UCI database. To extend testing of our method we will introduce selected common problems into the data and we will test if our inductive preprocessing can handle them correctly.

There are several other possible approaches to computer assisted preprocessing. But all of them still requires interaction with user. In general there are three possible ways how to assist user in data preprocessing or usually more general in KDD process [6].

- *Design sequence from scratch* – system allows user to create sequence on his own. Just controls meta-features after each preprocessing method in sequence and indicates to user which preprocessing methods he may use and if the sequence is valid.

- *Design sequence from existing one* – system examines new dataset and searches the most similar problem solved in past. Found sequences are offered to user. Search of similar datasets are usually done using meta-data. Meta-data describes properties of dataset. For example indication of missing values in dataset, statistical properties of data and so on.
- *Design sequence via task decomposition* – user defines a goal which he wants to achieve and system extracts meta-data. With these data system guides user through series of task decomposition.

One possible approach is presented by Intelligent Discovery Assistant (IDA) by Bernstein and Provost [4, 5]. Their approach is based on ontology. IDA is framework for ontology-driven process-oriented assistants for KDD [3]. Assistant concerns about whole KDD process not just preprocessing but the preprocessing is part. IDA helps user to create valid KDD process. Process is composed of several blocks. Each block contains *pre-conditions*, *post-conditions* and *heuristic indicators* [6].

*Pre-conditions* indicates meta-features which data must met before block is applicable. For example input data may contain missing values or must be nominal values, etc. The *post-conditions* describes which meta-features data posses after this block. For example data are normalized or in One-of-N code. With *pre-conditions* and *post-conditions* the IDA may indicate to user which block he may use. And what operations he have to apply to the data.

*Heuristic indicators* indicates influence of block on whole KDD process. How the block affects speed, accuracy, comprehensibility of model, etc... Data reduction increases speed, pruning decreases speed but increases comprehensibility of model (examples are taken from [6]). Definition of *heuristic indicators* allows the IDA to search for the KDD sequence which suits the most to the user defined conditions.

Other possible approach presents MiningMart project [8, 10, 7, 9]. It presents *Design sequence from existing one* approach. MiningMart tries to reuse successful preprocessing sequences. It collects information about both data and preprocessing sequences. Both data and preprocessing sequence compose a case. After successful preprocessing user can add a case to database. When user faces new problem he/she may search through the database of cases and seek for the most similar to the current problem [1].

In MiningMart the building of preprocessing sequence is only on user and is not supported by MiningMart. But successful case is stored in database with meta-data of original dataset. When new dataset is presented to MiningMart it calculates meta-data of dataset and compares them to meta-data of cases stored in database and matching cases are offered to user [6].

## 2 Inductive preprocessing

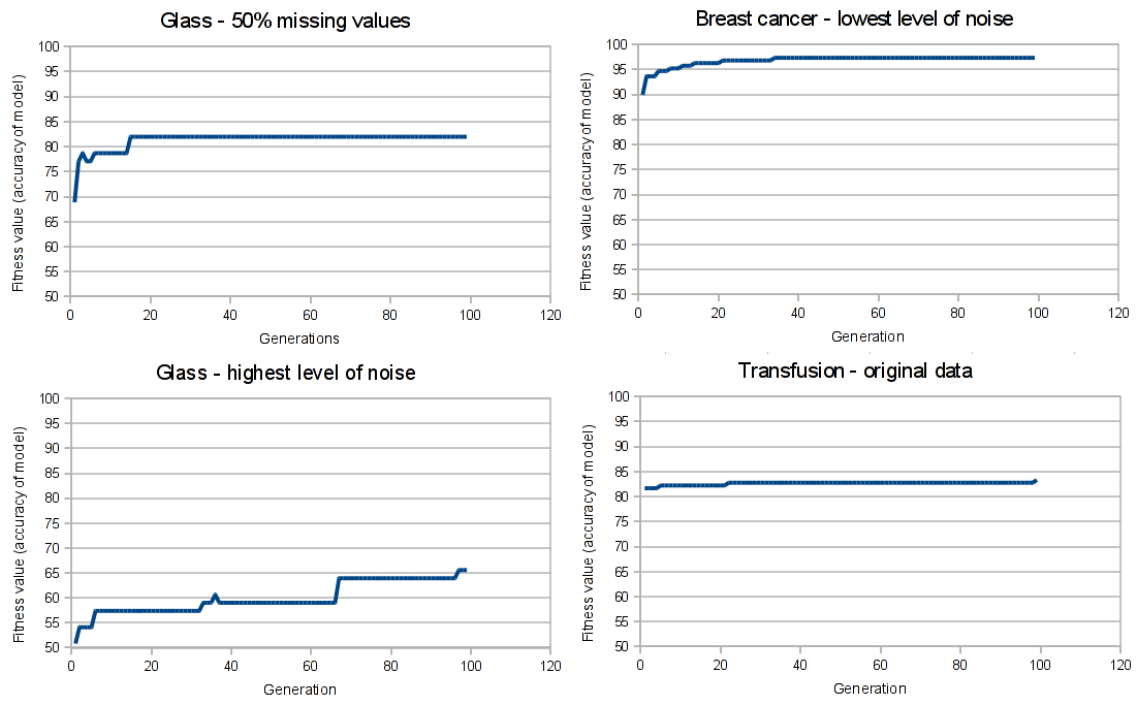
Our solution of complexity of data preprocessing is idea of inductive preprocessing. The idea is to let modelling algorithm to select preprocessing methods which suits to it. For this to utilize genetic algorithm and let it select the most useful preprocessing methods. The genome of each individual consists of lists of preprocessing methods. Individual contains a sequence of preprocessing methods for each input attribute. Methods in given sequence are applied to corresponding attribute only. In addition individuals contain one more sequence called global sequence. This global sequence contains methods which treats the dataset as whole. For example PCA or data reduction methods.

In inductive preprocessing we use only mutation. The mutation may swap, add and/or delete a preprocessing methods in all sequences in individual. We decided to leave out cross-over operation. The reason is that sequence of preprocessing methods makes sense only as a whole. And take one part from one sequence and the other from second does not improve anything and more probably mess up both individuals. We utilize tournament selection with four individuals in tournament. We also utilize concept of elitism which means that two best individuals are automatically copied into next generation.

As fitness function we use accuracy of a model created from preprocessed data. Because each evaluation of fitness mean training of model we have to select modelling method with fast learning. In the end we selected J4.8 decision tree. It is simple and fast yet resulting model is sufficiently accurate.

## 3 Data and Experiments

In our experiments we used several datasets available from UCI Machine learning repository [2]. To be exact we used Breast-cancer, Ecoli, Glass, Transfusion, Wine. For fitness calculation and presented results we used testing data. The



**Fig. 1.** Progress of fitness during generations

original dataset we split into training dataset and testing part. The testing part of dataset is left as is and will be used for calculation of fitness function during inductive preprocessing and also for calculation of errors presented in this article.

The training part is used to train models. In the first two experiments simulates selected common problems with data – missing values, noise and imbalanced classes. More we created three different levels of severity of problems. It means we created three training datasets with different portions of missing values, three different amplitudes of noise and three dataset with different portions of one class removed.

As mentioned above first we will perform two experiments. The first experiment tests improvement of fitness value along generation of genetic algorithm. This test is crucial to test if inductive preprocessing is working. The second experiment will demonstrate that inductive preprocessing is able to find sequence of preprocessing methods which surpasses simple manual preprocessing or no preprocessing. For this experiment we will use datasets with missing data. The reason is that this problem is solvable by application of single imputation preprocessing method. We will impute zero value instead of missing values.

In the last experiment we will demonstrate ability of inductive preprocessing to work with dataset containing more complex problems. We will use Breast cancer dataset and we will introduce missing values, outliers and we will imbalance classes. And then we will use inductive preprocessing to select suitable methods and allow successful methods to be created.

## 4 Results

First we tested if our inductive preprocessing works correctly and if it is able to find preprocessing methods which allows good model to be constructed. Since we use genetic algorithm it is important how the fitness improves along generations. This we will present in this part.

In figure 1 we demonstrate that inductive preprocessing is able to improve fitness over generations and therefore improve accuracy of models. This figure shows fitness for following datasets – Glass with 50% of missing values, Glass with the highest level of noise, original Transfusion data and Breast cancer dataset with the lowest level of noise. The improvement of fitness or accuracy of models on testing data shows that inductive preprocessing is able to find and combine preprocessing methods in way that resulting model is more accurate.

The table 1 demonstrates that models trained with inductive preprocessing are more accurate on testing data than models with manual preprocessing or with no preprocessing. In this experiment we will use only missing data since in this case it is obvious that we will have to apply missing data imputation method. We chose to replace missing values with zero. The first column in table shows accuracy of models with no preprocessing method applied. The second with manual preprocessing. This preprocessing consists of replacing missing values with zeros. The last column shows accuracy of model trained with best sequence of preprocessing method applied to the training data. All accuracies was obtained from testing data which are the same for all three columns.

**Tab. 1.** Comparison of Inductive preprocessing with manual preprocessing and no preprocessing on data with missing values

Dataset name	without preprocessing	manual preprocessing	inductive preprocessing
Breast Cancer - 1% missing	95.7	96.3	97.3
Breast Cancer - 25% missing	95.5	94.7	98.4
Breast Cancer - 50% missing	95.2	92.6	98.4
Ecoli - 1% missing	85.3	88	94.6
Ecoli - 25% missing	85.3	84	92
Ecoli - 50% missing	81.3	84	88
Glass - 1% missing	73.7	72.1	86.6
Glass - 25% missing	68.8	62.3	82
Glass - 50% missing	67.2	57.3	82
Transfusion - 1% missing	79.4	80	84.4
Transfusion - 25% missing	76.6	77.2	81.7
Transfusion - 50% missing	76.6	76.6	83.9
Wine - 1% missing	97.7	97.7	100
Wine - 25% missing	95.3	88.3	100
Wine - 50% missing	88.3	51.1	100

Table 1 shows that models created with data after the inductive preprocessing in all cases surpasses models created with both simple and no preprocessing. The difference is sometimes very significant – in case of Glass and Wine datasets the difference in accuracy is more than 10%.

There is also one more interesting point. Sometimes the manual preprocessing of data brings less accurate results than no preprocessing. The reason is that decision tree J4.8 is able to handle missing values and J4.8 is able to take correct decision even if some values is missing. On the other hand when missing value is replaced with zero all values are taken into consideration. And since there are many zeros in dataset model is confused and is not much accurate.

In the table 2 we present remaining results of inductive preprocessing. The table contains only results of inductive preprocessing and results for data without preprocessing. It shows that the Inductive preprocessing always improves accuracy of model.

The last experiment represents more complex problem for inductive preprocessing. In this experiment we used the Breast cancer dataset and removed 25% of values, added 20% of outliers and removed about 50% of instances of class with less instances. The original ratio was 334 to 165 (first to second class). After imbalancing the ratio is 334 to 81.

The result of this experiment is quite encouraging. The model with distorted training dataset achieved accuracy of 90% on testing set. On the other hand the accuracy of model created with inductively preprocessed data achieved accuracy of 96.8%. Though this improvement is not that high shows that inductive preprocessing is able to handle several problems present in the data at once. Another interesting point is that the model with the distorted training dataset achieves only 90% which is far less than in any other training dataset. But inductively preprocessed training dataset achieves accuracy just lightly worse than original dataset (without any distortion).

This shows that even from worse starting condition the inductive preprocessing is able to achieve very good results.

## 5 Conclusion

In this paper we tested our concept of inductive preprocessing. We tested our concept with several real-world datasets from UCI repository. To test our inductive preprocessing approach we performed three experiments.

In the first experiment we tested if genetic algorithm is working. We examined value of fitness of the best individual along the generations and examined progress of best-so-far individual. The results of this experiment look satisfactory and

The second experiment demonstrates that datasets treated with inductive preprocessing creates more accurate models. As shown in tables 1 and 2 models created after inductive preprocessing always outperform models with no preprocessing and also models with manual preprocessing. This is very important result. It shows that inductive preprocessing finds such sequence of methods which suits the data. And with application of inductive preprocessing one obtain better results than without any preprocessing.

The last experiment shows that inductive preprocessing achieves satisfactory results even when several distortions are present in the dataset. And results shows that even in case of distorted dataset the inductive preprocessing is able to find preprocessing methods which brings – in terms of accuracy – dataset close to original data.

## Acknowledgments

This research is partially supported by the grant Automated Knowledge Extraction (KJB201210701) of the Grant Agency of the Academy of Science of the Czech Republic and the research program "Transdisciplinary Research in the Area of Biomedical Engineering II" (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

## References

- [1] Miningmart internet case base, available <http://mmart.cs.uni-dortmund.de/end-user/casebase.html>.
- [2] Uci machine learning repository, available at <http://www.ics.uci.edu/mllearn/mlrepository.html>, Sept. 2006.
- [3] B. A., P. F., and H. S. Intelligent assistance for the data mining process: An ontology-based approach. *Information Systems Working Papers Series*, 2002.
- [4] A. Bernstein and F. Provost. An intelligent assistant for the knowledge discovery process. *Proceedings of the IJCAI-01 Workshop on Wrappers for Performance Enhancement in KDD*, 2001.
- [5] A. Bernstein, F. Provost, and S. Hill. Towards intelligent assistance for a data mining proces. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):503518, 2005.
- [6] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning, Applications to Data Mining*. Cognitive Technologies. Springer Berlin Heidelberg, 2009.
- [7] T. Euler. Publishing operational models of data mining case studies. *Proceedings of the ICDM Workshop on Data Mining Case Studies*, 2005.
- [8] T. Euler, K. Morik, and M. Scholz. Miningmart: Sharing successful kdd processes. *LLWA 2003 Tagungsband der GI-Workshop-Woche Lehren Lernen Wissen Adaptivitat*, 2003.
- [9] T. Euler and M. Scholz. Using ontologies in a kdd workbench. *Proceedings of the ECML/PKDD Workshop on Knowledge Discovery and Ontologies*, 2004.
- [10] K. Morik and M. Scholz. The miningmart approach to knowledge discovery in databases. In N. Zhong and J. Liu, editors, *Intelligent Technologies for Information Analysis*. Springer, 2004.
- [11] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, 1999.

**Tab. 2.** Comparison of Inductive preprocessing with manual preprocessing and no preprocessing on data with missing values

Dataset name	Without Pre-processing (Accuracy [%])	Inductive Preprocessing (Accuracy [%])
Breast Cancer - Original data	95.8	97.8
Breast Cancer - Disbalanced – Low	95.7	97.4
Breast Cancer - Disbalanced – Medium	95.7	97.4
Breast Cancer - Disbalanced – High	94.6	97.8
Breast Cancer - Noise – Low	39.2	97.4
Breast Cancer - Noise – Medium	60.8	97.4
Breast Cancer - Noise – High	60.8	97.8
Ecoli - Original data	85.3	92
Ecoli - Disbalanced – Low	85.3	92
Ecoli - Disbalanced – Medium	85.3	92
Ecoli - Disbalanced – High	85.3	93.3
Ecoli - Noise – Low	45.3	72
Ecoli - Noise – Medium	45.3	64
Ecoli - Noise – High	45.2	74.6
Glass - Original data	67.2	83.6
Glass - Disbalanced – Low	67.2	95.4
Glass - Disbalanced – Medium	67.2	81.9
Glass - Disbalanced – High	65.5	86.6
Glass - Noise – Low	34.4	70.4
Glass - Noise – Medium	44.2	63.9
Glass - Noise – High	36	63.9
Transfusion - Original data	80	83.3
Transfusion - Disbalanced – Low	80	83.6
Transfusion - Disbalanced – Medium	77.2	85.2
Transfusion - Disbalanced – High	76.6	80.3
Transfusion - Noise – Low	76.6	77.2
Transfusion - Noise – Medium	76.6	77.2
Transfusion - Noise – High	76.6	76.6
Wine - Original data	95.3	100
Wine - Disbalanced – Low	95.3	100
Wine - Disbalanced – Medium	90.7	100
Wine - Disbalanced – High	95.3	100
Wine - Noise – Low	41.8	88.3
Wine - Noise – Medium	41.8	88.3
Wine - Noise – High	41.8	100

**Tab. 3.** Comparison of model accuracy of Inductive preprocessing to No preprocessing

Dataset name	Without Pre-processing (Accuracy [%])	Inductive preprocessing (Accuracy [%])
Original Breast-Cancer dataset	95.8	97.8
Distorted Breast-Cancer dataset	90	96.8