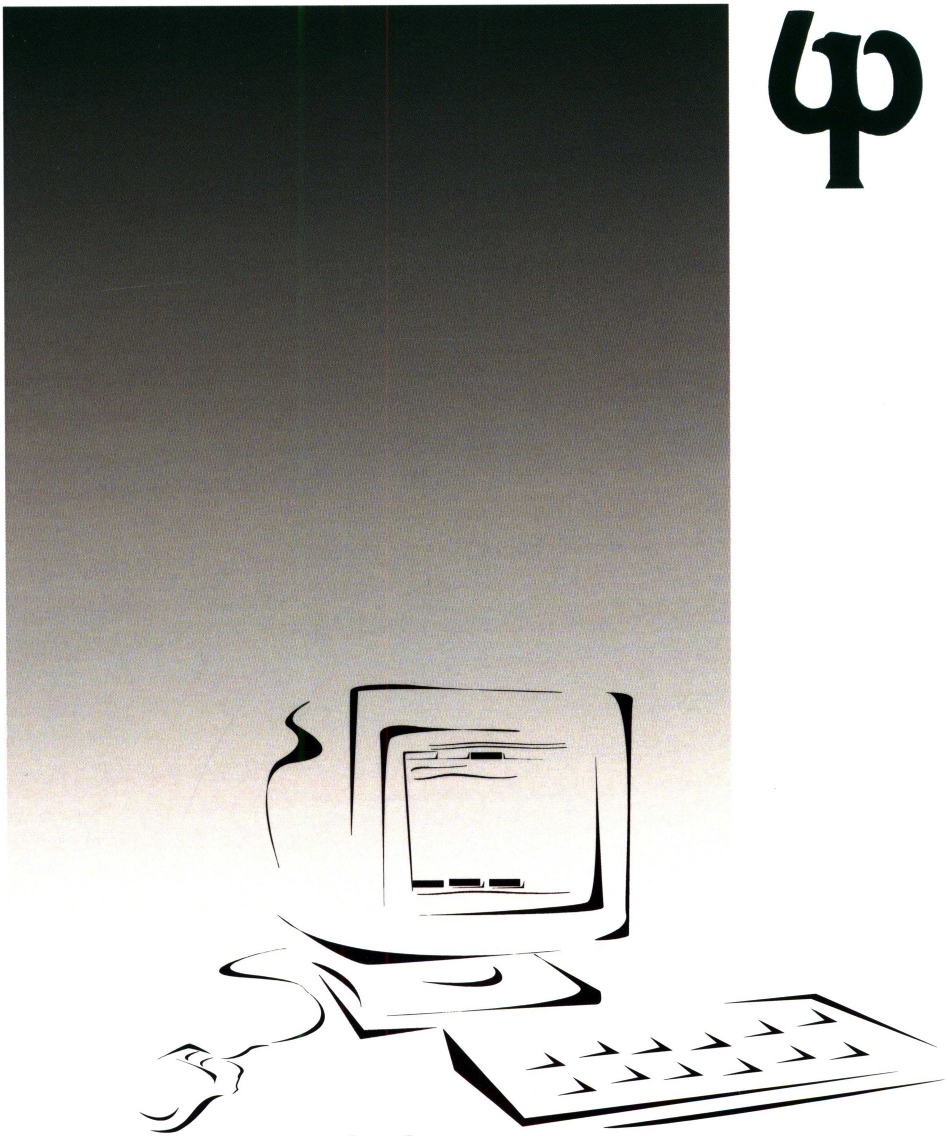
BIAŁYSTOK UNIVERISTY OF TECHNOLOGY

# Advances in Computer Science Research

ISSN 2300-715X



14
2018

# Advances in Computer Science Research

## Volume 14

Oficyna Wydawnicza Politechniki Białostockiej
Białystok 2018

**EDITOR-IN-CHIEF / REDAKTOR NACZELNY:**
Marek Krętowski (m.kretowski@pb.edu.pl, 85 746 90 95)

**DEPUTY EDITOR / ZASTĘPCA REDAKTORA NACZELNEGO:**
Marek Parfieniuk (m.parfieniuk@pb.edu.pl, 85 746 91 08)

**SCIENTIFIC EDITOR / SEKRETARZ NAUKOWY:**
Magdalena Topczewska (m.topczewska@pb.edu.pl, 85 746 90 86)

**STATISTICAL EDITOR / REDAKTOR STATYSTYCZNY:**
Dorota Mozyrska (d.mozyrska@pb.edu.pl, 85 746 90 82)

**TECHNICAL EDITOR / SEKRETARZ TECHNICZNY:**
Tomasz Łukaszuk (t.lukaszuk@pb.edu.pl, 85 746 90 86)

The articles published in *Advances in Computer Science Research*
have been given a favourable opinion by reviewers designated by Editor-In-Chief and Scientific Board

**ISSN 2300-715X**

This journal is the continuation of *Zeszyty Naukowe Politechniki Białostockiej. Informatyka*
issued in the years 2002-2012, ISSN 1644-0331

**THE ADRESS FOR THE CORRESPONDENCE / ADRES DO KORESPONDENCJI:**

„*Advances in Computer Science Research*"
Faculty of Computer Science / Wydział Informatyki
Bialystok University of Technology / Politechnika Białostocka
ul. Wiejska 45a, 15-351 Białystok
tel. 85 746 90 50, fax 85 746 97 22
e-mail: znpb@irys.wi.pb.edu.pl
http://irys.wi.pb.edu.pl/acsr

# CONTENTS

# IMPACT OF IMAGE PREPROCESSING ON RECOGNITION OF LETTERS OF SIGN LANGUAGE

Paweł Abramowicz, Magdalena Topczewska

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

**Abstract:** The article concerns the problem of the selected sign language letters in the form of images classification. The impact of the image preprocessing methods as adaptive thresholding or edge detection is tested. In addition, the influence of the found shapes filling is checked, as well as centering the hands on the images. The following classification methods were chosen: SVM classifier with linear kernel function, Naive Bayes and Random Forests. The accuracy, F-measure, the AUC, MAE and Kappa coefficient were reported as measures of classification quality.

**Keywords:** image preprocessing, sign language, classification

## 1. Introduction

In Poland according to the Universal Newborn Hearing Screening Program 3 children out of 1000 are born with hearing impairment [12], while the report of the Central Statistical Office shouts that 14% of people ranging in age from 15 to 70 have hearing defect [18,5]. Thus the problem may be present from birth or may also occur at a later age. Some deaf or hearing-impaired people use a sign language as a form of communication and expressing emotions.

There are several hundred sign languages around the world. Each sign language consists of ideographic and dactylographic signs. Ideographic signs can be considered as equivalents of short phrases in spoken language, while among dactylographic signs finger alphabet, characters assigned to punctuation or numerals may be mentioned.

To recognize hand postures many different techniques can be applied not only to sign language applications, but also to games or human-computer interaction sys-

Advances in Computer Science Research, vol. 14, pp. 5-16, 2018. DOI: 10.24427/acsr-2018-vol14-0001

tems. Some approaches concern the preprocessing of images, some – feature extraction. The differences also apply to the classification methods. For instance, in [16] a cascade classifier applying AdaBoost method was used to separate 21 letters in Thai finger-spelling. The main feature of hand postures was an object detection approach based on Histogram of Orientation Gradient (HOG). In case of [17], detection of the finger contour using hidden Markov models in the American Sign Language gestures was described, while in [13] Canny edge detection and boundary tracing algorithm were applied to detect fingers location. Additionally, in many applications contour detection techniques were applied to extract features, for instance, in [14] the Moore–Neighbor algorithm was used to obtain the external shape of every image. Next, the number of pixels to form the shape was reduced, and finally neural networks were implemented to classify objects. Finger detection could also be achieved by a color segmentation and a contour extraction [9,7]. The SVM method and HOG descriptors were used to recognize Arabic Sign Language alphabet [2].

In this paper selected methods of image preprocessing were used to verify the ability to improve the classification quality of the chosen finger alphabet letters collected as photographs. The comparison between original images and images after preprocessing (adaptive thresholding or edge detection) was performed. Additionally, the results of two more experiments were obtained and confronted – shapes filling and centering. All mentioned transformations were compared using three well-known classification algorithms (SVM, Naive Bayes, and Random Forests) and detailed performance measures as the classification quality, F-measure, the AUC, Kappa coefficient and MAE were reported.

The data set used in the experiments was a subset of the Hand Posture and Gesture Datasets [11] and contained four letters of the sign alphabet: A, B, C, and V. The set consisted of 191 elements. Each character was photographed on a dark and light background – 24 photographs were available for each character on a given background. Only the letter V was photographed 23 times on a dark background. A size of each photo was 128 × 128 pixels. The colors of the images were represented in the grayscale.

## 2. Selected methods of image preprocessing

In this paper two main approaches have been selected to convert original photographs into images that should enable methods of classification improvement of the allocation of the letters to classes: adaptive thresholding and edge detection. Next using obtained results shape filling and objects centering have been additionally applied to investigate the total impact on the results.

## 2.1 Thresholding

Thresholding is an image segmentation method that based on a colour or a grayscale image creates a binary image as a result [4,6]. The algorithm in its simplest form adapts the threshold value on each pixel comparing it to the intensity of a pixel. Pixels with intensity lower than the threshold are replaced by pixels with one colour (ex. black), while these with intensity greater that the threshold are replaced by pixels with the second colour (*maxValue*, ex. white):

$$destination(x,y) = \begin{cases} maxValue & \text{if } source(x,y) > threshold, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $source(x,y)$ is an intensity of a pixel, $threshold$ is set by the user.

Due to the fact that on the considered images hands showing sign alphabet letters are illuminated unevenly, a simple thresholding may not give the expected effect. Therefore, the adaptive thresholding is worth using.

The image is divided into separate regions and for each region the threshold is calculated separately. In addition to the parameter corresponding to the size of a region (blockSize), the *c* parameter as the global threshold is present. It is a constant that is subtracted from the average intensity of pixels in a given region. Hence, the method allows a user to reject background pixels where there is no differential intensity.

By controlling two mentioned parameters different final effects can be achieved.

## 2.2 Edge detection

The second method of image preprocessing considered in the paper is edge detection [19,10]. The main purpose is to identify points in the digital image in which the light intensity changes rapidly. The Canny's method was used to achieve it [3].

The Canny's algorithm starts by reducing noise in the image. Edge detection is very susceptible to a noise in the raw image and false edges may be created. To reduce a noise a 5x5 Gauss filter is applied to the image resulting in a slightly blurred image that is not affected by interference in a significant way.

In the next step, to detect horizontal and vertical edges, the gradient is searched using the Sobel's operator [10,15]. The operator returns the value of the first derivative for the vertical and for the horizontal direction. The value and the direction of the gradient can then be calculated.

Afterwards, non-maximum suppression pixels are removed, because they are not considered as a part of the edge. Therefore, only thin lines composed of individual pixels remain.

The last step of the algorithm is thresholding using the hysteresis to eliminate irrelevant edges that have a slope below a given threshold. The Canny's method uses two thresholds: lower and upper. If the pixel gradient is greater than the upper threshold, the pixel is considered as the edge. If the pixel gradient is smaller than the lower threshold, the pixel is discarded. Otherwise, if it is between the lower and the upper threshold, the pixel will only be accepted if it is connected to a pixel whose gradient value is above the upper threshold. By controlling these two parameters different final results can be obtained.

### 2.3 Shape filling and centering

In order to improve image recognition, two additional approaches have been implemented and tested. The first was to fill the shapes of hands obtained using methods described in two previous subsections, and the successive approach was to center the filled hands on the image.

The algorithm of the shape filling is as follows. A given row of the image is checked until the first pixel in appropriate colour occurs. Then the last pixel in such a colour is found. Finally, specified segment of pixels is filled with a chosen colour.

The hand centering algorithm works on a simple principle. To assess whether the photographed hand is in the middle of the image, the last row of pixels on which the wrist of the photographed person is visible is checked. Based on the first and the last pixel of the wrist, the current position of the hand is calculated. If the calculated center of the wrist is not in the center of the image, the hand is moved in the right direction, so that it is exactly in the middle.

## 3. Experiments

To perform the experiments the Hand Posture and Gesture Dataset [11] was chosen and A, B, C, and V letters of the sign alphabet. Additional impediment is the occurrence of three different types of a background of the photographs: white, black, and mixed. The number of chosen images every type is presented in Table 1.

Each image was processed into a feature vector with 16384 values of attributes. Finally, four experiments were performed using the own implementations and the Java - ML Library [1,8]:

- experiment 1: to examine the classification quality on the feature vectors created on the basis of original images;
- experiment 2: to examine the classification quality on the feature vectors created on the basis of images after adaptive thresholding or edge detection;

**Table 1.** Number of images for each letter and background colour

| | Background | | |
|---|---|---|---|
| Letter | white | black | mixed |
| A | 24 | 24 | 48 |
| B | 24 | 24 | 48 |
| C | 24 | 24 | 48 |
| V | 24 | 23 | 47 |

- experiment 3: to examine the classification quality on the feature vectors created on the basis of images after adaptive thresholding or edge detection, then filled;
- experiment 4: to examine the classification quality on the feature vectors created on the basis of images after adaptive thresholding or edge detection, then filled and centered.

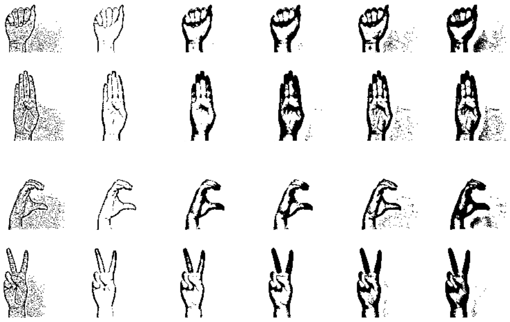


**Fig. 1.** Selected images after adaptive thresholding with a number of sample parameters

Weka system [20] was used to examine the classification quality with the 10-folds crossvalidation testing. The range of parameters for edge detection was from 10 to 140 every 10, while for adaptive learning it was from 3 to 23 every 2 for the first parameter and from 3 to 24 for the second parameter. The examples of images with a different set of parameters are presented in Figure 1 for adaptive thresholding for white background and in Figure 2 for edge detection and black background. The final dataset contained images as a result of all combinations of parameters to not have biased results by arbitrary choice of best parameters. The goal is to check if these methods work in general, thus the results should be interpreted as average results for methods, but not the best to obtain.

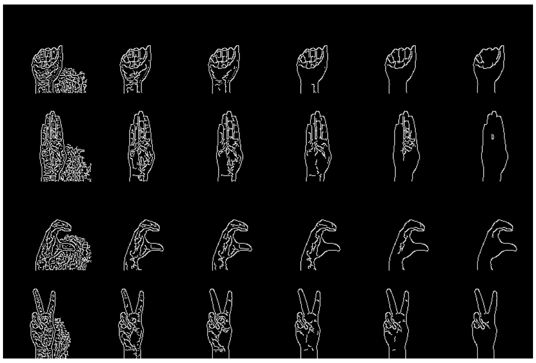


**Fig. 2.** Selected images after edge detection with a number of sample parameters

Detailed results for all experiments are aggregated in Table 2.

The classification quality (the accuracy) is presented as a percentage of the number of correctly classified objects over the number of all objects in a data set. F-measure considers both precision (the proportion of relevant objects that have been correctly classified over the total amount of objects classified as relevant) and recall (the proportion of relevant objects that have been correctly classified over the total amount of relevant objects) and signifies the harmonic mean of these two measures. The higher and closer to 1, the better the predictive property of a classifier. The Kappa coefficient describes the agreement of prediction with true class, and the value 1 signifies complete agreement. AUC signifies the area under the Receiver Operating Characteristic curve and also quantifies the classifier performance. It determines which of the used models predicts the classes best. It combines true positive rate (recall) and false positive rate (proportion of second class objects classified incorrectly as relevant over the total amount of second class objects). The closer AUC for a model comes to 1, the better it is. The last reported measure is the Mean Absolute Error (MAE). In the classification problem it is the sum over all the objects and their absolute error per object divided by the number of objects in the test set with an actual class label and zero means a perfect classification.

Table 2: Classification results (BC - Background Color; Cl - Classifier; MAE - Mean Absolute Error; F - mean value of F-measure; AUC - mean value of Area under the ROC curve; AT - Adaptive Thresholding; ED - Edge Detection

| BC | Cl | Q | Kappa | MAE | F | AUC |
|---|---|---|---|---|---|---|
| Experiment 1 | | | | | | |
| Mixed | SVM | 52.356 | 0.365 | 0.320 | 0.526 | 0.726 |
| | Naive Bayes | 32.984 | 0.107 | 0.335 | 0.338 | 0.574 |
| | Random Forest | **67.539** | 0.567 | 0.257 | 0.671 | 0.866 |
| Black | SVM | **81.052** | 0.747 | 0.275 | 0.810 | 0.891 |
| | Naive Bayes | 38.947 | 0.189 | 0.305 | 0.389 | 0.597 |
| | Random Forest | 64.211 | 0.522 | 0.266 | 0.628 | 0.807 |
| White | SVM | **88.542** | 0.847 | 0.264 | 0.886 | 0.947 |
| | Naive Bayes | 56.25 | 0.417 | 0.219 | 0.543 | 0.713 |
| | Random Forest | 69.792 | 0.597 | 0.216 | 0.695 | 0.923 |
| AT − experiment 2 | | | | | | |
| Mixed | SVM | **76.937** | 0.692 | 0.277 | 0.767 | 0.890 |
| | Naive Bayes | 70.499 | 0.607 | 0.148 | 0.705 | 0.880 |
| | Random Forest | 58.202 | 0.442 | 0.283 | 0.573 | 0.823 |
| Black | SVM | 66.238 | 0.550 | 0.296 | 0.659 | 0.813 |
| | Naive Bayes | **66.672** | 0.556 | 0.167 | 0.665 | 0.855 |
| | Random Forest | 48.882 | 0.317 | 0.316 | 0.477 | 0.732 |
| White | SVM | **71.419** | 0.619 | 0.281 | 0.712 | 0.878 |
| | Naive Bayes | 69.559 | 0.594 | 0.152 | 0.697 | 0.876 |
| | Random Forest | 56.577 | 0.421 | 0.286 | 0.556 | 0.822 |
| ED − experiment 2 | | | | | | |
| Mixed | SVM | **81.136** | 0.748 | 0.273 | 0.811 | 0.903 |
| | Naive Bayes | 76.589 | 0.688 | 0.117 | 0.766 | 0.920 |
| | Random Forest | 48.948 | 0.319 | 0.317 | 0.472 | 0.745 |
| Black | SVM | **79.817** | 0.730 | 0.276 | 0.790 | 0.881 |
| | Naive Bayes | 72.449 | 0.633 | 0.137 | 0.721 | 0.895 |
| | Random Forest | 44.930 | 0.264 | 0.331 | 0.434 | 0.697 |
| White | SVM | **66.061** | 0.547 | 0.291 | 0.657 | 0.833 |
| | Naive Bayes | 64.063 | 0.521 | 0.180 | 0.638 | 0.872 |
| | Random Forest | 40.795 | 0.211 | 0.339 | 0.394 | 0.673 |
| AT, shape filling − experiment 3 | | | | | | |
| Mixed | SVM | **78.850** | 0.718 | 0.277 | 0.788 | 0.891 |
| | Naive Bayes | 66.648 | 0.555 | 0.167 | 0.664 | 0.815 |
| | Random Forest | 72.515 | 0.633 | 0.204 | 0.720 | 0.907 |
| Black | SVM | **67.377** | 0.565 | 0.295 | 0.673 | 0.819 |
| | Naive Bayes | 62.679 | 0.503 | 0.187 | 0.620 | 0.789 |
| | Random Forest | 60.696 | 0.475 | 0.247 | 0.597 | 0.825 |
| White | SVM | **82.825** | 0.771 | 0.268 | 0.828 | 0.926 |
| | Naive Bayes | 67.209 | 0.563 | 0.164 | 0.659 | 0.844 |
| | Random Forest | 75.887 | 0.678 | 0.192 | 0.754 | 0.936 |
| ED, shape filling − experiment 3 | | | | | | |

| BC | Cl | Q | Kappa | MAE | F | AUC |
|---|---|---|---|---|---|---|
| Mixed | SVM | **89.181** | 0.856 | 0.264 | 0.892 | 0.948 |
| | Naive Bayes | 71.557 | 0.621 | 0.142 | 0.712 | 0.865 |
| | Random Forest | 82.781 | 0.770 | 0.155 | 0.826 | 0.958 |
| Black | SVM | **83.340** | 0.778 | 0.271 | 0.833 | 0.918 |
| | Naive Bayes | 66.853 | 0.558 | 0.166 | 0.658 | 0.828 |
| | Random Forest | 70.473 | 0.606 | 0.201 | 0.698 | 0.900 |
| White | SVM | **84.471** | 0.793 | 0.268 | 0.845 | 0.925 |
| | Naive Bayes | 66.210 | 0.549 | 0.169 | 0.648 | 0.835 |
| | Random Forest | 74.979 | 0.666 | 0.184 | 0.744 | 0.933 |
| AT, shape filling, centering – experiment 4 | | | | | | |
| Mixed | SVM | **80.732** | 0.743 | 0.275 | 0.807 | 0.899 |
| | Naive Bayes | 68.400 | 0.579 | 0.158 | 0.682 | 0.828 |
| | Random Forest | 74.393 | 0.658 | 0.182 | 0.741 | 0.914 |
| Black | SVM | **72.318** | 0.631 | 0.289 | 0.722 | 0.841 |
| | Naive Bayes | 65.411 | 0.539 | 0.173 | 0.650 | 0.797 |
| | Random Forest | 67.342 | 0.564 | 0.216 | 0.665 | 0.851 |
| White | SVM | **85.580** | 0.808 | 0.265 | 0.856 | 0.939 |
| | Naive Bayes | 76.438 | 0.686 | 0.118 | 0.763 | 0.886 |
| | Random Forest | 79.468 | 0.726 | 0.162 | 0.793 | 0.948 |
| ED, shape filling, centering – experiment 4 | | | | | | |
| Mixed | SVM | **92.403** | 0.899 | 0.259 | 0.924 | 0.964 |
| | Naive Bayes | 80.142 | 0.735 | 0.099 | 0.800 | 0.800 |
| | Random Forest | 85.543 | 0.807 | 0.125 | 0.854 | 0.965 |
| Black | SVM | **91.096** | 0.881 | 0.260 | 0.911 | 0.961 |
| | Naive Bayes | 73.566 | 0.648 | 0.132 | 0.732 | 0.850 |
| | Random Forest | 78.099 | 0.708 | 0.164 | 0.777 | 0.928 |
| White | SVM | **89.371** | 0.858 | 0.261 | 0.894 | 0.954 |
| | Naive Bayes | 82.068 | 0.761 | 0.090 | 0.820 | 0.903 |
| | Random Forest | 83.270 | 0.777 | 0.139 | 0.831 | 0.964 |

In case of the first experiment and analysis of original images, it can be observed that for all types of classifiers the accuracy for images with white background were higher than for other groups of images. The SVM classifier achieved the highest 88.54% correctness of classification, F-measure (0.886) and the area under the ROC curve (0.947), while Random Forest gave 69.79% of accuracy, F-measure at the level 0.695 and the area under the curve equaled 0.923. The lowest values can be observed for the Naive Bayes: accuracy - 56.25%, F-measure - 0.543 and the area under the curve - 0.713. The Kappa measure estimated the agreement between the original and obtained by the classifier belonging to the class as almost perfect for SVM (0.847), moderate for Random Forest (0.597), and fair for the Naive Bayes (0.417).

The images with black background had similar results as presented for images with white background. The approach with the highest results was SVM - the images

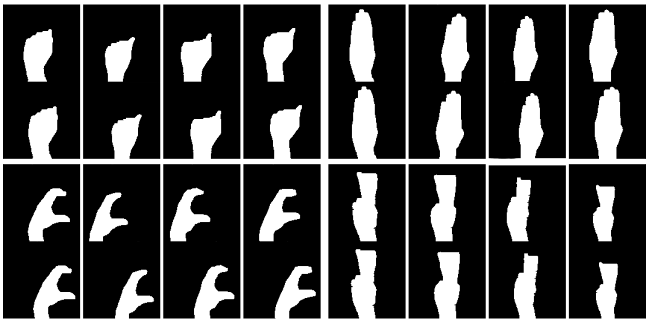


**Fig. 3.** Selected images after shape filling and edge detection versus centering, shape filling and edge detection

were the most precisely classified with 81.052% accuracy, F-measure 0.81 and the area under the ROC curve 0.891, while the Naive Bayes occurred to have the lowest results: accuracy equaled 38.947, F-measure 0.389 and the area under the ROC curve 0.597. The agreement of real and estimated class belonging was substantial in the case of SVM (0.747), moderate for Random Forest (0.522), while in the case of Naive Bayes it occurred slight.

In case of mixed background images the best results were obtained for Random Forest classifier (accuracy - 67.539; F-measure - 0.671; the area under the ROC curve - 0.866; Kappa coefficient - 0.567).

Analyzing the use of edge detection or adaptive thresholding in the second experiment, it can be observed that the application of these methods influenced the quality of classification, but not sufficiently. The biggest difference was achieved for edge detection, SVM classifier and mixed background images when the accuracy after edge detection increased by 28.78%, for Naive Bayes by 43.605%. A significant increase can be also mentioned for the black background images and Naive Bayes by about 33.502%. It can be concluded that pictures on a mixed background, which had previously dropped very poorly, now turned out to be the best, while the pictures on the black background from the middle position fell on the last. Differences between the best and the weakest result of the classification decreased.

Trying to achieve better classification results, the obtained hand shapes in the previous two experiments were filled (Fig. 3: the first and third row). Regardless of the edge detection approach final images do not differ practically. Unlike previous experiments, this time the best-classified objects are pictures taken against a white background. Each of the classifiers achieved the highest classification results

- 82.83% SVM, 67.21% Naive Bayes and 75.89% Random Forest. The best classifier for each type of background again turned out to be SVM. In the second place it was Random Forest, whose results were similar to SVM results for mixed and white background images. In case of images with black background, Random Forest managed poorer than the Naive Bayes, which in the remaining groups achieved the lowest results.

The last performed experiment was centering of hands, which underwent adaptive thresholding or edge detection and were filled previously. The received results in the form of images are presented in Figure 3 in the second and fourth row. In case of centering the filled hands after any approach of the edge detection, it can be noticed that it is impossible to define a group of images with the given background, on which the results were the best for each classifier. Whereas comparing classifiers in this paper in case of the SVM the results were the highest - the accuracy, the Kappa coefficient, F-measure or the area under the curve.

Statistical tests were carried out to determine the differences between the methods of image preprocessing. The variables were dependent and could not be described as normally distributed, thus the Friedman test for repeated blocks was applied and then post-hoc tests comparing a single pair of methods. Considering all preprocessing techniques a statistically significant difference in medians was found between this methods (Friedman F=6500; Kendall=0.546; p<0.0001). Similar results were obtained for edge detection (F=3000; Kendall=0.563; p<0.0001) and adaptive thresholding (F=3900; Kendall=0.594; p<0.0001). Comparing the background, the differences in the average results were also obrained: black (F=2000; Kendall=0.519; p<0.0001); mixed F=1200; Kendall=0.3167; p<0.0001); white - (F=2000; Kendall=0.5114; p<0.0001). The results of detailed comparisons are not presented because of the limited space - the only not significant difference was detected for white background images for experiment 2, 3 and 4 and Random Forest classifier.

## 4. Conclusions

Applying any of the edge detection techniques improved results, but the features obtained by the edge detection combined with the shape fill and centering allowed to achieve the highest classification accuracy regardless of the selected classifier.

SVM among chosen methods of classification turned out to be the best both for the whole set of images and subgroups of images differentiated by the background. The results were obtained for the whole set of parameters' mixture of edge detection techniques. In the next step only the results for arbitrarily chosen the best combination of two parameters for both methods will be checked and presented. The results are much better for any kind of a classifier comparing with the original images.

A proper choice of preprocessing selection method, then a reasonable selection of parameters' values and additional even simple techniques and corrections usage may lead to the binary images on which the selection or extraction methods can give improvements not only concerning the accuracy of classification systems but also the decision-making time, that is one of the most important issue in real-life applications.

## References

[1] Abeel T., P.Y.V., Saeys Y.: Java-ML: A Machine Learning Library, Journal of Machine Learning Research, 2009.

[2] Alzohairi R., Alghonaim R., Alshehri W. and Aloqeely S.: Image based Arabic Sign Language Recognition System. International Journal of Advanced Computer Science and Applications (IJACSA), 9(3), 2018.

[3] Canny J.: A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.

[4] Davies E. R.: Computer and Machine Vision: Theory, Algorithms and Practicalities, Academic Press, Waltham 1990, pp. 82-94.

[5] Główny Urząd Statystyczny, Departament Badań Społecznych: Stan zdrowia ludności Polski w 2009 r., Zakład Wydawnictw Statystycznych, Warszawa, 2011.

[6] Gonzalez R. C., Woods R. E. Digital Image Processing, Addison-Wesley Publishing Company, New Jersey 1992, pp. 595-602.

[7] Imagawa K., Lu S., and Igi S.: Color-based hands tracking system for sign language recognition. In Proc.: Third IEEE Intl. Conference on Automatic Face and Gesture Recognition, pp. 462–467, April 14-16, 1998.

[8] Java Machine Learning Library (Java-ML): [http://java-ml.sourceforge.net]

[9] Sung Kwan Kang, Mi Young Nam, Phill Kyu Rhee: Color Based Hand and Finger Detection Technology for User Interaction In Proc.: Intl. Conference on Convergence and Hybrid Information Technology, 2008. ICHIT '08.

[10] Kumar M., Saxena R.: Algorithm and technique on various edge detection: a survey, Signal & Image Processing: An Intl. Journal, vol. 4, no. 3, June 2013.

[11] S. Marcel: Hand Posture and Gesture Datasets, [http://www.idiap.ch/resource/gestures]

[12] Program Powszechnych Przesiewowych Badań Słuchu u Noworodków: [http://www.sluch.ump.edu.pl], [https://www.wosp.org.pl/medycyna/programy/badania-sluchu]

[13] Ravikiran J., Mahesh K., Mahishi S., Dheeraj R., Sudheender S., Pujari N.V.: Finger Detection for Sign Language Recognition. Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, Vol. I, IMECS 2009, Hong Kong, March 18-20, 2009.

[14] Rivera-Acosta M., Ortega-Cisneros S., Rivera J., Sandoval-Ibarra F.: American Sign Language Alphabet Recognition Using a Neuromorphic Sensor and an Artificial Neural Network. Sensors (Basel). 2017 Oct; 17(10): 2176.

[15] Sanduja V., Patial R.: Sobel Edge Detection using Parallel Architecture based on FPGA, International Journal of Applied Information Systems, vol. 3, no. 4, July 2012.

[16] Silanon K.: Thai Finger-Spelling Recognition Using a Cascaded Classifier Based on Histogram of Orientation Gradient Features. Computational Intelligence and Neuroscience. 2017: 9026375 (PMC5611514).

[17] Starner T. and Pentland A.: Real-time American Sign Language recognition from video using hidden Markov models. In Proc.of International Symposium on Computer Vision, pp. 265–270, 21-23 Nov. 1995.

[18] Świdziński M. i in.: Sytuacja osób głuchych w Polsce. Raport zespołu ds. głuchych przy Rzeczniku Praw Obywatelskich. Biuro Rzecznika Praw Obywatelskich, Warszawa, 2014.

[19] Vernon D.: Machine Vision, Prentice-Hall, 1991.

[20] Weka 3: Data Mining Software in Java, [https://www.cs.waikato.ac.nz/ml/weka/]

# WPŁYW PRZETWARZANIA WSTĘPNEGO OBRAZÓW NA ROZPOZNAWANIE ZNAKÓW ALFABETU MIGOWEGO

**Streszczenie** Artykuł dotyczy klasyfikacji wybranych liter alfabetu migowego w postaci obrazów. Badany jest wpływ na wyniki kilku metod przetwarzania wstępnego obrazów, w tym progowania adaptacyjnego oraz detekcji krawędzi. Dodatkowo sprawdzane jest wypełnianie znalezionych kształtów, a także centrowanie dłoni na obrazach. Jako metody klasyfikacji wybrane zostały: klasyfikator SVM z liniową funkcją jądrową, klasyfikator Naive Bayes oraz Random Forest. Jako miary jakości klasyfikacji raportowane są jakość klasyfikacji,miara F, pole pod krzywą ROC oraz współczynnik Kappa.

**Słowa kluczowe:** przetwarzanie wstępne, alfabet migowy, klasyfikacja

# THE IMPACT OF FILTERS ON THE QUALITY OF BINARIZATION FOR HANDWRITING IMAGES

Marcin Adamski, Krzysztof Pryzmont

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

**Abstract:** Filtration and binarization techniques are often used in handwriting recognition systems. These operations are performed as part of a stage called preprocessing, the result of which is passed to feature extraction and classification processes. Operations performed as part of the preprocessing are important because their result affects the outcome of the entire system. This paper focuses on the assessment of filtration techniques influence on the binarization of handwriting images. In the experiments, four filtration methods were tested with seven thresholding algorithms for various combinations of filtration and binarization parameters. The experiments were conducted on handwriting images selected from Document Binarization Competitions (DIBCO) datasets with ground truth images for the assessment of binarization correctness. The final evaluation was conducted based on the average of quality measures: F-measure, Accuracy, Relative Foreground Area Error and Region Non-uniformity.

**Keywords:** image filtering, binarization, handwriting

## 1. Introduction

Automatic text recognition from images of scanned documents is an important research area that has led to many successful applications. Whilst there are many available solutions for typed documents such as OCR tools (Optical Character Recognition), analysis of handwritten texts may still poses a challenge, especially when handwritten symbols are not well separated but form a continuous curve. Another source of difficulty when dealing with handwritten documents is the quality of input data. In many cases one has to work with low quality images that may be acquired from stained and damaged documents. Artifacts may be also introduced during acquisition or storage due to low scanning DPI and lossy compression.

Three important stages can be distinguished in handwriting recognition systems: preprocessing, feature extraction and classification (Fig. 1). The purpose of the initial

Advances in Computer Science Research, vol. 14, pp. 17-30, 2018. DOI: 10.24427/acsr-2018-vol14-0002

preprocessing is to facilitate the feature extraction by separation of text from background information that may be discarded. At this stage, one of the most commonly used techniques is image binarization. There are many methods for binarizing images, and new algorithms are still being developed. An example of this activity can be seen in Document Image Binarization Competition (DIBCO) [10,11] that is organized since 2009. The input to the binarization process can be a raw image that was acquired by the scanning process. However, initial filtering may be performed to reduce noise and enhance image quality. Examples of techniques that may be used for this task include: median filtering, average and Gaussian blur [3], anisotropic processing such as Perona-Malik diffusion [9].

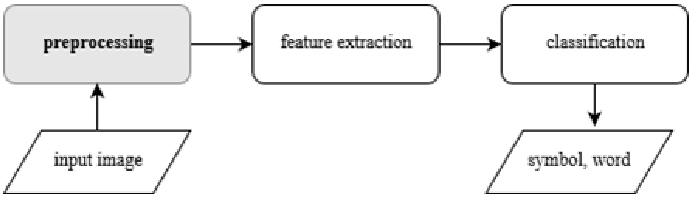


**Fig. 1.** Stages of text recognition system

Whilst there are many works in the literature on binarization algorithms, the effects of filtration methods executed prior binarization process on its result have not been widely studied. Some results related to this subject can be found in [4] and [15]. The authors of [4] investigated the effect of preprocessing and postprocesing on binarization. The preprocessing was performed using nose reduction technique based on wavelet transform applied to images of typed script. However, according to [4], this approach did not led to satisfactory results. In work [15] selection of filtering techniques were investigated. The results showed that none of the applied methods was best in all cases. The experiments were carried out on a mixed set of images that contained both handwritten and typed texts. In our investigation we focused on handwritten texts due to their distinct characteristics and studied filtering techniques such as Gaussian, Kuwahara and Perona-Malik that were not included in [15].

The aim of this investigation was to assess the impact of selected filtering techniques executed before binarization process on its result. The research was carried out on exemplary handwriting images from DIBCO competition that contain various types of artifacts. To assess the results the ground truth images were used together with various quality measures.

The article begins with a short review of selected filtering and binarization techniques. The next section contains description of the methodology used during experiments and is followed by presentation of the results and conclusions.

## 2. Image filtering

The image obtained from scanner or camera devices may contain noise negatively affecting subsequent stages of the text recognition. There are many available techniques that can be used for noise reduction. During selection of methods one should take into account the type of noise present in the image, as well as the characteristics of objects which should be preserved after filtration. The main problem is that, in addition to noise reduction, the filtering operation may remove information that is important at later stages of image processing. In case of images containing text this may include blurring text edges, merging separated structures or removing parts of script. In this work we selected widely used methods based on image filtering: median, Gaussian, Kuwahara and Perona-Malik filters. The selection was based on author's experience from earlier investigations related to preprocessing and verification of handwritten signatures [1] and results of other related works ([4,15]). The brief description of selected filters is given below.

**Median filter** [3] is a simple operation which for each pixel computes new value that is a median of pixels intensities covered by the median filter mask. The mask is usually a square region centered at the input pixel that is currently processed. This type of filter can be applied to remove "salt and pepper" type of noise. This operation usually preserves sharp edges but may also remove thin structures when the size of the mask exceeds their width. An example output of median filter can be seen in Fig. 3b.

**Gaussian filter** [3] is performed by convolving input image with filter mask that approximates 2D Gaussian function (1). This results in smoothing input image which may reduce image noise but also blurs object edges which is disadvantageous. The extent of smoothing is controlled by the sigma parameter and the mask size. Example output in presented in Fig. 3c.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

**Kuwahara filter** [8] is a nonlinear smoothing filter that allows to preserve object edges. In order to calculate the output value for selected input pixel, the Kuwahara filter computes mean and standard deviation in four overlapping regions (Fig. 2) in the neighborhood of selected pixel. The output value of the filter is the mean value from the region with smallest standard deviation (2). Example is given in Fig. 3d.

$$I_{out}\left(x,y\right) = \mu_k,\;\; k = \arg min_j \sigma_j\left(x,y\right) \tag{2}$$

where $\mu_i$ and $\sigma_i\left(x,y\right)$ are the mean and standard deviation of pixel intensities in i-th region.

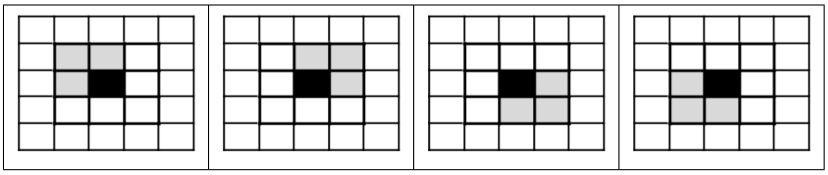


**Fig. 2.** Pixel's neighborhood regions analyzed in 3x3 Kuwahara filter.

**Perona-Malik diffusion** [9] is anisotropic smoothing filter that suppresses its effect near boundaries, therefore preserves sharp edges. The diffusion operation is based on modified heat equation (3) where $I$ denotes image, $t$ is time controlled by iteration process, $c$ is a function that controls rate of diffusion at particular point at position $(x,y)$.

$$\frac{\delta I}{\delta t} = c(x,y,t)\Delta I + \nabla c \nabla I \tag{3}$$

In order to reduce image smoothing near object boundaries the rate of diffusion may be constrained based on image gradient. In this work diffusion rate was controlled using function given in (4). The value of K parameter scales the gradient strength.

$$c(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2} \tag{4}$$

## 3. Image binarization

Image binarization is a basic segmentation technique used in many areas of image processing. The input to the binarization method is a grayscale image and the output is a binary map, where one of the values represents the background and the other is used for the segmented object. In this study the object (handwriting) is assigned

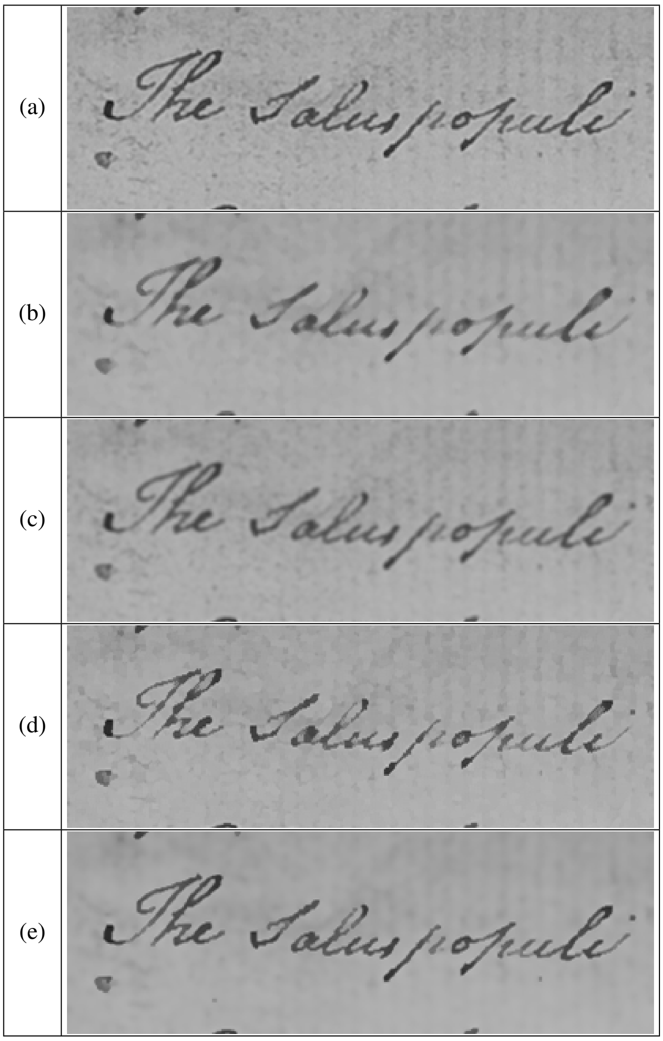


**Fig. 3.** Result of filtering methods applied to sample image from DIBCO dataset (a): median filter (b), Gaussian filter (c), Kuwahara filter (d), Perona-Malik diffusion (e)

a value of zero (shown as black) and the background pixels have the value of one (shown as white). In general, the binarization can be described as a thresholding operation that assigns a value of 1 or 0 to each input pixel by comparing it to the threshold value (5).

$$I_{out}(x,y) = \begin{cases} 1, & if\, I(x,y) > T \\ 0, & otherwise \end{cases} \tag{5}$$

The threshold value can be global – the same value is used for the whole image, or it may vary depending on location. This leads to two main categories of binarization algorithms: global and local methods. Many approaches to binarization have been proposed in the image processing literature. Comprehensive review that describes over 40 methods can be found in [14]. Example of image binarization is presented in Fig. 4. In this work we used seven approaches, three of which were global (Kapur-Sahoo-Wong, Otsu, Ridley-Calvard) and four local (Bradley, Niblack, Sauvola, White-Rohrer). They are briefly described in this section.

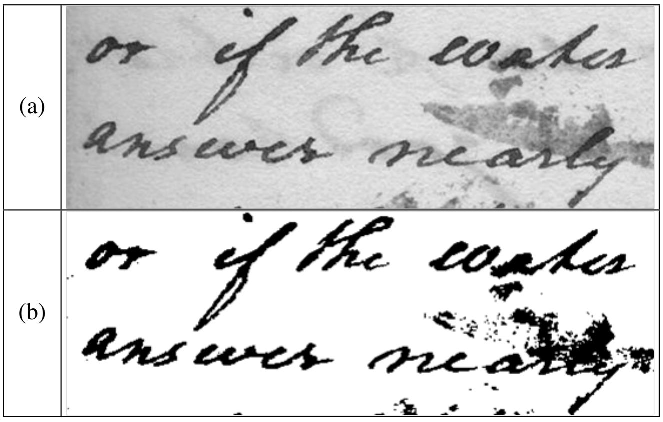


**Fig. 4.** Grayscale image (a) and its binarized version using Otsu metod (b)

**Kapur-Sahoo-Wong method** [5] is global technique that computes threshold value maximizing the sum of objects (foreground) and background distribution entropies. The distribution of foreground $p_f$ and background $p_b$ are computed based on normalized histogram values $p(i)$ as given by equation (6).

$$p_f = \sum_{i=0}^{t} p(i), \;\; p_b = \sum_{i=t+1}^{255} p(i) \tag{6}$$

Entropies of foreground and background are computed using equations (7) and (8), respectively.

$$H_f = -\sum_{i=0}^{t} \frac{p(i)}{p_f} \ln \frac{p(i)}{p_f}) \tag{7}$$

$$H_b = -\sum_{i=t+1}^{255} \frac{p(i)}{p_b} \ln \frac{p(i)}{p_b} \tag{8}$$

The global threshold is selected as the value that maximizes sum of entropies (9).

$$T_{opt} = \max_{t=0,\,...,\,255} H_f(t) + H_b(t) \tag{9}$$

**Otsu method** [7] is one of the most widely used binarization techniques. The threshold value is computed globally for the whole input image based on optimization procedure that maximizes intra-class variance between two classes of pixels defined by threshold criterion (1). The intra-class variance $\sigma_B^2$ is computed using equation (10).

$$\sigma_B^2 = P_o P_1 (\mu_0 - \mu_1)^2 \tag{10}$$

where $\mu_i$ is the mean value of input pixel values that are classified as foreground (i=0) and as background (i=1), and $P_i$ is the sum of normalized histogram values for the levels that belong to a particular class. The value of global threshold is given by (11).

$$T_{opt} = \arg max_{0<t<L-1} \left(\sigma_B^2(t)\right) \tag{11}$$

**Ridley-Calvard method** [12] represent iterative approach to computing global threshold value. At each iteration $i$ the threshold value calculated as the average of mean intensities of foreground and background pixels (12). Initial means can be computed as the mean of corner pixels (background) and the rest of the image (foreground). After computing new threshold, the mean values are updated based on pixel classification obtained using new threshold. The process continues until the threshold value does not change.

$$T(i) = \frac{\mu_b(i-1) + \mu_f(i-1)}{2} \quad i = 1,\, 2,\, \ldots \tag{12}$$

In **Bradley method** [2] the threshold value is computed locally based on the formula (13):

$$T(x,y) = \mu(x,y) \left(\frac{100-t}{100}\right) \tag{13}$$

The mean value $\mu(x,y)$ is computed using a window of fixed size. The parameter $t$ is selected heuristically. To compute the mean value, this method uses integration image which allows to perform computations in linear time.

**Niblack method** [6] is a simple local binarization technique where the threshold value is computed for each pixel based on mean and standard deviation values in a window centered at pixel's location (14).

$$t(x,y) = \mu(x,y) + k * \sigma(x,y), \;\; k < 0 \tag{14}$$

where (x,y) is pixel position; $\mu(x,y)$, $\sigma(x,y)$ are mean and standard deviation computed in local neighborhood window centered at $(x,y)$, $k$ is user defined parameter.

**Sauvola algoritm** [13] is modification of Niblack method where additional scaling factors are included to reduce sensitivity to background noise (15).

$$T(x,y) = \mu(x,y)\left[1 + k\left(\frac{\sigma(x,y)}{R} - 1\right)\right], \; k > 0 \tag{15}$$

where $R$ is dynamic range of standard deviation and k is user defined parameter. The size of the window and parameter $k$ are selected heuristically by the analysis of results obtained for particular class of images. The parameter $R$ can be computed separately for each image as a maximum value of standard deviation $\sigma(x,y)$.

**White-Rohrer method** [16] computes local threshold value using formula (16). The method was initially used for text images based on assumption that intensity of pixels comprising text differ significantly from pixel values of neighboring background.

$$I_{out}(x,y) = \begin{cases} 1, & I(x,y) \geq \frac{\mu(x,y)}{k}, \; k > 1 \\ 0, & otherwise \end{cases} \tag{16}$$

## 4. Evaluation procedure

To assess the impact of filtration methods on the results of binarization algorithms we conducted several experiments using images from the Document Image Binarization Competition (DIBCO) databases. Each sample image from DIBICO dataset has its ground truth version – a binary image with "ideal" binarization. By comparing ground truth image with the result of particular method its performance can be assessed. In this study we used eight images from DIBCO datasets (Fig. 5). Selected samples represent different types of distortions that may occur in scanned images of handwritten documents.

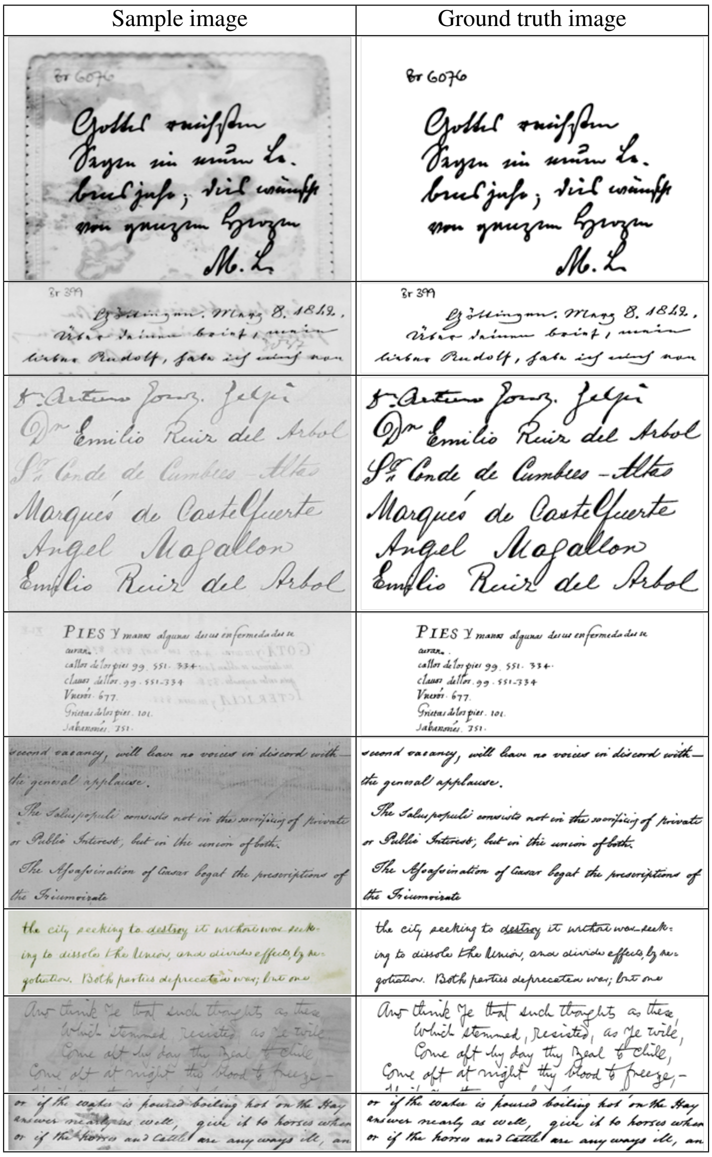

Sample image
Ground truth image

**Fig. 5.** Sample images and their ground truth versions

### 4.1 Measures

There are several measures that may be used to compare ground truth image with the output of binarization method. In our investigation we used evaluation functions proposed in works [11,14].

**F-measure** (17) is harmonic mean of recall and precision measures (18) . Its value is in range <0,1> where 1 represents perfect binarization.

$$FM = \frac{2 * Recall * Precision}{Recall + Precision} \tag{17}$$

$$Recall = \frac{TP}{TP + FN}, \ Precision = \frac{TP}{TP + FP} \tag{18}$$

where TP, FP, FN denote True Positive, False Positive and False Negative.

**Accuracy** is a ratio of correctly binarized pixels to total number of pixels in the image (19). Its value is in range <0,1> , 1 represents perfect binarization.

$$AC = \frac{TP + TN}{TP + TN + FN + FP} \tag{19}$$

where TN is True Negative.

**Relative foreground area error** represents relative accuracy of foreground pixels classification using notion of area (20). Its value is in range <0,1>, 0 represents perfect binarization.

$$RAE = \begin{cases} \frac{A_O - A_T}{A_O}, A_T < A_O \\ \frac{A_T - A_O}{A_T}, A_T \geq A_O \end{cases} \tag{20}$$

where $A_O$ is the area of objects in ground truth image, $A_T$ is the area of objects in binarized (test) image.

**Region nonuniformity** is computed using only grayscale image and its binarized version (21), without using ground truth reference. Well segmented image should have value of NU close to 0, where the value close to 1 means that foreground and background are hardly distinguishable.

$$NU = \frac{|F_T|}{|F_T + B_T|} \frac{\sigma_f^2}{\sigma^2} \tag{21}$$

where $\sigma^2$ and $\sigma_f^2$ represent standard deviations of pixel intensities for the whole image and foreground, respectively, $F_T$ is the number of foreground pixels, $B_T$ is the number of background pixels in binarized image.

In order to summarize all measures using one statistic we calculated arithmetic average. Due to different semantics of measure limits, RAE and NU values had to be reflected in final expression (22).

$$AV = \frac{FM + AC + (1 - RAE) + (1 - NU)}{4} \tag{22}$$

## 4.2 Method parametrization

The results of selected filtering and binarization methods depend on their parametrization. For certain methods one may find recommended values in the literature, however, the parametrization is usually chosen heuristically and may be suboptimal if the same setting is applied to images with different characteristics. To take this into account we conducted experiments with several possible configurations. For each image various combinations of filters and binarization methods parametrizations were verified. For final evaluation we selected only the best setting per image obtained for every filter and binarization algorithm pair.

The following settings were investigated during experiments:

- median, Gauss and Kuwahara filter size : 3x3, 5x5,
- number of iterations Perona-Malik filter: 5, 10,
- window size for local binarization techniques: 9, 15, 25, 45
- Bradley binarization t parameter: 7, 10, 15, 20
- White'a-Rohrer binarization k parameter: 1.2, 1.5, 2.0, 2.3
- Niblack binarization k parameter: -0.25, -0.5, -1.0, -1.5
- Sauvola binarization k parameter: 0.15, 0.3, 0.5, 0.7

As a result we verified 603 configurations per image giving the total number of evaluations 4824.

## 5. Results

In accordance with the adopted assumptions seven binarization algorithms were tested in combination with four filtration techniques. For comparison the images were also binarized without prior filtration. The experiments were carried out on 8 images from DIBICO datasets. To evaluate the effectiveness of analyzed methods their output was compared to ground truth images using the AV measure based on 4 separate coefficients. Table 1 shows obtained results. Each row of the table represents particular binarization algorithm, whilst separate filtration techniques are collected into columns. Each value was computed as an arithmetic average of results obtained for all images, where for each image only the best result among all parameterizations was used. The last row shows mean values for given filtration technique. The highest (best) values obtained for each binarization are given in bold font whilst the lowest (worst) results are grayed out. As can be seen from the Tab. 1, the worst results were obtained for the Kuwahara filtration method. This technique preserved the sharpness of the edges, however, it also narrowed handwriting lines – some of the pixels

**Table 1.** Evaluation of binarization and filtration techniques using AV measure

| | **Filtering** | | | | |
|---|---|---|---|---|---|
| **Binarization** | **Raw** | **Median** | **Gaussian** | **Kuwahara** | **Perona-Malik** |
| **Kapur-S-H** | 0,8580 | **0,8669** | 0,8599 | 0,8424 | **0,8669** |
| **Otsu** | 0,9002 | 0,9015 | **0,9057** | 0,8856 | 0,8993 |
| **Ridley-Calvard** | 0,9021 | 0,9028 | **0,9063** | 0,8860 | 0,8995 |
| **Bradley** | 0,9115 | 0,9149 | 0,9172 | 0,8976 | **0,9180** |
| **Niblack** | 0,8644 | 0,8647 | 0,8622 | 0,8702 | **0,8735** |
| **Savola** | 0,9136 | 0,9121 | 0,9129 | 0,9008 | **0,9143** |
| **White-Rohrer** | **0,8981** | 0,8956 | 0,8966 | 0,8896 | 0,8979 |
| **Average** | 0,8926 | 0,8941 | 0,8944 | 0,8817 | **0,8956** |

comprising script were incorrectly marked as a background. which had a negative impact on the final assessment. This is the effect of the Kuwahara algorithm, where each pixel is replaced with mean value of the neighbourhood section with smallest variance. As a result, the writing curve was partially replaced by background mean, which is characterized by lower variability. The Malik-Perona diffusion algorithm performed best on average, but the improvement did not occur for every binarization method. The local thesholding techniques: Bradley, Niblack and Savola obtained best results when applied to the output of Malik-Perona filtration. On the other hand, the global methods: Otsu and Ridley-Calvard did not achieved their highest performance. This may be due to the fact that the Malik-Perona algorithm also preserves artifacts edges, and therefore makes it more difficult to separate them from true script using one global threshold. In the case of local binarization methods, the threshold adapts to the surroundings and it is easier to separate the artifacts even if their edges become more pronounced. Otsu and Ridley-Calvard global methods were better suited to Gauss filtration. Kapur performed best with both median and Malik-Perona, however the performance it achieved was smallest among all of the analyzed algorithms. For the White-Rohrer method, the best result was obtained without the use of filtration at all. This indicates that the filtering image before binarization is not always helpful.

## 6. Conclusions

Based on the obtained results it can be concluded that the decision whether to use the initial filtering or which filtration algorithm should be selected depends on the binarization algorithm. Initial filtration, if well chosen for the binarization method, may improve the result, but if it is not appropriate, the effect may be opposite of what is expected. As part of further research, it is planned to examine other techniques of filtration and binarization. We also planning to increase the number of analyzed images and conduct separate experiments with images containing different types of artifacts.

## References

[1] Adamski, M. and Saeed, K.: Signature verification by only single genuine sample in offline and online systems, AIP Conference Proceedings, vol. 1738, no. 1, 2016.

[2] Bradley, D. and Roth, G.: Adaptive Thresholding using the Integral Image, Journal of Graphics Tools, vol. 12, no. 2, pp. 13-21, 2007.

[3] Gonzalez, R. C. and Woods, R. E.: Digital Image Processing, 3rd Edition, Prentice-Hall, Inc., 2007.

[4] Gupta, M. R., Jacobson, N. P., and Garcia, E. K.: OCR binarization and image pre-processing for searching historical documents, Pattern Recognition, vol. 40, no. 2, pp. 389-397, 2007.

[5] Kapur, J. N., Sahoo, P. K., and Wong, A. K. C.: A new method for gray-level picture thresholding using the entropy of the histogram, Computer Vision, Graphics, and Image Processing, vol. 29, no. 3, pp. 273-285, 1985.

[6] Niblack, W.: An introduction to image processing, Prentice-Hall, 1986.

[7] Otsu, N.: A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, 1979.

[8] Papari, G., Petkov, N., and Campisi, P.: Artistic Edge and Corner Enhancing Smoothing, IEEE Transactions on Image Processing, vol. 16, no. 10, pp. 2449-2462, 2007.

[9] Perona, P. and Malik, J.: Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629-639, 1990.

[10] Pratikakis, I., Zagoris, K., Barlas, G., and Gatos, B.: ICDAR2017 Competition on Document Image Binarization (DIBCO 2017), 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 1395-1403, 2017.

[11] Pratikakis, I., Zagoris, K., Barlas, G., and Gatos, B.: ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016), 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 619-623, 2016.

[12] Ridler, T. W. and Calvard, S.: Picture Thresholding Using an Iterative Selection Method, IEEE Transactions on Systems, Man, and Cybernetics, vol. 8, no. 8, pp. 630-632, 1978.

[13] Sauvola, J. and Pietikäinen, M.: Adaptive document image binarization, Pattern Recognition, vol. 33, no. 2, pp. 225-236, 2000.

[14] Sezgin, M. and Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, vol. 13, no. 1, pp. 13-20, 2004.

[15] Smith, E. H. B., Likforman-Sulem, L., and Darbon, J.: Effect of pre-processing on binarization, Document Recognition and Retrieval XVII, SPIE Electronic Imaging Symposium, vol. 7534, 2010.

[16] White, J. M. and Rohrer, G. D.: Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction, IBM Journal of Research and Development, vol. 27, no. 4, pp. 400-411, 1983.

# WPŁW FILTRÓW NA JAKOŚĆ BINARYZACJI OBRAZÓW PISMA ODRĘCZNEGO

**Streszczenie** Filtracja i binaryzacja są często stosowanymi technikami w systemach rozpoznawania pisma odręcznego. Operacje te są wykonywane w ramach etapu zwanego wstępnym przetwarzaniem , którego rezultat jest przekazywany do kolejnych etapów: ekstrakcji cech i klasyfikacji. Operacje wykonywane w ramach wstępnego przetwarzania są istotne ponieważ ich wynik wpływają na poprawność pracy całego systemu. W niniejszej pracy skupiono się nad oceną wpływu wyboru metody filtracji obrazu na efekt procesu binaryzacji dla obrazów z pismem odręcznym. W eksperymentach zbadano 4 metody filtracji w połączeniu z 7 metodami progowania dla różnych kombinacji parametrów tych metod. Do eksperymentów użyto wybrane obrazy z pismem odręcznym z baz konkursów binaryzacji dokumentów DIBCO oraz obrazy referencyjne do oceny poprawności binaryzacji. Ocenę wykonano na bazie średniej z miar F-measure, Accuracy, Relative Foreground Area Error, Region nonuniformity.

**Słowa kluczowe:** filtracja obrazu, binaryzacja obrazu, pismo odręczne

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

# FEATURE SELECTION FOR PROGNOSTIC MODELS BY LINEAR SEPARATION OF SURVIVAL GENETIC DATA SETS

Leon Bobrowski[1,2], Tomasz Łukaszuk[1]

[1] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Institute of Biocybernetics and Biomedical Engineering PAS, Warsaw, Poland

**Abstract:** Designing regression models based on high dimensional (e.g. genetic) data sets through exploring linear separability problem is considered in the paper. The linear regression model designing has been reformulated here as the linear separability problem. Exploring the linear separability problem has been based on minimization of the convex and piecewise linear (CPL) criterion functions. The minimization of the CPL criterion functions was used not only for estimating the prognostic model parameters, but also for most effective selecting feature subsets (model selection) in accordance with the relaxed linear separability (RLS) method. This approach to designing prognostic models has been used in experiments both with synthetic multivariate data, and with genetic data sets containing censored values of dependent variable. The quality of the prognostic models resulting from the linear separability postulate has been evaluated by using the measure of the model discrepancy and the estimated classification error rate. In order to reduce the bias of the evaluation, the value of the model discrepancy and the classification error have been computed in different feature subspaces, in accordance with the cross-validation procedure. A series of new experiments described in this paper shows that the designing of regression models can be based on the linear separability principle. More specifically, the high-dimensional genetic sets with censored dependent variable can be used in designing procedure. The proposed measure of prognostic model discrepancy can be effectively used in the search for the optimal feature subspace and for selecting the linear regression model.

**Keywords:** data mining, interval regression, model selection, relaxed linear separability

## 1. Introduction

Multivariate regression analysis includes many techniques aimed at modelling the linear relationship between dependent variable and independent variables. In this case,

Advances in Computer Science Research, vol. 14, pp. 31-54, 2018. DOI: 10.24427/acsr-2018-vol14-0003

the value of a dependent variable is predicted to be the linear combination of some independent variables. The linear regression function is based on a finite number of unknown parameters that are estimated from the learning data set. The least squares method of the parameters estimation is commonly used in the regression analysis [11].

It has been recently demonstrated that the task of linear regression model designing can be formulated as a linear separability problem [3,5]. The linear separability problem has been investigated for many years in the context of the theory of neural networks and pattern recognition [1,9]. We use the convex and piecewise linear (*CPL*) criterion functions in our approach to the linear separability problem [2]. The basis exchange algorithms, which are similar to the linear programming, allow to efficiently find the minimal value of the *CPL* criterion function [6]. The parameters that create the minimum of an adequate *CPL* criterion function can be also used in the definition of the optimal regression models.

The *perceptron criterion function* belongs to the family of the *CPL* criterion functions [2]. The perceptron criterion function was modified by adding a regularization component for the purpose of the feature subset selection in accordance with the relaxed linear separability (*RLS*) method [4]. This regularization component used in the *RLS* method has a similar structure to those used in the *Lasso regression* [14]. The relaxed linear separability (*RLS*) method of feature subset selection is based on minimization of the modified perceptron criterion function. This method allows for a successive reduction of unnecessary features while preserving the linear separability of the learning sets.

Prognostic models in the area of survival analysis are designed on the basis of the so-called *censored* data sets. The Cox model plays a fundamental role in the survival analysis [12]. The modified perceptron criterion function can be also used for designing prognostic models (selection) on the basis of censored data sets. The possibility of the regression (prognostic) models selection from high-dimensional genetic data set with censored dependent variable is considered in the paper. Particular attention is paid to evaluating the quality of prognostic models obtained in this way.

The novelties introduced in the paper include: a) introduction of a new prognostic model quality measure, i.e. *discrepancy* coefficient, b) the series of new experiments proving the correctness of the concept adopted.

# 2. Methods

## 2.1 Different types of regression learning sets

Multivariate regression models are based on linear (affine) transformations of $n$-dimensional feature vectors $\mathbf{x}[n]$ taken from a given feature space $F[n]$ ($\mathbf{x}[n] \in F[n]$) on points $t$ on the line ($t \in R^1$):

$$t(\mathbf{x}[n]) = \mathbf{w}[n]^T\mathbf{x}[n] + w_0 \tag{1}$$

where $\mathbf{w}[n] = [w_1, ..., w_n]^T \in R^n$ is the parameters' (*weight*) vector and $w_0$ is the threshold (*intercept coefficient*) ($w_0 \in R^1$).

Properties of the model (1) depend on the choice of the parameters $\mathbf{w}[n]$ and $w_0$. The weights $w_i$ and the threshold $w_0$ are estimated from regression learning sets. In the case of classical regression analysis the learning sets are structured as follows:

$$\begin{aligned} C_0 = \{\mathbf{x}_j[n]; t_j\} = \{x_{j1}, ..., x_{jn}; t_j\}, \\ where\ \ j = 1, ..., m_0 \end{aligned} \tag{2}$$

Each object $O_j$ in the set $C_0$ is characterized by values $x_{ji}$ of *n independent variables* (*features*) $X_i$, and by the observed value $t_j$ ($t_j \in R^1$) of the *dependent* (*target*) *variable* $T$. Components $x_{ji}$ of the $j$-th feature vector $\mathbf{x}_j[n]$ could be treated as numerical results of $n$ standardized examinations of the given object $O_j$ ($x_{ji} \in \{0,1\}$ or $x_{ji} \in R^1$). Each feature vector $\mathbf{x}_j[n]$ can be also treated as a point in the $n$-dimensional feature space $F[n]$ [9].

In case of *classical regression*, the parameters $\mathbf{w}[n]$ and $w_0$ are estimated on the basis of set $C_0$ (2), in accordance with the method of *least squares* in such a way that the sum of the squared differences $(t_j - \hat{t_j})^2$ between the observed target variable $y_j$ and the modelled variable $\hat{t_j} = \mathbf{w}[n]^T\mathbf{x}_j[n] + w_0$ (1) is minimal [11].

In case of *interval regression* learning sets $C_I$, additional knowledge about values $t_j$ of the dependent variable $T$ of particular objects $O_j$ is represented by the intervals $[t_j^-, t_j^+]$ ($t_j^- < t_j^+$) instead of exact values $t_j$ (1) [8,10]:

$$C_I = \{(\mathbf{x}_j[n], [t_j^-, t_j^+]), where\ j \in J_I\} \tag{3}$$

where $J_I$ is the set of indices $j$ of $m_R$ objects $O_j$ (feature vectors $\mathbf{x}_j[n]$), $t_j^-$ is the lower bound ($t_j^- \in R^1$) and $t_j^+$ is the upper bound ($t_j^+ \in R^1$) of unknown value $t_j$ ($t_j^- < t_j < t_j^+$) of the target variable $T$, which accompanies the $j$-th feature vector $\mathbf{x}_j[n]$.

Let us introduce the *right censored* set $C_R$ and the *left censored* set $C_L$:

$$C_R = \{\mathbf{x}_j[n], [t_j^-, +\infty)\}, where\ j \in J_R \tag{4}$$

and

$$C_L = \{\mathbf{x}_j[n], (-\infty, t_j^+]\}, where\ j \in J_L \tag{5}$$

The set $J_R$ contains the indices $j$ of $m_R$ objects $O_j$ (feature vectors $\mathbf{x}_j[n]$) which are characterized by *right censored* values of the *dependent* variable $T$ [12]. Similarly, the set $J_L$ contains the indices $j$ of such objects $O_j$, that are characterized by *left censored* values of the *dependent* variable $T$. It is assumed, that the sets $C_R$ and $C_L$ are disjoined ($C_R \cap C_L = \emptyset$). The censored sets $C_R$ (4) or $C_L$ (5) can be treated as a special type of the interval regression set $C_I$ (3) in which either $t_j^+ = +\infty$ or $t_j^- = -\infty$.

The classical learning set $C_0$ (2) can be transformed into the interval learning set $C_I$ (3) through introducing artificial boundary values $t_j^- = t_j - \varepsilon$ and $t_j^+ = t_j + \varepsilon$, where $\varepsilon$ is a small positive parameter (*margin*) ($\varepsilon > 0$):

$$C_I' = \{\mathbf{x}_j[n], [t_j - \varepsilon, t_j + \varepsilon]\}, where\ j = 1, ..., m_0 \tag{6}$$

The following linear inequalities can be expected in case of prognostic model (1) designing on the basis of the interval learning set $C_I$ (3):

$$(\forall j \in J_I) \quad t_j^- < \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 < t_j^+ \tag{7}$$

or equivalently

$$\begin{aligned} (\forall j \in J_I) \quad & \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 - t_j^- > 0 \\ and\ & \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 - t_j^+ < 0 \end{aligned} \tag{8}$$

The feature vectors $\mathbf{x}_j[n]$ belonging to the censored sets $C_R$ (4) or $C_L$ (5) can be linked in a similar way to the below linear inequalities:

$$(\forall j \in J_R) \quad \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 - t_j^- > 0 \tag{9}$$

and

$$(\forall j \in J_L) \quad \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 - t_j^+ < 0 \tag{10}$$

We can note that censoring of some feature vector $\mathbf{x}_j[n]$ ($t_j^+ = +\infty$ or $t_j^- = -\infty$) results in removing one inequality from the set of inequalities (8).

## 2.2 Linear separability of the positive set $Z^+[n+2]$ and the negative set $Z^-[n+2]$

The interval learning set $C_I$ (3) can be represented as the below sum of the disjoined subsets $C_I'$, $C_R$ (4), and $C_L$ (5):

$$C_I = C_I' \cup C_R \cup C_L \tag{11}$$

where the subset $C_I'$ contains such intervals $[t_j^-, t_j^+]$ that are not censored (the constraints $t_j^-$ and $t_j^+$ are finite):

$$C_I' = \{(\mathbf{x}_j[n], [t_j^-, t_j^+]) : -\infty < t_j^- < t_j^+ < +\infty\} \tag{12}$$

The subsets $C_R$ (4), $C_L$ (5) and $C_I'$ (12) are used in defining the *augmented feature vectors* $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ based on the linear inequalities (8), (9), and (10):

$$(\forall \mathbf{x}_j[n] \in C_I' \cup C_R) \quad \mathbf{z}_j^+[n+2] = [\mathbf{x}_j[n]^T, 1, -t_j^-]^T \tag{13}$$

and

$$(\forall \mathbf{x}_j[n] \in C_I' \cup C_L) \quad \mathbf{z}_j^-[n+2] = [\mathbf{x}_j[n]^T, 1, -t_j^+]^T \tag{14}$$

Let us introduce the *positive set* $\mathbf{Z}^+[n+2]$ and the *negative set* $\mathbf{Z}^-[n+2]$ which are composed of $(n+2)$-dimensional vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14):

$$\begin{aligned} &\mathbf{Z}^+[n+2] = \{\mathbf{z}_j^+[n+2]\} \; \textit{and} \\ &\mathbf{Z}^-[n+2] = \{\mathbf{z}_j^-[n+2]\} \end{aligned} \tag{15}$$

**Definition 1.** *The positive set* $\mathbf{Z}^+[n+2]$ *and the negative set* $\mathbf{Z}^-[n+2]$ *(15) are linearly separable, if and only if there exists a parameter vector* $\mathbf{v}'[n+2]$ *(*$\mathbf{v}'[n+2] \in R^{n+2}$*), for which all the below inequalities are fulfilled [3,5]:*

$$\begin{aligned} &(\exists \mathbf{v}'[n+2]) \; (\forall \mathbf{z}_j^+[n+2] \in \mathbf{Z}^+[n+2]) \\ &\qquad \mathbf{v}'[n+2]^T \mathbf{z}_j^+[n+2] \geq 1 \\ &\textit{and} \quad (\forall \mathbf{z}_j^-[n+2] \in \mathbf{Z}^-[n+2]) \\ &\qquad \mathbf{v}'[n+2]^T \mathbf{z}_j^-[n+2] \leq 1 \end{aligned} \tag{16}$$

The parameter vector $\mathbf{v}'[n+2]$ defines the below hyperplane $H(\mathbf{v}'[n+2])$ in the feature space $F[n+2]$ ($\mathbf{z}[n+2] \in F[n+2]$):

$$H(\mathbf{v}'[n+2]) = \{\mathbf{z}[n+2] : \mathbf{v}'[n+2]^T \mathbf{z}[n+2] = 0\} \tag{17}$$

If all the inequalities (16) are fulfilled, then the hyperplane $H(\mathbf{v}'[n+2])$ (17) separates the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15). This means that each augmented feature vector $\mathbf{z}_j^+[n+2]$ (13) from the set $\mathbf{Z}^+[n+2]$ is situated on the *positive side* of the hyperplane $H(\mathbf{v}'[n+2])$ (17) ($\mathbf{v}'[n+2]^T\mathbf{z}_j^+[n+2] > 0$) and each augmented feature vector $\mathbf{z}_j^-[n+2]$ (14) from the set $\mathbf{Z}^-[n+2]$ is situated on the *negative side* of this hyperplane ($\mathbf{v}'[n+2]^T\mathbf{z}_j^+[n+2] < 0$).

The desirable inequalities (8), (9), (10) can be represented as the linear separability problem (16), if the parameter vector $\mathbf{v}[n+2]$ has the below structure [3]:

$$\mathbf{v}[n+2] = [v_1, ..., v_{n+2}]^T = [\mathbf{w}[n]^T, w_0, \beta]^T \tag{18}$$

where $\beta$ is the *interval parameter* ($\beta \in R^1$).

The parameter vector $\mathbf{v}[n+2]$ (18) allows to define the below prognostic model:

$$y(\mathbf{x}[n]) = (\mathbf{w}[n]/\beta)^T\mathbf{x}[n] + w_0/\beta \tag{19}$$

The following *Lemma* can be proved [3]:

**Lemma 1.** *All the desirable inequalities (8), (9), (10) are fulfilled by the prognostic model $y(\mathbf{x}[n])$ (19) defined by the parameters vector $\mathbf{v}'[n+2] = [\mathbf{w}'[n]^T, w_0', \beta']$ (18) if and only if the hyperplane $H(\mathbf{v}'[n+2])$ (17) fully separates (16) the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15).*

If the number $n$ of features $X_i$ is larger than the number $m$ of the vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14) in the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15), then such sets are usually linearly separable [9]. The most interesting are cases when linear separability (16) occurs in the opposite circumstances, when the number m of feature vectors is large in comparison to the number n of features.

The concept of *linear separability* has been used for many years in the theory of neural networks and in pattern recognition methods. The linear separability has been used in the proof of the convergence of the error-correction algorithm - classical learning algorithm of neural networks [9]. The optimal linear classifiers in pattern recognition can be designed through exploration of the linear separability of the learning sets [2].

### 2.3 Convex and piecewise linear (CPL) criterion function defined on the positive set $\mathbf{Z}^+[n+2]$ and the negative set $\mathbf{Z}^-[n+2]$

The positive set $\mathbf{Z}^+[n+2]$ and the negative set $\mathbf{Z}^-[n+2]$ (15) are composed of the $(n+2)$ - dimensional vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14), adequately. The sets

$\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15) are linearly separable if and only if the inequalities (16) are fulfilled.

The below convex and piecewise-linear (*CPL*) penalty functions $\varphi_j^+(\mathbf{v}[n+2])$ and $\varphi_j^-(\mathbf{v}[n+2])$ are introduced for solving inequalities (16):

$$\begin{array}{ll} (\forall \mathbf{z}_j^+[n+2]) & \\ & 1-\mathbf{v}[n+2]^T\mathbf{z}_j^+[n+2] \\ \varphi_j^+(\mathbf{v}[n+2]) = & \quad if\ \mathbf{v}[n+2]^T\mathbf{z}_j^+[n+2] < 1 \\ & 0 \\ & \quad if\ \mathbf{v}[n+2]^T\mathbf{z}_j^+[n+2] \geq 1 \end{array} \tag{20}$$

$$\begin{array}{ll} (\forall \mathbf{z}_j^-[n+2]) & \\ & 1+\mathbf{v}[n+2]^T\mathbf{z}_j^-[n+2] \\ \varphi_j^-(\mathbf{v}[n+2]) = & \quad if\ \mathbf{v}[n+2]^T\mathbf{z}_j^-[n+2] > -1 \\ & 0 \\ & \quad if\ \mathbf{v}[n+2]^T\mathbf{z}_j^-[n+2] \leq -1 \end{array} \tag{21}$$

The *perceptron criterion function* $\Phi(\mathbf{v}[n+2])$ is defined as the weighted sum of the penalty functions $\varphi_j^+(\mathbf{v}[n+2])$ (20) and $\varphi_j^-(\mathbf{v}[n+2])$ (21) [2]:

$$\Phi(\mathbf{v}[n+2]) = \sum_j \alpha_j^+\varphi_j^+(\mathbf{v}[n+2]) + \sum_j \alpha_j^-\varphi_j^-(\mathbf{v}[n+2]) \tag{22}$$

where non-negative parameters $\alpha_j^+$ ($\alpha_j^+ > 0$) determine the *importance* of particular vectors $\mathbf{z}_j^+[n+2]$ (13) and parameters $\alpha_j^-$ ($\alpha_j^+ > 0$) determine the *importance* of particular vectors $\mathbf{z}_j^-[n+2]$ (14). Standard values of the parameters $\alpha_j^+$ and $\alpha_j^-$ can be provided as follows [2]:

$$\begin{array}{ll} (\forall \mathbf{z}_j^+[n+2]) & \alpha_j^+ = 1/(2m^+)\ and \\ (\forall \mathbf{z}_j^-[n+2]) & \alpha_j^- = 1/(2m^-) \end{array} \tag{23}$$

where $m^+$ is the number of the vectors $\mathbf{z}_j^+[n+2]$ (13) and $m^-$ is the number of the vectors $\mathbf{z}_j^-[n+2]$ (14).

The optimal vector $\mathbf{v}^*[n+2]$ constitutes the global minimum of the *CPL* criterion function $\Phi(\mathbf{v}[n+2])$ (22):

$$(\forall \mathbf{v}[n+2])\ \Phi(\mathbf{v}[n+2]) \geq \Phi(\mathbf{v}^*[n+2]) = \Phi^* \geq 0 \tag{24}$$

where $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$, and $\mathbf{w}^*[n] = [w_1^*, ..., w_n^*]^T$.

The basis exchange algorithms which are similar to linear programming, allow to find the minimal value $\Phi^*$ (24) of the function $\Phi(\mathbf{v}[n+2])$ (22) and the optimal parameters $\mathbf{v}^*[n+2]$ efficiently, even in case of large, multidimensional data sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15) [3,5].

The following remarks describe useful properties of the minimal value $\Phi^* = \Phi(\mathbf{v}^*[n+2])$ (24) of the perceptron criterion function $\Phi(\mathbf{v}[n+2])$ (22) [2]:

*Remark 1.* detection of the linear separability The minimal value $\Phi^* = \Phi(\mathbf{v}^*[n+2])$ (24) of the criterion function $\Phi(\mathbf{v}[n+2])$ (22) with the standard values (23) of the parameters $\alpha_j^+$ and $\alpha_j^-$ is contained in the interval $< 0, 1 >$

$$0 \leq \Phi^* \leq 1 \tag{25}$$

where $\Phi^* = 0$ if and only if the positive set $\mathbf{Z}^+[n+2]$ and the negative set $\mathbf{Z}^-[n+2]$ (15) are linearly separable (16).

*Remark 2.* the positive monotonicity property The omission of an arbitrary pair $(\mathbf{x}_j[n], [y_j^-, y_j^+])$ from the learning set $C_I$ (3) can not increase the value of $\Phi^*$ (24) (the value of $\Phi^*$ usually *decreases*).

*Remark 3.* the negative monotonicity property The omission of any of the component $x_{ji}$ (feature $X_i$) in all the $m$ feature vectors $\mathbf{x}_j[n] = [x_{j1}, ..., x_{jn}]^T$ (3) can not reduce the value of $\Phi^*$ (24) (the value of $\Phi^*$ usually *increases*).

*Remark 4.* the invariancy property The minimal value $\Phi^*$ (24) of the criterion function $\Phi(\mathbf{v}[n+2])$ (22) does not depend on linear (affine), nonsingular transformations of feature vectors $\mathbf{x}_j[n]$ (3):

$$\begin{gathered} if\ (\forall j \in \{1, ..., m\})\ \mathbf{x}'_j[n] = \mathbf{A}\mathbf{x}_j[n] + \mathbf{b}[n], \\ where\ \mathbf{A}^{-1}\ exists,\ then\ \Phi^*_{x'} = \Phi^*_x \end{gathered} \tag{26}$$

where $\mathbf{b}[n]$ is a constant vector ($\mathbf{b}[n] \in R^n$), and $\Phi^*_{x'}$ is the minimal value (24) of the perceptron criterion function $\Phi_{x'}(\mathbf{v}[n+2])$ (22) defined on elements of the learning set $C'_I = \{(\mathbf{x}'_j[n], [y_j^-, y_j^+]),\ where\ j \in J_I\}$ (3).

The optimal parameters $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$ (24) that constitute the minimal value $\Phi^* = \Phi(\mathbf{v}^*[n+2])$ (24) of the criterion function $\Phi(\mathbf{v}[n+2])$ (22) are used in the definition of the below *CPL* prognostic model [5]:

$$t^*(\mathbf{x}[n]) = (\mathbf{w}^*[n]^T\mathbf{x}[n] + w_0^*)/\beta^* \tag{27}$$

**Lemma 2.** *If the minimal value* $\Phi^* = \Phi(\mathbf{v}^*[n+2])$ *(24) of the criterion function* $\Phi(\mathbf{v}[n+2])$ *(22) is equal to zero (*$\Phi^* = 0$*) in the extreme point* $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$ *with* $\beta^* > 0$*, then the optimal prognostic model (27) fulfills all the inequalities (7):*

$$(\forall j \in J_I)\; t_j^- < (\mathbf{w}^*[n]/\beta^*)^T \mathbf{x}_j[n] + w_0^*/\beta^* < t_j^+ \tag{28}$$

The above conditions can be proved directly from the linear separability inequalities (16). If the minimal value $\Phi^*$ (24) is greater than zero ($\Phi^* > 0$) in the extreme point $\mathbf{v}^*[n+2]$, then the optimal prognostic model (27) satisfies the majority but not all the inequalities (28) [5].

### 2.4 The modified criterion function $\Psi(\mathbf{v}[n+2])$

The perceptron criterion function $\Phi(\mathbf{v}[n+2])$ (22) has been modified in order to allow selecting features task though including *feature penalty functions* $\phi_i(\mathbf{v}[n+2])$ and the *costs* $\gamma_i$ ($\gamma_i \geq 0$) related to particular features $X_i$ [4]. The feature penalty functions $\phi_i(\mathbf{v}[n+2])$ are defined in the below manner:

$$\begin{aligned}&(\forall i \in \{1,...,n\})\\ &\quad \phi_i(\mathbf{v}[n+2]) = |\mathbf{e}_i[n+2]^T \mathbf{v}[n+2]| = |w_i|\end{aligned} \tag{29}$$

The modified criterion function $\Psi(\mathbf{v}[n+2])$ is the sum of the basic function in the form of perceptron criterion function $\Phi(\mathbf{v}[n+2])$ (20) and regularization components with the penalty functions $\phi_i(\mathbf{v}[n+2])$ [4]:

$$\begin{aligned}&\Psi_\lambda(\mathbf{v}[n+2]) =\\ &\quad = \Phi(\mathbf{v}[n+2]) + \lambda \textstyle\sum_{i \in \{1,...,n\}} \gamma_i \phi_i(\mathbf{v}[n+2]) =\\ &\quad = \Phi(\mathbf{v}[n+2]) + \lambda \textstyle\sum_{i \in \{1,...,n\}} \gamma_i |w_i|\end{aligned} \tag{30}$$

where $\lambda$ ($\lambda \geq 0$) is the *cost level*, and $\gamma_i$ are feature costs ($\gamma_i \geq 0$).

The standard assumption about the *feature costs* $\gamma_i$ is that these costs are equal to one:

$$(\forall i \in \{1,...,n\})\; \gamma_i = 1 \tag{31}$$

The modified criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) is used in the *relaxed linear separability* (*RLS*) method of feature subset selection [2]. The regularization component $\lambda \sum \gamma_i |w_i|$ used in the modified criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) is similar to these used in the *Lasso* method [14]. The *Lasso* method was developed as a part of regression analysis for the model selection [14]. The main difference between the *Lasso* and the *RLS* methods is the type of the basic criterion function. This difference affects the computational techniques used to minimize the modified criterion functions. The perceptron criterion function $\Phi(\mathbf{v}[n+2])$ (22) plays fundamental role in case of the *RLS* method. Both the basic criterion $\Phi(\mathbf{v}[n+2])$ (22), as well as the penalty functions $\phi_i(\mathbf{v}[n+2])$ (29) are convex and piecewise-linear (*CPL*). As a result, the modified criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) is also convex and piecewise-linear (*CPL*). The basis exchange algorithms allow to find efficiently the optimal vector of parameters (*vertex*) $\mathbf{v}_\lambda^*[n+2]$ constituting the minimum of the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) with the cost level $\lambda$:

$$\begin{array}{l}(\exists \mathbf{v}_\lambda^*[n+2])\ (\forall \mathbf{v}[n+2]) \\ \quad \Psi_\lambda(\mathbf{v}[n+2]) \geq \Psi_\lambda(\mathbf{v}_\lambda^*[n+2]) = \Psi_\lambda^*\end{array} \tag{32}$$

where $\mathbf{v}_\lambda^*[n+2] = [\mathbf{w}_\lambda^*[n]^T, w_{\lambda 0}^*, \beta_\lambda^*]^T = [w_{\lambda 1}^*, ..., w_{\lambda n}^*, w_{\lambda 0}^*, \beta_\lambda^*]^T$ (24). In the *RLS* method the optimal parameters $w_{\lambda i}^*$ are used in the feature reduction rule below:

$$(w_{\lambda i}^* = 0)\ => \ (the\ feature\ X_i\ is\ reduced) \tag{33}$$

We can remark that the features $X_i$ which have the weights $w_{\lambda i}^*$ equal to zero ($w_{\lambda i}^* = 0$) in the optimal vertex $\mathbf{v}_\lambda^*[n+2]$ (32) can be reduced (33) without changing the optimal prognostic model $y^*(\mathbf{x}[n])$ (27) which is defined by the parameters $\mathbf{v}_\lambda^*[n+2]$ (32).

It can be proved that the vector $\mathbf{v}_\lambda^*[n+2]$ which constitutes the minimum (32) of the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) can be located on the optimal vertex $\mathbf{v}_k^*[n+2]$ ($\mathbf{v}_\lambda^*[n+2] = \mathbf{v}_k^*[n+2]$) linked to some basis $\mathbf{B}_k^*[n+2]$ in the $(n+2)$ - dimensional feature space $F[n+2]$ [2]:

$$\mathbf{B}_k^*[n+2]\mathbf{v}_k^*[n+2] = \delta_k^*[n+2] \tag{34}$$

where $\mathbf{B}_k^*[n+2]$ is the non-singular matrix (*basis*) with rows consisting of $(n+2)$ linearly independent vectors $\mathbf{z}_j^+[n+2]$ $(j \in J_k^+)$, $\mathbf{z}_j^-[n+2]$ $(j \in J_k^-)$ or by unit vectors $\mathbf{e}_i[n+2]$ $(i \in I_k^0)$, and $\gamma_k^*[n+1]$ is the *margin vector* with components equal to 1, $-1$ or 0, adequately to the below equations fulfilled in the vertex $\mathbf{v}_k^*[n+2]$:

$$\begin{array}{l}(\forall j \in J_k^+)\ \mathbf{z}_j^+[n+2]^T\mathbf{v}_k^*[n+2] = 1,\ and \\ (\forall j \in J_k^-)\ \mathbf{z}_j^-[n+2]^T\mathbf{v}_k^*[n+2] = -1,\ and \\ (\forall i \in I_k^0)\ \mathbf{e}_i[n+2]^T\mathbf{v}_k^*[n+2] = 0\end{array} \tag{35}$$

where $J_k^+$, $J_k^-$ and $I_k^0$, are the sets of indices of the basis vectors $\mathbf{z}_j^+[n+2]$, $\mathbf{z}_j^-[n+2]$, and $\mathbf{e}_i[n+2]$, adequately.

*Remark 5.* The features $X_i$ which are linked to the unit vectors $\mathbf{e}_i[n+2]$ ($i \in I_k^0$) in the optimal basis $\mathbf{B}_k^*[n+2]$ (34) can be reduced (33) in the related vertex $\mathbf{v}_k^*[n+2] = [\mathbf{w}_k^*[n]^T, w_{k0}^*, \beta_k^*]^T = [w_{k1}^*, ..., w_{kn}^*, w_{k0}^*, \beta_k^*]^T$.

The Remark 5 can be justified by the below implication (33) [4]:

$$\begin{aligned} &(\forall i \in I_k^0) \\ &(\mathbf{e}_i[n+2]^T \mathbf{v}_k^*[n+2] = 0) => (w_{ki}^* = 0) => \\ &=> (the\ feature\ X_i\ is\ reduced) \end{aligned} \tag{36}$$

The minimal value $\Psi_\lambda(\mathbf{v}_\lambda^*[n+2])$ (32) of the *CPL* criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) in the vertex $\mathbf{v}_k^*[n+2]$ represents an equilibrium between the "force" of linear separability (16) and the "force" of features costs determined by the parameters $\lambda$ and $\gamma_i$. We can remark that an increase of the value of the parameter $\lambda$ in the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) causes an increase in the number of the unit vectors $\mathbf{e}_i[n+2]$ in the basis $\mathbf{B}_k^*[n+2]$ (34) linked to the optimal vertex $\mathbf{v}_k^*[n+2]$ (32). As a consequence, an increase of the *cost level* $\lambda$ value in the minimized function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) results in an increased number of the reduced features $X_i$ (36). Furthermore, the dimensionality of the feature $F[n]$ can be reduced arbitrarily in accordance with the rule (36) by a sufficient increase of the parameter $\lambda$ in the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30). Such method of feature selection is called *relaxed linear separability* (*RLS*) [4]. A successive increase of the *cost level* $\lambda$ in the minimized function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) allows to reduce the less important (redundant) features $X_i$ and to generate descending sequence of feature subspaces $F_k[n_k]$ ($F_k[n_k] \supset F_{k+1}[n_{k+1}]$, where $n_k > n_{k+1}$) [4]:

$$\begin{gathered} F[n] \to F_1[n_1] \to ... \to F_k[n_k], \\ where\ 0 \leq \lambda_0 < \lambda_1 < ... < \lambda_k \end{gathered} \tag{37}$$

Each feature subspace $F_k[n_k]$ in the above sequence has been linked to a certain value $\lambda_k$ of the cost level $\lambda$ in the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30). The sequence (37) of the feature subspaces $F_k[n_k]$ is generated in a deterministic manner based on the positive set $\mathbf{Z}^+[n+2]$ and the negative set $\mathbf{Z}^-[n+2]$ (15) in accordance with the *relaxed linear separability* (*RLS*) method [4]. Each step $F_k[n_k] \to F_{k+1}[n_{k+1}]$ has been realized by a minimal increase $\lambda_k \to \lambda_{k+1} = \lambda_k + \Delta_k$ (where $\Delta_k > 0$) of the cost level $\lambda$ in the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30).

A high value $\lambda_k$ of the cost level $\lambda$ in criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) can cause replacement of all vectors $\mathbf{z}_j^+[n+2]$ (13) or $\mathbf{z}_j^-[n+2]$ (14) in the basis $\mathbf{B}_k^*[n+2]$ (34) linked to the optimal vertex $\mathbf{v}_k^*[n+2]$ (32) by unit vectors $\mathbf{e}_i[n+2]$ and the solution $\mathbf{v}_k^*[n] = 0$ could appear. Such solution is not a constructive one, because it means that all features $X_i$ have been eliminated (33). A compromise solution is needed, which would allow to preserve the most important feature subset. Such postulate can be realized through an adequate stop criterion in the process of the feature space $F[n]$ reduction (37). The stop criterion could be based on evaluating the quality of particular feature subspaces $F_k[n_k]$ in the sequence (37).

In accordance with the *relaxed linear separability* (*RLS*) approach to feature subset selection, the quality of particular subspaces $F_k[n_k]$ (37) is evaluated on the basis of the optimal linear classifier designed in this subspace [4]. A better optimal linear classifier means a better feature subspace $F_k[n_k]$. In the context of this paper, a better feature subspace $F_k[n_k]$ (37) should guarantee a better prognostic model (27).

### 2.5 Evaluation of the *CPL* regression models based on non censored data

Let us consider the learning set $C_I$ (3) composed of the disjoined subset of not censored data $C_0$ (2) and the censored subsets $C_R$ (4), and $C_L$ (5):

$$C_I = C_0 \cup C_R \cup C_L \tag{38}$$

The *right censored* subset $C_R$ (4) as well as the *left censored* subset $C_L$ (5) could be empty. The nonempty subset $C_0$ (2) of feature vector $\mathbf{x}_j[n]$ with not censored values $t_j$ can be transformed into the subset $C_I'$ (6) with the intervals $[t_j - \varepsilon, t_j + \varepsilon]$ in order to define the augmented vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14). Each element $\mathbf{x}_j[n]$ of the not censored data set $C_0$ (2) generates two augmented vectors $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$:

$$\begin{aligned} &(\forall \mathbf{x}_j[n] \in C_0) \\ &\mathbf{z}_j^+[n+2] = [\mathbf{x}_j[n]^T, 1, -t_j + \varepsilon]^T \\ &\mathbf{z}_j^-[n+2] = [\mathbf{x}_j[n]^T, 1, -t_j - \varepsilon]^T \end{aligned} \tag{39}$$

Each element $\mathbf{x}_j[n]$ of the censored subsets $C_R$ (4) or $C_L$ (5) generates only one augmented vector $\mathbf{z}_j^+[n+2]$ (13) or $\mathbf{z}_j^-[n+2]$ (14):

$$(\forall \mathbf{x}_j[n] \in C_R)\ \mathbf{z}_j^+[n+2] = [\mathbf{x}_j[n]^T, 1, -tj - \varepsilon]^T \tag{40}$$

and

$$(\forall \mathbf{x}_j[n] \in C_L)\ \mathbf{z}_j^-[n+2] = [\mathbf{x}_j[n]^T, 1, -tj + \varepsilon]^T \tag{41}$$

The convex and piecewise-linear (*CPL*) criterion functions $\Phi(\mathbf{v}[n+2])$ (22) and $\Psi_\lambda(\mathbf{v}[n+2])$ (30) were defined on the elements $\mathbf{z}_j^+[n+2]$ (39) or (40) of the set $\mathbf{Z}^+[n+2]$ (15) and on the elements $\mathbf{z}_j^-[n+2]$ (39) or (41) of the set $\mathbf{Z}^-[n+2]$ (15). The parameters $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$ constituting the minimal value (24) of the criterion functions $\Phi(\mathbf{v}[n+2])$ (22) were used for defining the prognostic model $t^*(\mathbf{x}[n])$ (27). The *CPL* prognostic model (27) allows to compute the predicted values $t_j^*$ for the feature vectors $\mathbf{x}_j[n]$ belonging to the not censored data set $C_0$ (2):

$$(\forall \mathbf{x}_j[n] \in C_0) \; t_j^* = \mathbf{a}[n]^T \mathbf{x}_j[n] + b \tag{42}$$

where

$$\mathbf{a}[n] = \mathbf{w}^*[n]/\beta^* \;\; and \;\; b = w_0^*/\beta^* \tag{43}$$

We can compare the predicted values $t_j^*$ with the observed values $t_j$ from the not censored data set $C_0$ (2). In the case of the *classical regression*, the quality of the prognostic model (42) is evaluated on the basis of the sum of the squared differences $(t_j - t_j^*)^2$ between the observed variable $t_j$ and the modelled $t_j^*$ value (42).

We are using the absolute differences $|t_j - t_j^*|$ instead of the squared differences $(t_j - t_j^*)^2$ in the prognostic model (42) evaluation because the absolute differences $|t_j - t_j^*|$, in the same way as the criterion functions $\Phi(\mathbf{v}[n+2])$ (22) and $\Psi_\lambda(\mathbf{v}[n+2])$ (30) can be linked to the $L_1$ norm [2]. On the other hand, the squared differences $(t_j - t_j^*)^2$ are linked to the $L_2$ (*Euclidean*) norm [11].

The *discrepancy* coefficient $Q_a$ of the prognostic model (42) is determined as the mean value of the absolute differences $|t_j - t_j^*|$:

$$Q_a = \sum_j |t_j - t_j^*|/m_0 \tag{44}$$

where the summation is over the all $m_0$ elements of the not censored data set $C_0$ (2).

The *CPL* prognostic model (42) appears as a result of attempted linear separation (16) of the positive set $\mathbf{Z}^+[n+2]$ from the negative set $\mathbf{Z}^-[n+2]$ (15). This linear separation (16) also allows to define the below linear classifier of the augmented vectors $\mathbf{z}_j[n+2]$ (39), (40), (41) [3]:

$$\begin{array}{l} if \; \mathbf{v}^*[n+2]^T \mathbf{z}_j[n+2] \geq 0, \\ \;\; then \; \mathbf{z}_j[n+2] \; is \; located \; in \; the \; set \; \mathbf{Z}^+[n+2] \\ if \; \mathbf{v}^*[n+2]^T \mathbf{z}_j[n+2] < 0, \\ \;\; then \; \mathbf{z}_j[n+2] \; is \; located \; in \; the \; set \; \mathbf{Z}^-[n+2] \end{array} \tag{45}$$

The quality of the linear classifiers (45) can be evaluated in the usual way by using the error estimator (*apparent error rate*) $e_a(\mathbf{v}^*[n+2])$ as the fraction of wrongly classified elements $\mathbf{z}_j[n+2]$ (39), (40), (41) of the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15):

$$e_a(\mathbf{v}^*[n+2]) = m_a(\mathbf{v}^*[n+2])/m \tag{46}$$

where $m$ is the number of all elements $\mathbf{z}_j^+[n+2]$ (39), (40) of the set $\mathbf{Z}^+[n+2]$ (15) and the elements $\mathbf{z}_j^-[n+2]$ (39), (41) of the set $\mathbf{Z}^-[n+2]$ (15) and $m_a(\mathbf{v}^*[n+2])$ is the number of these elements which are wrongly allocated by the rule (45).

The optimal parameters $\mathbf{v}^*[n+2]$ in the classification rule (45) are obtained through the minimization of the criterion function $\Phi(\mathbf{v}[n+2])$ (22) which is defined on $m$ elements $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ of the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15), adequately.

It is known that if the same vectors $\mathbf{z}_j[n+2]$ are used for classifier (45) designing and classifier evaluation (46), then the evaluation results are too optimistic (*biased*). In order to reduce the bias of the apparent error rate estimator $e_a(\mathbf{v}^*[n+2])$ (46) is usually replaced by the cross-validation error rate $e_{CVE}(\mathbf{v}^*[n+2])$ [2].

It was assumed in the earlier approach to the *CPL* prognostic model (27) evaluation that a "good" prognostic model should have the lowest error rate $e_{CVE}(\mathbf{v}^*[n+2])$ estimated through the cross-validation procedure [2]. The *discrepancy* coefficient $Q_a$ (44) can also be used in the *CPL* prognostic model (42) evaluation. The *discrepancy* coefficient $Q_a$ (44), similarly to the error rate $e_a(\mathbf{v}^*[n+2])$ can be a biased evaluation of the quality of the prognostic model $t^*(\mathbf{x}[n])$ (27). The cross-validation (*leave-one-out*) procedure can be used for the bias reduction of the *discrepancy* coefficient $Q_a$ (44). The *leave-one-out* evaluation $Q_{CVE}$ of the discrepancy coefficient can be defined for this purpose:

$$Q_{CVE} = \sum_i Q'(i)/m_0 \tag{47}$$

where the above summation is over all of the $m_0$ temporarily removed elements of the non-censored data set $C_0$ (2) and

$$Q'(i) = \sum_{j \in J(i)} |t_j - t_j^*(i)|/(m_0 - 1) \tag{48}$$

where the symbol $J(i)$ stands for the set of indices $j$ of all $m_0 - 1$ elements $\mathbf{x}_j[n]$ of the set $C_0$ (2) apart from the $i$-th element $\mathbf{x}_i[n]$, which is temporarily removed.

The parameters $\mathbf{v}_i^*[n+2]$ of the prognostic model $t_j^*(i)$ (42) in the above expression (48) were determined through the minimization of the criterion function $\Phi_i(\mathbf{v}[n+2])$ (22) which was defined on all the $m_0 - 1$ elements of the set $C_0$ (2), except for the $i$-th element $\mathbf{x}_i[n]$.

In line with the modification proposed in this paper of the relaxed linear separability (*RLS*) method, the quality of particular subspaces $F_k[n_k]$ in the descending sequence (37) is evaluated based on the cross-validation values $Q_{CVE}$ (47) of the discrepancy coefficient $Q_a$ (44). It has been assumed here that a better feature subspace

$F_k[n_k]$ (37) allows to design *CPL* prognostic models $t^*(\mathbf{x}[n_k])$ (42) characterized by a lower cross-validation value $Q_{CVE}$ (47) of the discrepancy coefficient.

The minimal value of the *discrepancy* coefficient $Q_{CVE}$ (47) is used in this paper as the stop criterion for the process of feature space $F[n]$ reduction described by the descending sequence (37).

## 3. Experimental results and discussions

### 3.1 A toy model identification

The toy data set used in the experiment was generated by the authors. Seven points $x_j$ $(j = 1, ..., 7)$ were arbitrarily selected on the line $(x_j \in R^1)$. The values $t_j$ of dependent variable $Y$ were generated for each point $x_j$ in accordance with the below model (1):

$$(\forall j \in \{1, ..., 7\})\; t_j = 1 - x_j + \zeta_j \tag{49}$$

where the numbers $\zeta_j$ $(\zeta_j \in R^1)$ were generated in accordance with the normal probability distribution $(\zeta_j \sim N(0, \sigma))$ with the expected value zero and the variance $\sigma^2$ equal to three different values (0.3, 0.5, 0.7). The generated data sets are given in Table 1.

**Table 1.** Three toy data sets.

| | | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|
| $x_j$ | $t_j = 1 - x_j$ | $t_j = 1 - x_j + \zeta_j$ | $t_j = 1 - x_j + \zeta_j$ | $t_j = 1 - x_j + \zeta_j$ |
| | | $(\sigma^2 = 0.3)$ | $(\sigma^2 = 0.5)$ | $(\sigma^2 = 0.7)$ |
| -5 | 6 | 6.113 | 6.370 | 5.671 |
| -4 | 5 | 5.237 | 3.627 | 5.018 |
| -2 | 3 | 1.605 | 4.209 | 1.995 |
| 0 | 1 | 1.036 | 0.335 | 1.062 |
| 1 | 0 | -0.456 | -0.459 | -0.419 |
| 3 | -2 | -2.544 | -2.319 | -0.514 |
| 5 | -4 | -4.302 | -3.232 | -3.741 |

In this case, the classical learning set $C_0$ (2) and the interval learning set $C_I$ (3) have the below form:

$$C_1' = \{x_j, t_j, \textit{ where } j = 1, ..., 7\} \tag{50}$$

$$C_2' = \{x_j, [t_j - \varepsilon, t_j + \varepsilon], \textit{ where } j = 1, ..., 7\} \tag{51}$$

where $\varepsilon = 0.5$. The value of the parameter $\varepsilon$ specifying the length of the interval $[t_j - \varepsilon, t_j + \varepsilon]$ was set to 0.5 in all experiments described in this subsection.

The prognostic (regression) model designed on the basis of the learning sets $C'_1$ (50) or $C'_2$ (51) has the following form, depending on parameters $w_1$ ($w_1 \in R^1$) and $w_0$ ($w_0 \in R^1$):

$$t(x) = w_1 x + w_0 \tag{52}$$

The parameters $w_1$ and $w_0$ were estimated based on the learning set $C'_1$ (50) by using the classical method of least squares [13]. The parameters $w_1$ and $w_0$ of the model (52) were also estimated based on the interval learning set $C'_2$ (51) through minimization (24) of the *CPL* criterion function $\Phi(\mathbf{v}[n+2])$ (22). The results of these experiments are shown in the Table 2 and the Figure 1.

**Table 2.** Parameters $w_1$ and $w_0$ of the model (52) estimated from the toy data sets.

| | classical regression | | interval regression | |
|---|---|---|---|---|
| | $w_1$ | $w_0$ | $w_1$ | $w_0$ |
| Data 1 | -1.038 | 0.659 | -1.060 | 0.801 |
| Data 2 | -0.956 | 0.946 | -0.960 | 1.035 |
| Data 3 | -0.887 | 1.043 | -0.941 | 0.965 |

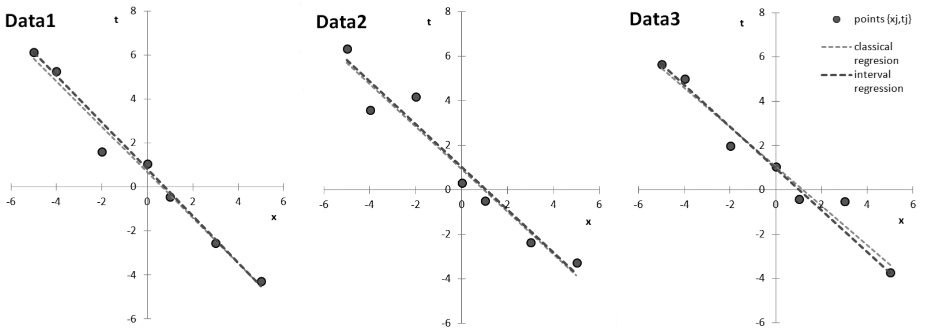


**Fig. 1.** The model $y = 1 - x$ identification by a classical regression and the model $t = 1 - x$ identification by an interval regression.

From Figure 1, it can be seen that the interval regression allowed to estimate the parameters $w_1$ and $w_0$ which are similar to the model of classical regression from this toy dataset. We can also note that he lowest identification quality of the model $t = 1 - x$ (45) has been obtained in the case of *Data 3* set which is characterized by the highest level of noise.

The *right censored* set $C'_R$ (4) and the *left censored* set $C'_L$ (5) have been also generated randomly from the toy data sets collected in the Table 1. The *indicator of censoring* $\delta_j$ ($\delta_j = 1$ or $\delta_j = 0$) has been used for this purpose. The value $\delta_j = 1$ implied that the interval $[t_j - \varepsilon, t_j + \varepsilon]$ (51) was censored to the form (4) or (5). In other words, if the value $\delta_j = 1$ appeared, the interval $[t_j - \varepsilon, t_j + \varepsilon]$ was replaced with equal probability $p = 0.5$ by $[-\infty, t_j + \varepsilon]$ or by $[t_j - \varepsilon, +\infty]$. The value $\delta_j = 0$ implied that the interval $[t_j - \varepsilon, t_j + \varepsilon]$ (51) was not changed. The censoring process was controlled by the parameter $p_c$ called *probability of censoring* ($0 \leq p_c \leq 1$). The censoring ($\delta_j = 1$) was drawn for each interval $[t_j - \varepsilon, t_j + \varepsilon]$ (51) with the probability $p = p_c$. The *CPL* prognostic models (27) obtained for several values of the parameter $p_c$ (*probability of censoring*) were sketched in Figure 2.

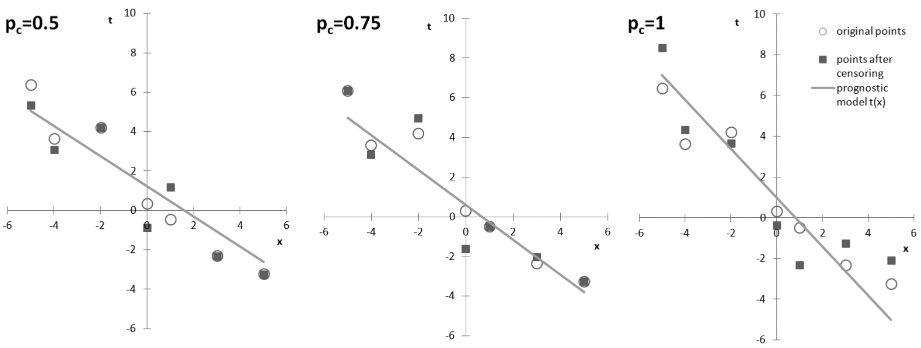


**Fig. 2.** The *CPL* prognostic models $t(x)$ (52) estimated from the toy data sets (Table 1) for several values of the censoring probability $p_c$.

We can remark that even learning sets with the full censoring ($p_c = 1$) allow to obtain a reasonably good prognostic model $t(x)$ (52).

### 3.2 Prognostic model selection on synthetic data

The synthetic data set contained $m = 100$ objects (feature vectors) $\mathbf{x}_j[n]$ ($j = 1, ..., 100$). Each object $\mathbf{x}_j[n]$ was represented by $n = 100$ features $X_i$ ($i = 1, ..., 100$). The value $x_{ji}$ of each feature $X_i$ of particular object $\mathbf{x}_j[n]$ were drawn from a uniform distribution on the unit interval $[0, 1]$ ($X_i \in [0, 1]$). The value of the dependent variable $t_j$ was computed as the bellow linear combination (*linear key*) with coefficients $\alpha_{ji}$

of the selected components $x_{ji}$ of the feature vector $\mathbf{x}_j[n]$:

$$\begin{aligned}&(\forall j \in \{1,...,m\})\\ &t_j = 3x_{j5} + 4x_{j10} + 7x_{j16} + 2x_{j37} + 6x_{j45} +\\ &\quad +3x_{j50} + 3x_{j67} + 8x_{j72} + x_{j84} + x_{j91} + 10\end{aligned} \tag{53}$$

The set of 10 features $X_i$ and their coefficients $\alpha_{ji}$ in the above linear key were defined arbitrarily before the experiment. The linear key (53) was used for generating the classical regression learning set $C_0 = \{(\mathbf{x}_j[n]; y_j)\}$ (2).

Some of the dependent values $y_j$ in this set $C_0$ were censored. The below scheme of censoring was adopted for the synthetic data set $C_0$. The censoring process was controlled, as in the case of the toy data set, by the parameter $p_c$ $(0 \leq p_c \leq 1)$. The censoring $(\delta_j = 1)$ was drawn for each element $\mathbf{x}_j[n]$ of the learning set $C_0$ (2) with the probability $p_c$. As a result, $m_c$ elements $\mathbf{x}_j[n]$ were selected to be censored. If the value $\delta_j = 1$ was drawn, the dependent value $t_j$ was replaced by the censored value $t_j^c$ $(0 \leq t_j^c \leq t_j)$. The *right censored* value $t_j^c$ was randomly generated in accordance with the triangle probability distribution determined on the interval $[0, t_j]$.

To allow the use of the *CPL* functions $\Phi(\mathbf{v}[n+2])$ (22) and $\Psi_\lambda(\mathbf{v}[n+2])$ (30) the non-censored elements $(\mathbf{x}_j[n]; t_j)$ of the classical learning set $C_0$ (2) were transformed in the interval learning set $C_2'$ (51):

$$C_2' = \{\mathbf{x}_j[n], [t_j - \varepsilon, t_j + \varepsilon]\} \tag{54}$$

where $\varepsilon = 0.1$.

The *CPL* prognostic model (27) was defined in the experiment with the synthetic data set by the parameters $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$ (22) constituting the minimal value (24) of the criterion functions $\Phi(\mathbf{v}[n+2])$ (22), where $n = 100$. The *leave-one-out* evaluation of the *discrepancy* coefficients $Q_a$ (44) and $e_a(\mathbf{v}^*[n+2])$ (46) were used in the experiment for the purpose of the bias reduction. In accordance with the leave-one-out procedure, the criterion functions $\Phi(\mathbf{v}[n+2])$ (22) was defined for each time by using $m - m_c - 1$ non censored elements $\mathbf{x}_j[n]$, because one non-censored element $\mathbf{x}_j[n]$ was used only for evaluating the resulting model (42). As a result, the criterion functions $\Phi(\mathbf{v}[n+2])$ (22) were defined on $2(m - m_c - 1) + m_c$ augmented vectors $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ (39), (40), (41).

The results of the experiments on the synthetic data set are shown in Figure 3. Two types of curves (the *learning curve* and the *testing curve*) are used to evaluate the *CPL* prognostic model (27) in different feature subspaces $F_k[n_k]$ (37), generated in accordance with the *RLS* method [4]. The same prognostic model (27) has been evaluated in a two manners (two curves) by using the same *discrepancy* coefficient

$Q_a$ (44). The *learning curve* shows the averaged values of the coefficient $Q_a$ (44) computed on all $m - m_c$ non censored elements $(\mathbf{x}_j[n];t_j)$ of the data set $C_0$ (2). The *testing curve* shows the averaged values of the coefficient $Q_{CVE}$ (47) computed on $m$ temporarily removed elements $(\mathbf{x}_{j'}[n];t_{j'})$ of the data set $C_0$ (2). In the case of the *learning curve*, the averaging is taking place from $m$ calculations of the coefficient $Q_a$ (44). In the case of the *testing curve*, the discrepancy coefficient $Q_{CVE}$ (47) is computed in different feature subspaces $F_k[n_k]$ (37).

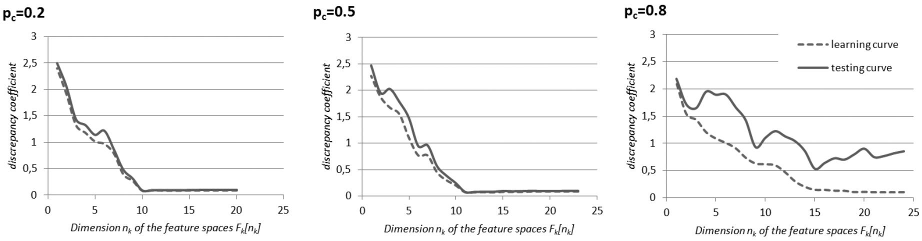


**Fig. 3.** The discrepancy evaluation $Q_a$ (44) (learning curve) and $Q_{CVE}$ (47) (testing curve) of the model (27) in different feature subspaces $F_k[n_k]$ (37) on the base of synthetic data with a few probabilities of censoring $p_c$.

We can note that the minimal value of the discrepancy coefficient $Q_{CVE}$ (47) on the Figure 3 is located in the feature subspace $F_k[n_k]$ (37) of dimensionality $n_k$ approximately equal to 10 ($n_k \approx 10$), as it was assumed in the linear key (53). Both the features $X_i$ and their coefficients $\alpha_{ji}$ constituting the linear key (53) were approximately reproduced as a result of the *CPL* prognostic model designing. The linear key (53) was most accurately reproduced from the synthetic data set with the lowest probability of censoring $p_c = 0.2$.

The results of these experiments on the synthetic data set show the usefulness of a criterion based on the discrepancy coefficient $Q_{CVE}$ (47) fore discovering the linear key (decision rule) separating two linear sets in a reasonably good manner.

### 3.3 Prognostic model selection based on the *Adrenocortical carcinoma* data set

The *Adrenocortical carcinoma* [7] data set consists of patient samples suffering from this type of cancer. The set contains 79 objects, each described by 20533 features (age, gender and gene expression values). Each object has a specified time value measured from start of observation until death (on average 915 days) or censoring (on average 1765 days). 51 patients (64.5%) were still alive at the final follow-up visit (censoring observations).

On the basis of 79 objects from the *Adrenocortical carcinoma* data set, 107 elements $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14) were created. The *RLS* method was applied to the newly formed data sets (15).

The main objective of this experiment was to examine the possibility of the *CPL* prognostic models $t^*(\mathbf{x}[n_k])$ (27) evaluation in different feature subspaces $F_k[n_k]$ (37) by using the discrepancy coefficient $Q_{CVE}$ (47). More specifically, the possibility of using the minimal value of the discrepancy coefficient $Q_{CVE}$ (47) as the stop criterion for the descending sequence of subspaces $F_k[n_k]$ (37) was examined. The optimal feature subspace $F_k^*[n_k]$ defined by this stop criterion was the last stage of the feature reduction procedure (35).

We can observe in Figure 4 that the minimal value of the discrepancy coefficient $Q_{CVE}$ (47) has been reached in the feature subspace $F_k^*[n_k]$ (37) with dimensionality $n_k$ of about 26 ($n_k \approx 26$).

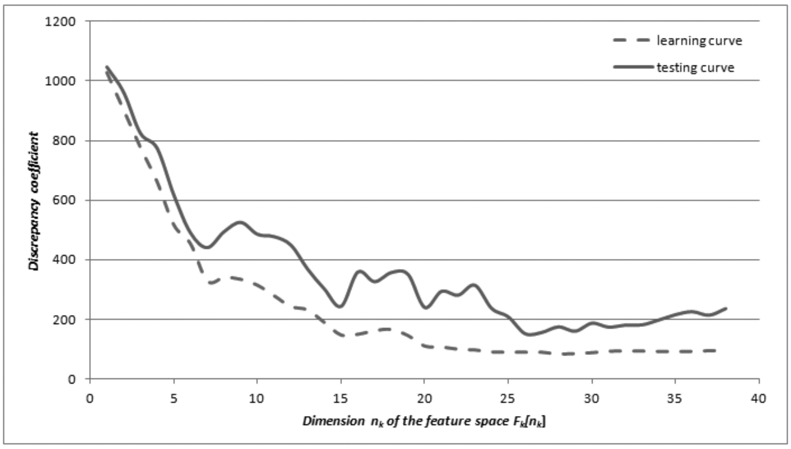


**Fig. 4.** The discrepancy evaluations $Q_a$ (44) (learning curve) and $Q_{CVE}$ (47) (testing curve) of the model (27) in different feature subspaces $F_k[n_k]$ (37) of the *Adrenocortical carcinoma* data set.

The prognostic model $t^*(\mathbf{x}[n_k])$ (27) was also evaluated, by using the cross validation error $e_{CVE}(\mathbf{v}^*[n+2])$ [2] of the linear classifier (45) of the augmented feature vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14). The optimal feature subspace $F_k^*[n_k]$ indicated in the sequence (37) by the minimal value of the discrepancy coefficient $Q_{CVE}$ (47) is similar to the optimal feature subspace identified in cross validation error

$e_{CVE}(\mathbf{v}^*[n+2])$ (see Figure 5). This result might have some practical meaning, because it is an additional confirmation that our methodology is correct. The next time, the optimal feature subspace $F_k^*[n_k]$ was identified through an attempted linear separation of the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15) of the augmented feature vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14). The augmented feature vectors $\mathbf{z}_j^+[n+2]$ (13) and $\mathbf{z}_j^-[n+2]$ (14) can represent both the non-censored (39), as well as censored cases (40), (41). As a result, the criterion based on the attempted linear separation of the sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15) is less clear than the criterion based on the minimal discrepancy $Q_{CVE}$ (47) intuitively assessed only on the non-censored cases (37).

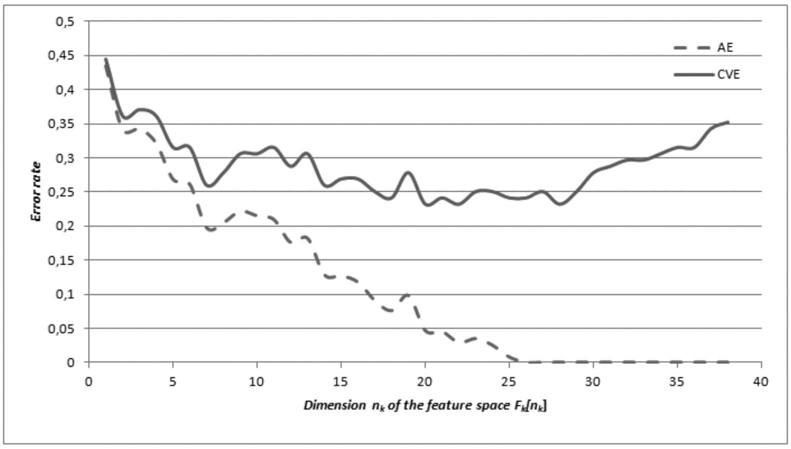


**Fig. 5.** The classifier error evaluations $e_a$ (46) (AE) and $e_{CVE}$ (CVE) of the model (27) in different feature subspaces $F_k[n_k]$ (37) of the *Adrenocortical carcinoma* data set.

## 4. Conclusions

Designing prognostic models (27) through exploring linear separability has been examined in the paper, based on examples of genetic data set with censored survival times. The task of the linear regression model designing has been reformulated as the problem of the linear separability of the augmented sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15). The augmented sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15) allow to represent both the non-censored, as well as the censored learning sets in the regression analysis.

The exploration of the linear separability of the augmented sets $\mathbf{Z}^+[n+2]$ and $\mathbf{Z}^-[n+2]$ (15) has been performed through minimization (24) of the perceptron (*CPL*) criterion function $\Phi(\mathbf{v}[n+2])$ (22). The parameters $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$ constituting the minimum (24) of the criterion function $\Phi(\mathbf{v}[n+2])$ (22) have been used in the definition of the optimal prognostic model $t^*(\mathbf{x}[n])$ (27).

The modified *CPL* criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) has been used for generating a sequence of feature subspaces $F_k[n_k]$ (37) in accordance with the relaxed linear separability (*RLS*) method of feature subset selection (model selection).

The prognostic models $t^*(\mathbf{x}[n])$ (27) designed in different feature subspaces $F_k[n_k]$ (37) have been validated through the discrepancy coefficient $Q_{CVE}$ (47) computed in accordance with the cross-validation (*leave one out*) procedure.

The proposed method of the prognostic models designing has been tested both on synthetic data set with the hidden linear key (49), as well as on real genetic data set *Adrenocortical carcinoma* [7] with censored survival times.

The experiments carried out on the genetic data set *Adrenocortical carcinoma* demonstrated, that the *RLS* method allows to find subsets of few genes $X_i$ with good prognostic properties, even if the number of genes $X_i$ is large at the beginning. The selection of optimal subsets of genes $X_i$ was based on the minimal value of the discrepancy coefficient $Q_{CVE}$ (47) computed in accordance with the leave-one-out procedure. The modified *RLS* method based on the discrepancy coefficient $Q_{CVE}$ (47) has been proposed and applied for the purpose of the *CPL* prognostic models selection.

The linear key based on 10 variables $X_i$ (53) was hidden in the synthetic data set composed of $n = 100$ variables (features) $X_i$. The *RLS* method allowed to find the model (53) hidden in the learning set containing $m = 100$ feature vectors $\mathbf{x}_j[n]$. The model $t(\mathbf{x}[n])$ (42) was approximately identified even when all values $t_j$ of the dependent variable $T$ were censored.

The linear prognostic model (27) have been designed in the reduced feature subspaces $F_k[n_k]$ (37) in a deterministic manner, even though the dimensionality of the genetic data sets was high and the survival times censored.

One of the promising results of the experiments is the possibility to use the discrepancy measure $Q_{CVE}$ (47) in the modified *RLS* method of the *CPL* prognostic models selection. The stop criterion for the sequence (37) of reduced feature subspaces $F_k[n_k]$ can be based on the minimal value of discrepancy coefficient $Q_{CVE}$ (47) (see Figure 4).

## 5. Acknowledgment

This work was partially supported by the S/WI/2/2018 grant from the Bialystok University of Technology and by the 4.2/St/2018 grant from the Institute of Biocybernetics and Biomedical Engineering PAS.

The results shown here are in part based upon data generated by the TCGA Research Network: http://cancergenome.nih.gov/.

## References

[1] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[2] Leon Bobrowski. *Data mining based on convex and piecewise linear (CPL) criterion functions (in Polish)*. Bialystok University of Technology Press, 2005.

[3] Leon Bobrowski. Prognostic models based on linear separability. *Advances in Data Mining. Applications and Theoretical Aspects*, pages 11–24, 2011.

[4] Leon Bobrowski and Tomasz Łukaszuk. Relaxed linear separability (RLS) approach to feature (gene) subset selection. In *Selected works in bioinformatics*. InTech, 2011.

[5] Leon Bobrowski and Tomasz Łukaszuk. Prognostic modeling with high dimensional and censored data. In *Industrial Conference on Data Mining*, pages 178–193. Springer, 2012.

[6] Leon Bobrowski and Wojciech Niemiro. A method of synthesis of linear discriminant function in the case of nonseparability. *Pattern Recognition*, 17(2):205–210, 1984.

[7] Broad Institute TCGA Genome Data Analysis Center. Analysis overview for adrenocortical carcinoma (primary solid tumor cohort) - 28 january 2016, 2016.

[8] Jonathan Buckley and Ian James. Linear regression with censored data. *Biometrika*, 66(3):429–436, 1979.

[9] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[10] Guadalupe Gómez, Anna Espinal, and Stephen W Lagakos. Inference for a linear regression model with an interval-censored covariate. *Statistics in medicine*, 22(3):409–425, 2003.

[11] Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 4. Prentice-Hall New Jersey, 2014.

[12] John P Klein and Melvin L Moeschberger. Survival analysis: techniques for censored and truncated data. 1997.

[13] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.

[14] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

*Leon Bobrowski, Tomasz Łukaszuk*

# SELEKCJA CECH NA POTRZEBY MODELI PROGNOSTYCZNYCH POPRZEZ LINIOWĄ SEPARACJĘ ZBIORÓW DANYCH GENETYCZNYCH DOTYCZĄCYCH ANALIZY PRZEŻYCIA

**Streszczenie** W artykule rozważane jest projektowanie modeli regresji opartych na wysokowymiarowych (np. genetycznych) zbiorach danych poprzez badanie problemu separacji liniowej. Projektowanie modelu regresji liniowej zostało tu przeformułowane jako problem separacji liniowej. Eksploracja problemu separacji liniowej opiera się na minimalizacji wypukłej i odcinkowo-liniowej (CPL) funkcji kryterialnej. Minimalizacja funkcji kryterialnej typu CPL została wykorzystana nie tylko do oszacowania parametrów modelu prognostycznego, ale również do skutecznego wyboru podzbioru cech (selekcji modelu) zgodnie z metodą relaksacji separacji liniowej (RLS). Takie podejście do projektowania modeli prognostycznych zostało wykorzystane w eksperymentach zarówno z syntetycznymi danymi wielowymiarowymi, jak i do zbiorów danych genetycznych zawierających cenzurowane wartości zmiennej zależnej. Jakość modeli prognostycznych otrzymywanych w oparciu o postulat liniowej separacji została oceniona przy użyciu miary rozbieżności modelu i szacowanego wskaźnika błędu klasyfikacji. W celu zmniejszenia obciążenia oceny, obliczono wartości rozbieżności modelu i błędu klasyfikacji w różnych podprzestrzeniach cech, zgodnie z procedurą walidacji krzyżowej. Seria nowych eksperymentów opisanych w niniejszym opracowaniu pokazuje, że projektowanie modeli regresji może być oparte na zasadzie separacji liniowej. W szczególności, w procedurze projektowania można użyć wysokowymiarowych zbiorów genetycznych o cenzurowanej zmiennej zależnej. Proponowana miara rozbieżności modelu prognostycznego może być skutecznie wykorzystana w poszukiwaniu optymalnej podprzestrzeni cech i selekcji modelu regresji liniowej.

**Słowa kluczowe:** eksploracja danych, regresja interwałowa, selekcja modelu, relaksacja separacji liniowej

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

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

Adrian Adam Falkowski, Anna Łupińska–Dubicka

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

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

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

## 1. Introduction

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

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

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

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

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

## 2. Bayesian Networks

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

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

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

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

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

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

## 3. Methods of handling missing data

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

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

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

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

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

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

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

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

## 4. Data sets and Models

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

**Car Evaluation data set** contains 1728 records and 7 attributes (the last attribute is a decision class): *buying* (price of a given car), *maint* (possible maintenance costs), *doors* (number of doors), *persons* (number of person who can travel in a given car), *lug_boot* (luggage capacity), and *safety* (level of a given car's safety) [3]. On the basis of attributes concerning each of the cars, it is possible to determine to which decision class the auto data can be assigned. The data set does not contain not many records, but it covers all cases. However, the distribution of decision class is very asymmetrical: 70% of records belong to class *unacc*, about 22% to class *acc*, and classes *good* and *vgood* contain about 4% of records each, while the distributions of each attribute in the data set are uniform. It is worth mentioning that Car Evaluation data set was derived from a simple hierarchical decision model originally developed for the demonstration of DEX [4]. Besides the six input variables, the model included three intermediate concepts: *price*, *tech*, *comfort*. However, the Car Evaluation data set in final form contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes.

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

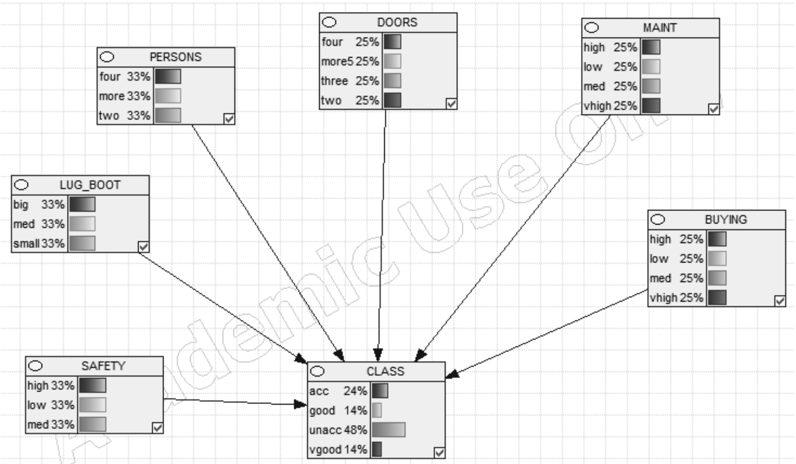


**Fig. 1.** A Bayesian network model of *Car Evaluation* data set.

**Nursery data set** contains 12,960 records consisting of 8 attributes plus the attribute of the decision class. The database presents information on the application of the child to kindergartens. The collection was created within a few years in the 1980s in Ljubljana, when many requests for admission were rejected. In the original research, the data set was used in the machine learning HINT evaluation (Hierarchy INduction Tool), which was able to completely recreate the original hierarchical model. The final decision depends on several factors of parental employment and financial situation, family structure or health status of the child. The attributes in data set are as follows: *parents* (parents' occupation), *has_nurs* (level of childcare), *form* (family structure), *children* (number of children in a family), *housing* (family's living conditions), *finance* (family's financial conditions), *social* (social condition of a family), *health* (child's health condition) [9]. Like the previous data set, records in the distribution of decision–making class is asymmetrical, while the distributions of all attributes are uniform.

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

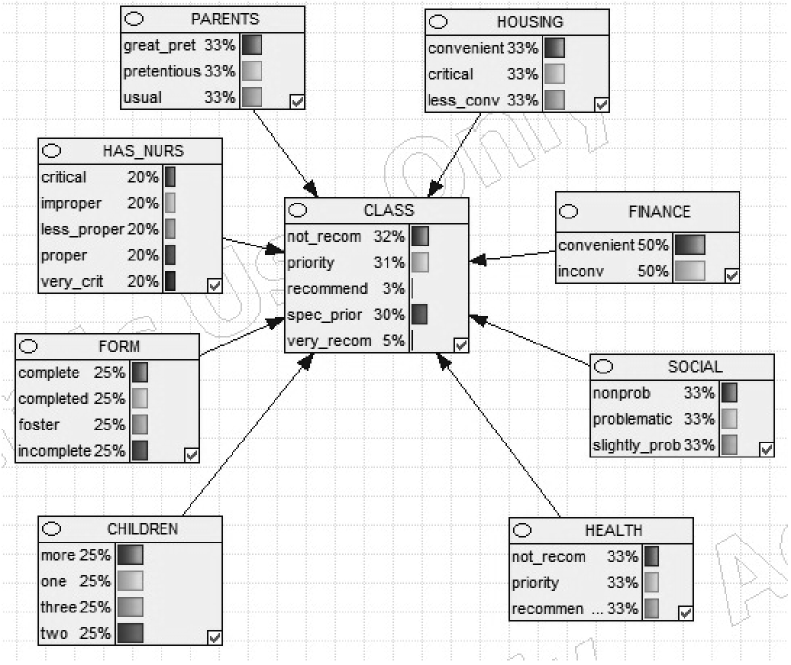


**Fig. 2.** A Bayesian network model of *Nursery* data set.

## 5. Experiments

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

### 5.1 Car Evaluation Experiment

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

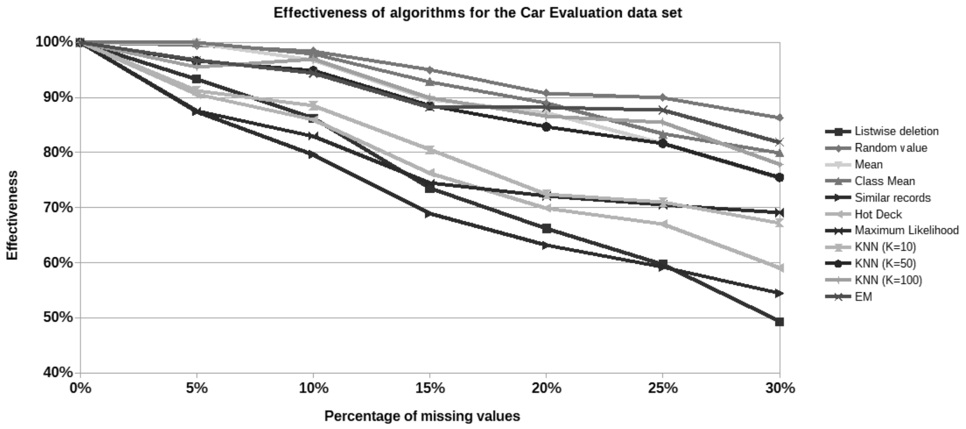


**Fig. 3.** Figure caption

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

**Table 1.** Results for Car Evaluation data set

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

ML means Maximum Likelihood

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

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

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

### 5.2 Nursery Experiment

In the case of the *Nursery* data set, the experiment was performed in a bit different way than in case the *Car Evaluation* data set. The *Car Evaluation* data set consisted of a small number of records, which is why the network learning methodology was used on the original data set and a training set consisting of 10,000 records was generated. In the case of the *Nursery* data set, there was no such need – the size of the data set equaled roughly 13,000 records. The further steps of the experiment were identical. The original data set was divided into a test set (size 1000 records) and a training set. Having a training set, again, missing values were generated at the level of 5%, 10%, 15%, 20%, 25% and 30% respectively. Each of the created data sets was saved separately. Next stages consisted in filling the missing values by using particular methods, learning created Bayesian network and verifying on the basis of the test set. Similarly to the first experiment, this procedure was repeated ten times, and the average was drawn from the results (see Figure 4 and Table 2).

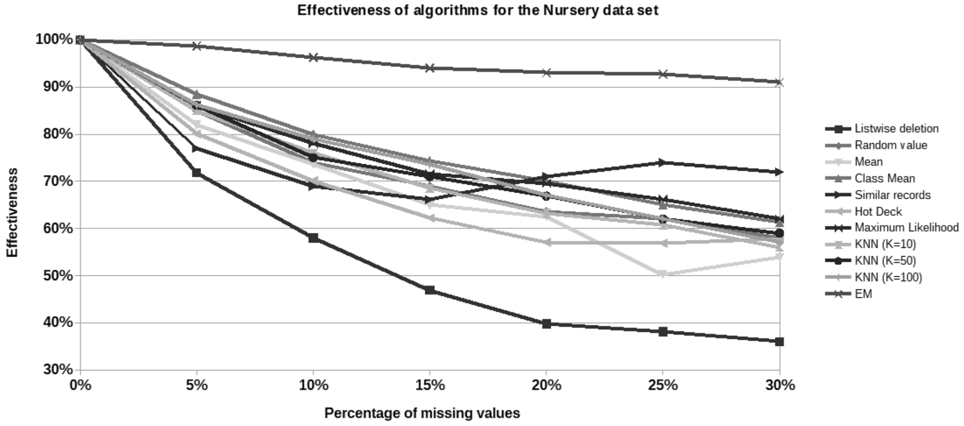


**Fig. 4.** Figure caption

**Table 2.** Results for Nursery data set

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

ML means Maximum Likelihood

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

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

## 6. Conclusions

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

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

## References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# PORÓWNANIE METOD UZUPEŁNIANIA DANYCH BRAKUJĄCYCH W UCZENIU MODELI PROBABILISTYCZNYCH

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

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

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

# GENERATING SYNTHETIC IRIS IMAGES FOR TESTING BIOMETRIC SYSTEMS

Łukasz Kasperuk, Marek Tabędzki

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

**Abstract:** Usage of biometric methods of person identification has recently become quite popular. Many developers are working on new biometric systems which have to be tested using biometric data. This work proposes the ways of generating synthetic biometric data designed for testing of iris based biometric systems. Two iris texture generation methods are presented here – one is based on Perlin noise, the other on the image quilting algorithm. Also a hybrid approach combining both methods is presented. The results of tests were also presented to demonstrate the utility of images generated using these methods in testing biometric systems and the consistency of the results obtained with the results for real images.

**Keywords:** iris, biometrics

## 1. Introduction

In recent years interest in biometric authentication systems has increased. This applies not only to professional applications, such as banking or security, but also access to a notebook or smartphone. One of the highly effective features is the image of the iris of the eye. It is characterized by high sample uniqueness as well as safety of use.

To assess the effectiveness of a biometric system, it is necessary to test it on a large data set. Probably the best test will be to use it in real conditions, on real data. However, the testing procedure can be significantly accelerated by using artificial data generated by the algorithms. In this way, we eliminate the tedious data acquisition procedure from the process. Alternatively, thanks to artificial data, we can multiply the size of our data set at a low cost. This can be especially important in the case of features whose acquisition is expensive or complicated.

This work presents a proposal to solve this problem. Two ways to generate artificial iris images for biometric purposes are presented here – the first one uses Perlin

Advances in Computer Science Research, vol. 14, pp. 69-86, 2018. DOI: 10.24427/acsr-2018-vol14-0005

noise [9], the second one combines fragments of natural images into synthetic samples. A hybrid method combining both solutions was also presented. The paper also presents the results of experimental tests to assess the suitability of the generated samples in the study of biometric systems.

## 2. Iris in biometric systems

The iris is a ring visible through the cornea that gives the eye color. Each iris has a dark back layer, the amount of pigment on the front and upper parts determines its color [10]. The main purpose of the iris is to adjust the pupil size to the current lighting. The iris has muscles that allow it to narrow and extend the pupil in response to the state of the body (eg fear, relaxation, sleep) or light intensity.

The iris pattern is usually rich in numerous bands and spots, which makes it a useful biometric feature. It has over 200 characteristic points, over four times more than the fingerprint [3]. It is recognized that details decorating the iris ring arise randomly during the development of the eye in the fetus. There are not yet two people with identical characteristics of the iris, even in the case of identical twins, the patterns of these details differ.

In early childhood, the amount of pigment and also the color of the iris may change. However, the later appearance of it remains in a relatively permanent form until death, after which it succumbs to self-destruction in a matter of seconds. It can therefore be concluded that the iris of the eye is a permanent biometric feature.

Systems based on the iris are also characterized by high acceptability. The measurement requires some cooperation from the tested person, however its duration is small and the process is relatively simple.

Additional procedures in biometric systems allow their effective protection against fraud attempts. For example, the technique that illuminates the examined eye can determine whether it responds correctly to changes in light intensity. In the case of false irises, the size of the pupil will not change. These systems are therefore characterized by high security.

### 2.1 Daugman's biometric system

John Daugman developed [3] the first fully functional and commercially used system that uses the iris of the eye for identification purposes, which has become the standard in its field. Daugman's patent described methods of sample acquisition, segmentation of the iris ring, normalization of data received, coding of a feature vector, and a method of comparing two processed biometric samples.

The first step is to take a biometric sample – a digital image of the eye from the examined person. Daugman suggested using the near-infrared spectrum to minimize the discomfort of the person being photographed. It is crucial to choose the appropriate wavelength and focus, allowing for accurate registration of the structure of the iris.

From the eye image received in the previous step, the part corresponding to the iris should be find. For this purpose, an approach known from face detection systems is used. The shape sought is described by a set of parameters defining its pattern. To describe a circle corresponding to the edge of the iris, three parameters are enough: $x$ and $y$ coordinates of its center, and its radius $r$. Daugman proposed a search of the space of values of these parameters using the integral-differential operator [4].

The outer edge of the iris is usually partially covered by the eyelids, the inner edge may be disturbed by light reflecting off the surface of the eye. In addition, the characteristics of both edges are affected by glasses. The method of active contours designed by Daugman [4] tolerates this type of disturbance.

The next step following the segmentation is to describe the characteristics of the iris in a way that allows comparison with other samples. Not all images have the same size – the distance of the eye from the camera affects the size of the iris in the picture, besides, lighting can cause the pupil to shrink or dilate. Daugman solved this problem by mapping the area of the iris obtained during segmentation to the normalized coordinate system – the iris ring is rolled into a rectangular ribbon, where the radius coordinates are on the vertical axis and the angular position on the horizontal axis.

Direct comparison of two images is largely susceptible to errors due to differences in lighting during their acquisition. To extract the clean structure of the iris, Daugman used a weave operation using two-dimensional Gabor filters. In the final vector of features, each pixel corresponds to two coded values depending on the real and imaginary value returned by the applied Gabor filter. Negative numbers are remembered as zero, positive as 1. In this way, the IrisCode is created – a 2048-bit code containing the characteristic features of the iris of the eye.

To compare two different iris codes, Daugman used the Hamming distance [6]. It allows measuring the part on which the compared strings differ.

## 2.2 Synthesis of iris data

Daugman's 1993 work is considered a breakthrough in the field of iris biometrics. Since its publication, interest in the techniques of classifying this biometric feature has increased. Methods of generating artificial data have also begun to appear.

In the work of Jiali Cui et al. from 2004 [2] a method was proposed to create a database of artificial eye images for testing biometric systems. It is based on the principal components analysis allowing the creation of iris images of the same class by appropriate selection of the feature vectors coefficients. In addition, it uses the super-resolution method to improve the quality of generated images. According to the calculations included in the work, this technician allows to create over 280 000 data classes under the described conditions. Conducted in 10 000 study classes using the Daugman system using the threshold obtained in the teaching process, they returned zero FAR and FRR error values.

In the work of Samir Shah and Arun Ross from 2006 [11], a technique for generating synthetic iris using the feature agglomeration was proposed. The image is created by synthesizing the iris background based on the Markov model and the real eye sample. Then, elements such as stripes or spots are added depending on the characteristics of said sample. In order to verify the results of the work, the authors carried out tests with the Daugman system comparing the differences in the codes within the three sets: filled with artificial and real images, and a set containing both types of images. Distributions of result data were similar for each case.

The paper of Jinyu Zuo et al. from 2007 [12] describes an approach that puts more emphasis on modeling the anatomy of the entire eye. At the beginning of the proposed method, continuous fibers are generated in a three-dimensional cylindrical space. Then, the layer imitating the eyeball is applied to the fiber projected onto the two-dimensional space. Finally, eyelids and randomly generated lashes are synthesized. In order to better simulate real samples, this technique takes into account effects that distort the generated images – noise, revolutions, light reflections and blur. The generated images were analyzed at the visual level and taking into account the vectors of iris features. They were also verified using the Daugman algorithm implementation to compare the real result distributions and the generated iris.

Despite the passage of about ten years from the first publication on the subject, the problem of generating synthetic eye images is still valid, as can be found, for example, in the work of Cardoso et al. from 2013 [1]. In addition to the precise generation of single layers of iris fibers, the developed synthesis method deeply addresses many aspects that allow realistic rendering of the human eye. In their work, the authors have taken into account the pupil distortion model, the sclera and the blood vessels placed in it, the light reflection properties of both the eye and the eyelids, and the ability to reproduce the reflected image of the surroundings on the surface of the eye. Intra-class diversity has been tested taking into account, inter alia, the sharpness of the image, the degree of closed eyelids, the distance and position of the head relative to the camera, lighting, pupil size or distortions caused by the glasses.

Qualitative tests comparing the generated base to existing real eye image databases were successfully carried out using the classic Daugman method based on images obtained using infrared light and a method using the visible spectrum.

## 3. Generating iris images for testing biometric system

As a part of the research, two algorithms for generating artificial iris images and a way of composing them with real eye photographs have been developed.

The most commonly used images in near infrared systems were used as the reference image. The goal was to create unique images of the iris rings. The remaining elements of the eyeball and surrounding face elements are not taken into account during the extraction process of features, therefore there is no need to ensure their unique appearance. The images generated for the needs of the work were created using the University of Tehran iris image depository UTIRIS[1] [8]. Fig. 1 shows an example sample from the aforementioned database.

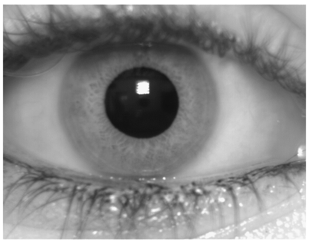


**Fig. 1.** Sample image from the UTIRIS iris data base

The process of eye image generation is as follows. The first step is to generate the texture of the iris ring. The following subsections describe two different approaches to this issue. The first proposes to use the method of noise generation created by Ken Perlin, the second assumes the use of existing iris samples to assemble a new image using the algorithm designed to generate textures developed by Efros and Freeman [5]. Regardless of the technique chosen, a rectangular image is generated (Fig. 2).

The iris strip obtained should then be rolled up to take the form of a ring. It is worth noting, however, that the circle of the inner edge of the iris does not have to be

[1] `https://utiris.wordpress.com/`

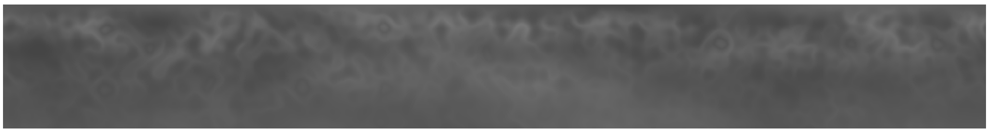


**Fig. 2.** Sample texture generated using Perlin noise

concentric with the circle of the outer edge, which is taken into account in this work. Figure 3 shows the texture image after the transformation.

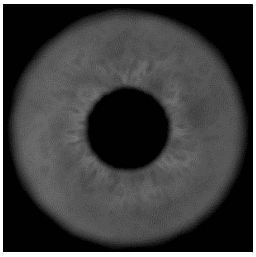


**Fig. 3.** Iris ring folded from the texture from Fig. 2

The last step is to combine the image so obtained with the real image of the eye. For this purpose, two manually created images are used as the bottom and top layer. The bottom layer contains the image of the iris-free eye, the image generated by the algorithm is applied, and then the image of the eyelid, which may partly obscure the pasted image of the iris, overlaps. These images were prepared manually using free, open-source raster graphics editor GIMP. The result of such composition is shown in Figure 4.

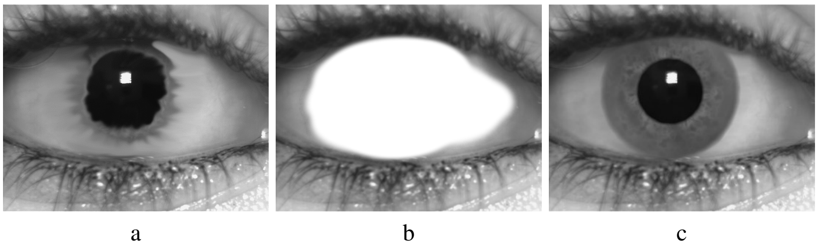


**Fig. 4.** Bottom layer (a), top layer (b), composite image (c)

To provide several biometric samples of the same person, five images of the eye were generated based on each iris strip. In order to simulate different images of the same eye, the generated images differ from each other in the size of the pupil and the slight angular deviation of the iris ring.

### 3.1 Generating iris images using Perlin noise

The first of the methods to generate an artificial iris image described in this work was created as a result of experimental attempts to synthesize Perlin noise samples. The essence of the method is to combine pseudorandom noise images of different frequencies into one that visually reproduces the characteristic irregularities that occur in the iris of the human eye.

At the beginning, using the Perlin algorithm [9], two images are generated with different frequencies (the frequency of the second is two times higher than of the first one), on the basis of which an averaged image is created (Figure 5).

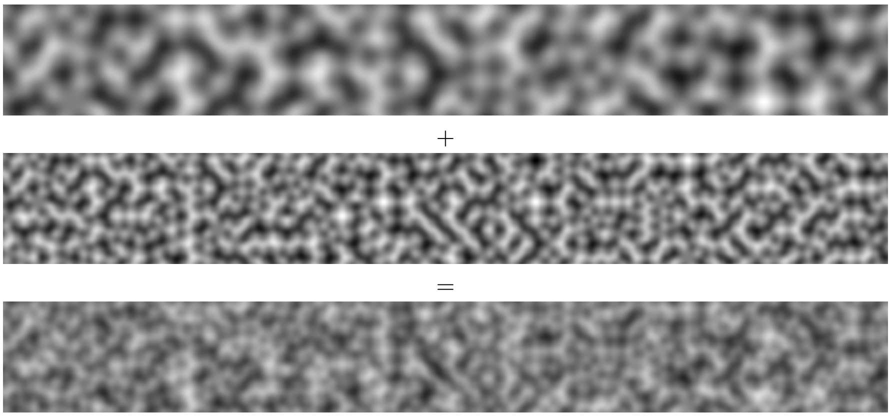


**Fig. 5.** The result of the averaging of two Perlin noise images with different frequencies

In the next stage, in order to obtain details that imitate the iris spots, the image palette is modified. As a result of a series of experiments involving the visual comparison of the effects of various functions, a formula enabling this operation was created:

$$f(x) = \begin{cases} 0.5 + 0.5 \cdot \sin(x \cdot 2\pi + \frac{\pi}{2}) & \text{if} \quad x \leq 0.25 \\ 0.5 + 0.5 \cdot \sin((x - 0.25) \cdot 2\pi) & \text{if} \quad 0.25 < x \leq 0.50 \\ 1 - \sin(x \cdot \pi - \frac{\pi}{2}) & \text{if} \quad x > 0.5 \end{cases} \tag{1}$$

Where the $x$ value for black pixel is 0, and for white is 1. The result of the operation is shown in the Fig 6.

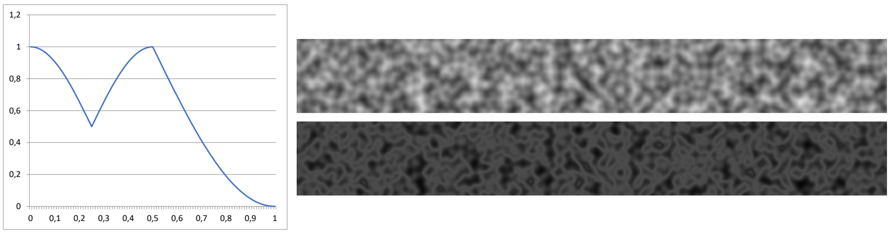


**Fig. 6.** Plot showing the palette function $f(x)$ and the noise image before and after modification

The next step is to reduce the detail level in the received image. For this purpose, another image is generated using Perlin noise (of frequency two times lower than the first of images from the step presented in Fig. 5). It serves as a detail map that adds uniform gray spots to the image being processed (Figure 7).

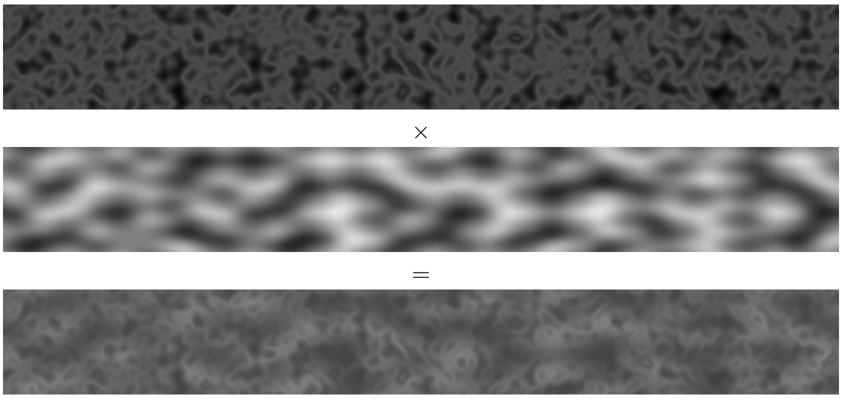


**Fig. 7.** Detail reduction operation

The operation shown in Figure 7 is defined by the formula:

$$y = (1-m) \cdot x + 0.25 \tag{2}$$

Where: $x$ is pixel value of the source image, $m$ – pixel value of the mask. The image obtained in this way will be used in the inner part of the iris ring, where the intensity of the characteristic points is greater. For the outer part, another sample of Perlin noise is created, this time with a small frequency (four time smaller than in the previous step) and amplitude (five times lower). It is presented in Fig. 8.

**Fig. 8.** Image created on the basis of Perlin noise with low frequency and amplitude

The final stage of the algorithm is the combination of the inner and outer parts of the iris using the mask created specially for this purpose (Figure 9). Interpolation along the vertical axis was made using the sine function, irregular elements are the result of applying an additional layer of noise to an unnaturally smooth gradient.

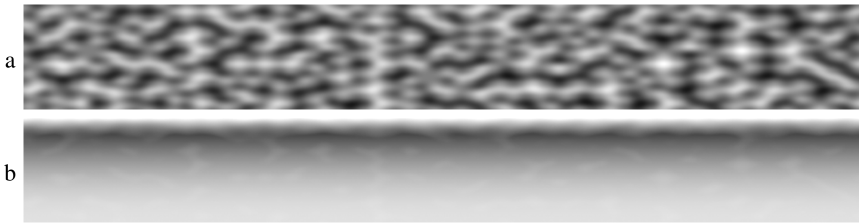


**Fig. 9.** Mask for combining the outer and inner texture of the iris (b) and Perlin noise used to create that mask (a)

Finally, after generating all the necessary elements, the final iris image can be created (Fig. 10). The operators used in this step are the same as in the previous ones.

### 3.2 Generating iris images using image quilting algorithm

The second method to generate the artificial iris image used in the work is based on the image quilting algorithm of Alexei A. Efros and Wiliam T. Freeman [5]. The following description presents the proposed method to use this technique when generating synthetic images of the iris.

Image quilting involves creating images from existing samples. Therefore, in order to generate the texture of the iris, a collection of source images is needed, on the basis of which new images will be created. In the research described in the work,

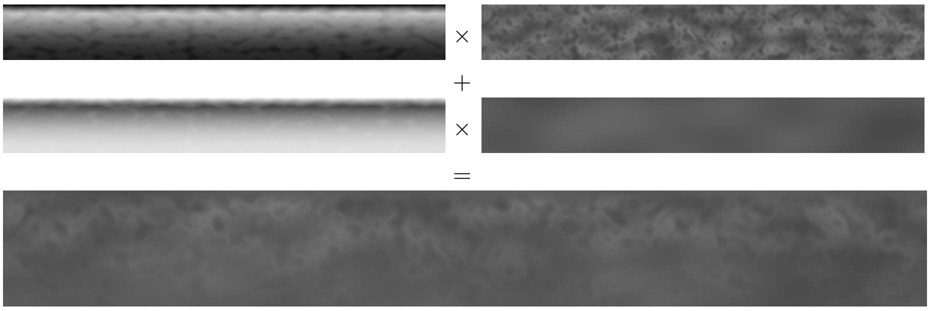


**Fig. 10.** Combining component images into the final iris texture

the UTIRIS database [8] was used. Out of 30 images of different classes, rectangular stripes representing iris (Figure 11) were obtained by unfolding the ring.

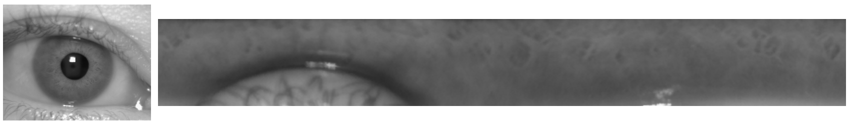


**Fig. 11.** Picture of the eye from the UTIRIS database with its iris after unfolding

From the unfolded iris strips, the fragments corresponding to the eyelids should be removed (Figure 12), and then the brightness of the images should be modified so that the average value of the pixel on each of them is equal. These steps were performed under human control to prevent errors of the automated system.

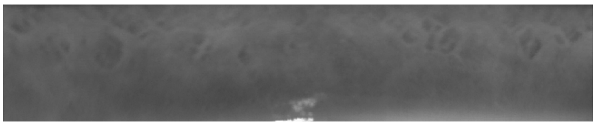


**Fig. 12.** Iris strip from Figure 11 after removing fragments containing the eyelids

Iris textures were made of forty-eight combined source fragments (in three rows, sixteen columns) with dimensions of 64 × 42 or 64 × 43 pixels plus an additional 10 pixels for each edge touching with another fragment (Figure 13.) Iris stripes with a resolution of 1024 × 128 pixels were obtained.

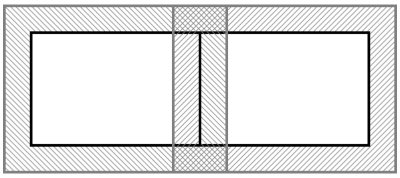


**Fig. 13.** A schematic diagram of the arrangement of two adjacent fragments – black indicates the inner, non-overlapping part, while the gray color – the edge overlapping the adjacent slices

The first step of the algorithm is to draw one of thirty previously prepared iris strips, from which a fragment of the appropriate size is randomly selected. The drawing takes place in the upper half of the iris strip so that the fragment contains the characteristics of the inner ring. The resulting fragment is placed in the upper left corner of the image.

For the remaining forty-seven elements, selecting one from thirty strips is repeated. This time, the drawn strip is searched in order to find a rectangle that best fits the image from the previous step. The decision is made on the basis of the sum of pixel differences in the contact area of the fragments. The smallest sum means that the fragment should blend well with the previously generated texture. The overlapping area is then marked with a path dividing it into two parts. The A* algorithm [7] is used to search for the shortest path, where the road graph is based on a matrix of pixels, and the weights of the points are equal to the square difference of the corresponding pixel values. The selected fragment is cut according to the calculated path and then applied to the resultant image. Fragments of subsequent lines must include two overlapping areas (overlapping the previous part of the line and the previous line). An exemplary image of the generated iris texture strip with marked connection lines is shown in Figure 14.

For the described case, in which the final image is the result of joining forty eight patches selected from thirty iris stripes, the number of possible combinations is $30^{47}$ multiplied by the number of possible random results of the first fragment (value of the order $10^{79}$).

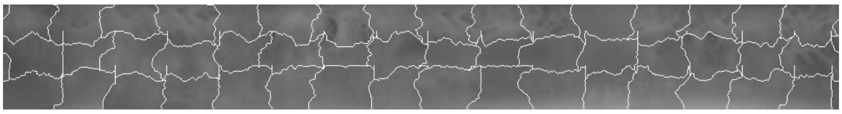


**Fig. 14.** Generated texture of the iris with marked joint lines

### 3.3 Image testing

The main issue addressed in this work is to generate artificial biometric data. It is difficult to assess the quality of the results obtained without first testing them using a biometric system. For this purpose, an application was created that allows to identify and verify the classes of generated images.

The quality indicator of generated images is their degree of uniqueness. In this case, a simplified version of the original Daugman algorithm, implemented in the developed application, will suffice. Simplification consists primarily in omitting the iris segmentation phase.

The implemented algorithm consists of stages of normalization and extraction of features. At the beginning, the iris ring from the loaded image is unfolded into rectangular strip. The next step is to use a convolution operation using Gabor filters. As a result, two matrices are created: real and imaginary. With the help of the coding proposed by Daugman, positive values are converted into ones and negative values are changed to zeros (Figure 15). The length of received codes depends on the resolution adopted during the normalization of the strips containing iris data.

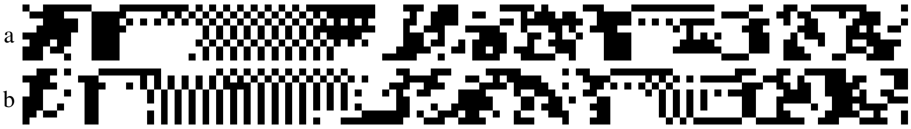


**Fig. 15.** Coded real (a) and imaginary (b) parts of the result

The process of loading and transforming images into binary codes is performed for all tested images. After receiving the set of codes grouped with classes, it is possible to start testing.

## 4. Experimental results

For the purpose of this research, a thousand iris classes were generated using each of three different methods:

- using Perlin noise (subsection 3.1),
- using the image quilting algorithm (subsection 3.2),
- a hybrid approach using image quilting, in which the source images are both the real iris images and those generated using Perlin noise.

For each of classes, five images of the same eye with different degrees of irises were generated. In addition, the images are characterized by a slight angular deviation, with each sample from one class was deviated in a different degree in relation to the others.

From among the generated images, three sets of 250, 500 and 1000 data classes were created (1250, 2500 and 5000 images respectively). All collections were tested in terms of both identification and verification using leave-one-out crossvalidation method.

The generated images were tested using the algorithm described in chapter 3.3. Performance metrics for a biometric data array of sizes $32 \times 4$, $32 \times 8$, $64 \times 8$ and $128 \times 8$ bits were compared.

## 4.1 Iris images generated using Perlin noise

The Ken Perlin's noise algorithm is characterized by high speed. Generator, created for this reaserch, managed to produce 1090 different iris classes in one hour (data for Intel Core i5-520M).

The results of verification tests (equal error rate values) are presented in Table 1. As one can see, the use of $64 \times 8$ bits matrix already provides virtually zero ERR value.

**Table 1.** Error of verification for Perlin images

| Feature matrix | 250 classes | 500 classes | 1000 classes |
|---|---|---|---|
| $32 \times 4$ | 1% | 0.8% | 0.9% |
| $32 \times 8$ | 0.06% | 0.05% | 0.05% |
| $64 \times 8$ | 0% | 0% | 0% |
| $128 \times 8$ | 0% | 0% | 0% |

The results of identification tests are shown in Table 2. For each of the tested sets using the matrix $64 \times 8$ and $128 \times 8$, an accuracy of 100% was obtained.

## 4.2 Iris images generated using image quilting

The disadvantage of this algorithm is its computational complexity. The generator created for the needs of this work, managed to produce only 102 different iris classes in one hour (data for Intel Core i5-520M).

The results of verification are presented in Table 3. A 0% equal error rate values were obtained for matrices with sizes $128 \times 8$ bits.

**Table 2.** Results of identification for Perlin images

| Feature matrix | 250 classes | 500 classes | 1000 classes |
|---|---|---|---|
| $32 \times 4$ | 99.6% | 99.24% | 99.1% |
| $32 \times 8$ | 100% | 99.96% | 99.98% |
| $64 \times 8$ | 100% | 100% | 100% |
| $128 \times 8$ | 100% | 100% | 100% |

**Table 3.** Error of verification for quilt images

| Feature matrix | 250 classes | 500 classes | 1000 classes |
|---|---|---|---|
| $32 \times 4$ | .3% | 2.4% | 2.4% |
| $32 \times 8$ | 0.55% | 0.57% | 0.57% |
| $64 \times 8$ | 0.01% | 0.02% | 0.02% |
| $128 \times 8$ | 0% | 0% | 0% |

The results of identification tests are presented in Table 4. As before, for each of the tested sets using the matrix $64 \times 8$ and $128 \times 8$, an accuracy of 100% was obtained.

**Table 4.** Results of identification for quilt images

| Feature matrix | 250 classes | 500 classes | 1000 classes |
|---|---|---|---|
| $32 \times 4$ | 96.72% | 94.96% | 93.00% |
| $32 \times 8$ | 99.76% | 99.84% | 99.78% |
| $64 \times 8$ | 100% | 100% | 100 |
| $128 \times 8$ | 100% | 100% | 100% |

### 4.3 Iris images generated using hybrid algorithm

Images are generated at the same speed as those described in previous subsection, because the hybrid technique uses an identical algorithm. Only the source elements from which the images are created are changed.

The generated images were tested analogously to the images described in previous subsections, the results are presented in Tables 5 and 6.

In the verification tests, for a features array of $128 \times 8$, the result of 0% equal error rate was obtained. The $64 \times 8$ and $128 \times 8$ matrices provide 100% correct identification for all three data sets.

**Table 5.** Error of verification for hybrid images

| Feature matrix | 250 classes | 500 classes | 1000 classes |
|---|---|---|---|
| $32 \times 4$ | 1.5% | 1.4% | 1.4% |
| $32 \times 8$ | 0.18% | 0.2% | 0.2% |
| $64 \times 8$ | 0% | 0.01% | 0.01% |
| $128 \times 8$ | 0% | 0% | 0% |

**Table 6.** Results of identification for hybrid images

| Feature matrix | 250 classes | 500 classes | 1000 classes |
|---|---|---|---|
| $32 \times 4$ | 99.20% | 98.40% | 97.56% |
| $32 \times 8$ | 100% | 100% | 99.92% |
| $64 \times 8$ | 100% | 100% | 100% |
| $128 \times 8$ | 100% | 100% | 100% |

### 4.4 Results comparison

It can be seen, that the best verification results have been obtained with images created with the algorithm that uses Perlin noise. Identification tests gave similar results. Except one case of a $32 \times 8$ matrix for a set of 500 classes, images based on Perlin noise showed the highest percentage of correctly identified samples.

Based on the obtained results, it can be concluded that the images generated using the Perlin noise technique are characterized by the highest degree of uniqueness. At the same time, it is the fastest method and allows to generate a large collection of images with unique biometric data classes.

It is worth noting that all tests carried out on matrices with sizes $128 \times 8$ bits (a total of 2048 bits) fell one hundred percent successfully. The size of the vectors tested by Daugman in 1993 [3] was just 2048 bits. The number of classes tested by him is 323 – more than three times smaller than in this work. It can therefore be assumed that synthetically generated irises are characterized by a sufficiently high diversity.

### 4.5 Comparison of generated images with the real ones

The strength of the authentication system is determined not only by the algorithms for processing attributes and classifications used, but also the correct calibration – that is, determining the thresholds for which the system rejects the person applying for access. For this purpose, training data is necessary. They can be real data or artificial data obtained using the methods described above.

To ensure that iris images generated by the developed methods are suitable for testing and calibrating biometric systems to a similar extent as real images, comparative studies were carried out. For this purpose, four sets of iris pictures were collected, 50 classes each. Three containing iris formed on the basis of each of the methods discussed and one consisting of real images from the UTIRIS iris database. Each of the sets was subjected to verification tests allowing to determine optimal thresholds and values of balanced errors using a data matrix with sizes of 32 × 4, 32 × 8, 64 × 8 and 128 × 8 bits. The results obtained are presented in Table 7.

**Table 7.** Thresholds (TH) and errors (ERR) received during verification tests

| | 32x4 | | 32x8 | | 64x8 | | 128x8 | |
|---|---|---|---|---|---|---|---|---|
| | TH | ERR | TH | ERR | TH | ERR | TH | ERR |
| Perlin's noise | 0.9 | 0.5% | 1.6–2 | $\sim 0\%$ | 1.7–2.8 | 0% | 2.2–3.1 | 0% |
| Image quilting | 1.05 | 1.3% | 2.1 | 0.1% | 2.4–2.8 | $\sim 0\%$ | 2.7–3.1 | 0% |
| Hybrid approach | 0.95 | 0.6% | 1.8–2.1 | $\sim 0\%$ | 2–2.8 | 0% | 2.5–3.1 | 0% |
| UTIRIS | 1.15 | 13% | 2.4 | 11.9% | 3 | 4% | 3.3 | 2% |

The threshold values shown in the Table 7 indicate the maximum acceptable mean number of matrix column elements on which the bits do not match. For each of the sets tested with the same matrix sizes similar values of optimal thresholds were obtained. The data set based on the UTIRIS database is characterized by the highest values. The most close results for generated data are obtained by the image quilting technique. On the other hand, the images generated using Perlin noise are characterized by the threshold values most deviating from natural samples.

The obtained values of the equal error rate differ to a larger extent between sets of artificial and real images. For the smallest size matrix among the tested, they range from 0.5% to 1.3% in the case of artificial collections, while the real images get as much as 13%. For larger sizes, the error value drops to 2%, while for artificial images it approaches 0%.

The reason for such large discrepancies in the obtained values of the equal error rates can be a simplified iris segmentation model applied during the performed tests. While the images generated contained exact data about the position and size of the iris ring, in the case of photos from the UTIRIS database it was necessary to determine these data in advance. In the case of any errors, the iris strips may have become distorted at the standardization stage, negatively affecting later attempts to match the pattern. The threshold values determined on the basis of the research of each of the collections turned out to be similar. In this case, the generated images of iris seem to fulfill their role, which is to enable the initial calibration of biometric systems without the need to obtain real biometric data.

## 5. Conclusions

Research on images created by designed methods has demonstrated the utility of simple pseudorandom algorithms, such as Perlin noise, in the field of biometrics. The authors' work, unlike the approaches presented in section 2.2, is not based on realistic modeling the eye structure. It was proposed to use universal image processing methods to create images that imitate images of real irises. Despite the trivial solution, it allows to test biometric systems. In the case when a large number of classes is an important element of research, Perlin noise seems to be a good solution, because it allows to generate a large data set with a low calculation cost. This feature even allows to generate data on the fly, avoiding the need for storage.

Images synthesized using the method of image quilting, apart from uniqueness enabling creation of data sets, showed a visual similarity to real images. This raises the question of whether other commonly used methods for generating textures are also suitable for use in the field of eye biometrics testing. Probably these universal techniques can also be applied to other biometric features. The presented solution is the first approach of the authors to the topic of generating artificial biometric data, other possibilities will be investigated in the future.

## References

[1] Luís Cardoso, André Barbosa, Frutuoso Silva, António M. G. Pinheiro, and Hugo Proença. Iris biometrics: Synthesis of degraded ocular images. *IEEE Transactions on Information Forensics and Security*, 8(7):1115–1125, July 2013.

[2] Jiali Cui, Yunhong Wang, JunZhou Huang, Tieniu Tan, and Zhenan Sun. An iris image synthesis method based on pca and super-resolution. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 471–474 Vol.4, Aug 2004.

[3] John Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, Nov 1993.

[4] John Daugman. New methods in iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1167–1175, Oct 2007.

[5] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 341–346, New York, NY, USA, 2001. ACM.

[6] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[7] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[8] Mahdi S. Hosseini, Babak N. Araabi, and Hamid Soltanian-Zadeh. Pigment melanin: Pattern for iris recognition. *Instrumentation and Measurement, IEEE Transactions on*, 59(4):792 –804, april 2010.

[9] Ken Perlin. Improving noise. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 681–682. ACM, 2002.

[10] Clarence E. Rash, Michael B. Russo, Tomasz R. Letowski, and Elmar T. Schmeisser. Helmet-mounted displays: Sensation, perception and cognition issues. Technical report, ARMY AEROMEDICAL RESEARCH LAB FORT RUCKER AL, 2009.

[11] Samir Shah and Arun Ross. Generating synthetic irises by feature agglomeration. In *Image Processing, 2006 IEEE International Conference on*, pages 317–320. IEEE, 2006.

[12] Jinyu Zuo, Natalia A. Schmid, and Xiaohan Chen. On generation and analysis of synthetic iris images. *IEEE Transactions on Information Forensics and Security*, 2(1):77–90, 2007.

# GENEROWANIE SYNTETYCZNYCH OBRAZÓW TĘCZÓWEK DLA CELÓW TESTOWANIA SYSTEMÓW BIOMETRYCZNYCH

**Streszczenie** Rośnie popularność stosowania metod biometrycznych w identyfikacji osób, wielu badaczy pracuje nad nowymi systemami, a wymaga to danych, na których można by je testować. W niniejszej pracy zaproponowano sposób generowania sztucznych danych biometrycznych dla potrzeb testowania systemów bazujących na biometrii tęczówki ludzkiego oka. Przedstawiono dwie metody – jedna bazuje na szumie Perlina, natomiast druga na algorytmie generowania tekstu image quilting. Przedstawiono również podejście hybrydowe, łączące obydwie metody. Zawarto również wyniki testów, mających pokazać użyteczność obrazów wygenerowanych przy pomocy zaproponowanych metod w opisywanym zastosowaniu. Pokazano również ich spójność i podobieństwo do prawdziwych danych.

**Słowa kluczowe:** tęczówka, biometria

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

# PREPROCESSING PHOTOS OF RECEIPTS FOR RECOGNITION

Wojciech Korobacz, Marek Tabędzki

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

**Abstract:** The subject of this work is methods of image pre-processing, applied to receipts photos. The purpose is to improve their quality, allowing to increase the efficiency of the conventional text recognition software (OCR). The authors had mainly difficult cases in mind – photos taken freehand in unfavorable lighting conditions. The work describes the analyzed methods of filtering, binarization, searching for the edge of the image, image straightening, marking the area of interest, thinning. The preliminary results with OCR software on a small data set were also presented. Thanks to pre-processing, character recognition efficiency has been improved by 25%. The final part presents conclusions and plans for future work.

**Keywords:** digital image processing, optical character recognition

## 1. Introduction

The subject of this work is a method of pre-processing of photographs containing receipts to a form readable by conventional text recognition software (OCR). This is an important issue from the point of view of technology development, in particular mobile and interactive technologies. The smartphone has become the object of everyday use these days. A portable computer with a camera and Internet access gives almost unlimited potential for practically every area of our lives. Only a few years ago, the peak of the possibilities was a telephone that, apart from calling and sending messages, was able to take pictures or store notes. Modern smartphones are used as handheld instant messengers, cameras, alarm clocks, schedules, information and news sources, gaming centers and much more. Today, we want to have everything on the smartphone that will make life easier for us. This idea is in line with the latest IoT trends "Internet of Things" or even newer IoE "Internet of Everything".

Following this idea, the smartphone has also become an e-wallet, with the help of NFC (Near Field Communication) we can, for example, pay for purchases and

Advances in Computer Science Research, vol. 14, pp. 87-103, 2018. DOI: 10.24427/acsr-2018-vol14-0006

collect data about expenses. However, only the amount can be drawn from the payment information, but not the content of our purchases. The fastest and easiest way to achieve this goal is to take a photo of the receipt using the application on our smartphone, which will properly process the photo and create a text file with information from the receipt. Then, this information should be processed, that is, to extract relevant information from unnecessary information, group them and save in a way that allows later analysis.

The problems that the following work deals with are the randomness and uniqueness of photographs, i.e. inhomogeneous lighting conditions, cropping, different angles of images taken, non-linear distortions and sharpness of images. For such conditions, it is necessary to analyze, study and select the best algorithms of image processing in such a way that the final result is as close as possible to the later readout by OCR software.

This issue is not new. With the appearance and popularization of smartphones, researchers considered the possibility of using their cameras to scan documents and read text. The quality of this type of material has always been a problem, significantly differing from the images obtained with the use of a flatbed scanner (usually expected by the OCR software). The influence of the environment resulted in degradation of quality, lowering readability. Therefore, the work devoted to preprocessing techniques and quality improvement, eg Sharma's work [13] treats about the impact of noise and blur, while Bieniecki in his work [1] emphasizes the correction of perspective and orientation. As a confirmation of the effectiveness of the proposed methods, likewise the authors of this paper, he uses existing OCR software. A similar direction was chosen by Shen [14], however, focusing on removing the background that could disrupt, thus negatively affect the process of text recognition – the answer is the appropriate thresholding procedure, which is able to get rid of the unwanted background image. Research shows that a properly selected pre-processing procedure is able to improve the quality of recognition, eg Wiraatmaja in his work [18] noted a quality increase of more than 25% compared to the standard approach, while Brisinello [2] in his low-resolution and low-quality image studies, for which simple OCR software provided only 35% of recognition efficiency, improved quality through preprocessing procedures by over 33%.

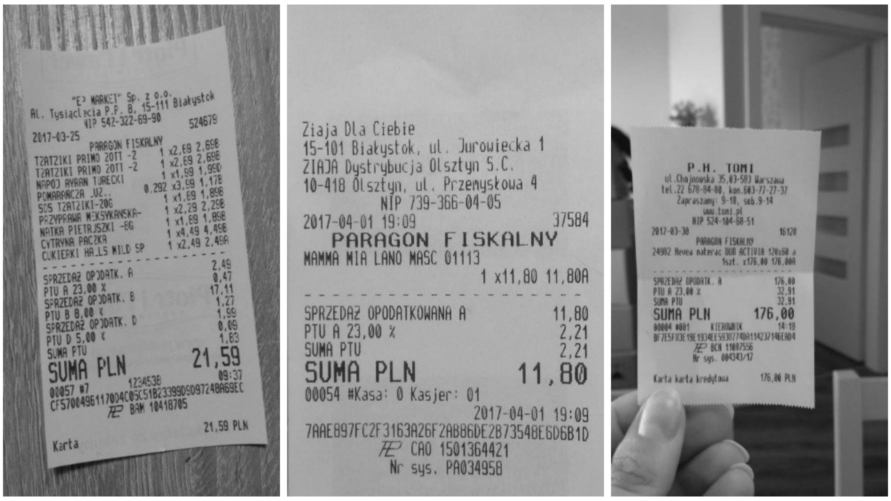
This publication presents the results of research carried out as part of the master's thesis of one of the authors [5]. Presented work explores available solutions and algorithms in the field of digital image processing. It does not deal with the next stage, which is the identification of data read from the image. Therefore, it should be treated as a preliminary to further research, which will allow to create a full system implementing all of the tasks described above.

## 2. Problem and solution concept

To select the appropriate samples for testing, the classes of photo receipts we can deal with were determined. The following characteristics of the samples were considered:

- Cropping – whether the entire receipt is visible, how much background is in the picture,
- Lighting – it can be artificial or natural, strong or weak, shadows can be seen on the receipt,
- Sharpness – whether the photo is sharp or blurred,
- Angle of rotation – how much the photo deviates from the vertical position,
- Folds – the receipt may be curled or folded.

On this basis, a database of samples for research was collected (some of them are shown in Figure 1), remembering their diversity and taking care of the representation of each of the mentioned classes.



"E> MARKET" Sp. z o.o.
Al. Tysiąclecia P.P. 8, 15-111 Białystok
NIP 542-322-69-90
2017-03-25 524679
PARAGON FISKALNY
TZATZIKI PRIMO ZOTT -2 1 x2,69 2,69B
TZATZIKI PRIMO ZOTT -2 1 x2,69 2,69B
NAPOJ AYRAN TURECKI 1 x1,99 1,99D
POMARANCZA _UZ.. 0,292 x3,99 1,17B
SOS TZATZIKI-20G 1 x1,89 1,89B
PRZYPRAWA MEKSYKANSKA- 1 x2,29 2,29B
NATKA PIETRUSZKI -EG 1 x1,89 1,89B
CYTRYNA PACZKA 1 x4,49 4,49B
CUKIERKI HALLS MILD SP 1 x2,49 2,49A
SPRZEDAZ OPODATK. A 2,49
PTU A 23,00 % 0,47
SPRZEDAZ OPODATK. B 17,11
PTU B 8,00 % 1,27
SPRZEDAZ OPODATK. D 1,99
PTU D 5,00 % 0,09
SUMA PTU 1,83
SUMA PLN 21,59
00057 #7 1234538 09:37
CF57004961170D4C05C51B23399D9D9724BA69EC
BAM 10418705
Karta 21,59 PLN

Ziaja Dla Ciebie
15-101 Białystok, ul. Jurowiecka 1
ZIAJA Dystrybucja Olsztyn S.C.
10-418 Olsztyn, ul. Przemysłowa 4
NIP 739-366-04-05
2017-04-01 19:09 37584
PARAGON FISKALNY
MAMMA MIA LANO MASC 01113
1 x11,80 11,80A
SPRZEDAZ OPODATKOWANA A 11,80
PTU A 23,00 % 2,21
SUMA PTU 2,21
SUMA PLN 11,80
00054 #Kasa: 0 Kasjer: 01
2017-04-01 19:09
7AAE897FC2F3163A26F2AB86DE2B73548E6D6B1D
CAO 1501364421
Nr sys. PA034958

P.H. TOMI
ul.Chojnowska 35,03-583 Warszawa
tel.22 678-04-80, kom.603-77-27-37
Zapraszamy: 9-18, sob.9-14
www.tomi.pl
NIP 524-104-68-51
2017-03-30 16128
PARAGON FISKALNY
24902 Hevea materac DUO ACTIVIA 120x60 a
1szt. x176,00 176,00A
SPRZEDAZ OPODATK. A 176,00
PTU A 23,00 % 32,91
SUMA PTU 32,91
SUMA PLN 176,00
00004 #001 KIEROWNIK 14:18
BF7E5F83E19E1934EE5938774DA114237146EAD4
BCN 11087556
Nr sys. 004343/17
Karta karta kredytowa 176,00 PLN

**Fig. 1.** Examples of collected data

In total, about 240 samples were collected – photographs of various receipts taken freehand using smartphone. All receipts were in Polish, but they could be different in font, content layout, shade and dimensions. It was ensured that the picture contained a receipt of at least 20% of the surface. Because the pictures were taken using different cameras, they could have different resolutions. Collected samples were grouped in 9 visually similar classes (in terms of the above-mentioned features), and then the subsequent stages were evaluated by researchers separately in each group.

The following test run was specified:

- Pre-processing: image conversion to grayscale, filtering, contrast enhancement,
- Binarization: conversion to black and white,
- Finding the receipt edge in the image,
- Straighten: rotate to a form in which the text is in a horizontal position,
- Finding the area of interest: location of the text in the picture,
- Thinning: changing the text to one pixel wide form,
- Tests of processed images in OCR programs.

The above steps to prepare a picture and read the text from it were considered in turn, and then the final solution was formulated, which will be presented further. The pre-processing stages were subjected to preliminary visual assessment. Often it was easy to assess whether the choice of methods or parameters improves the quality of the image or causes its further degradation. On this basis the most promising method was indicated and passed to further stages, including final experiments with OCR software.

The Java programming language was used for the implementation and the algorithms available in the open-source OpenCV library were used.

## 3. Filtering

The first step in processing the photo of the receipt was its conversion to a gray scale image. This operation does not require a specific description, as the OpenCV library procedures have been used. The image was converted from a 24-bit to an 8-bit form according to the following equation.

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{1}$$

Where $R$, $G$ and $B$ are red, green and blue intensities, and $Y$ is the output intensity.

A more important element of the pre-processing was smoothing and histogram equalization. Smoothing operation reduces noise in the image (but also blurs the focus). It was implemented using a Gauss $5 \times 5$ filter. The histogram shows the distribution of the brightness values in the image [6,4]. The operation of histogram equalization (flattening) improves the contrast by extending the histogram to the full gray scale and spreading out the most frequent intensity values.

In the next steps, both filtered and unfiltered images were taken into consideration to assess the impact of the pre-processing on the result.

### 3.1 Binarization

Image binarization (or thresholding), is changing the image from the gray scale, usually 8-bit, into an image of two intensities, white and black [9]. This operation accomplishes the task of segmentation – it separates the subject of interest (text, in the case of receipts) from the background in the examined image.

Binarization methods can be divided into three categories: global, local and adaptive [6]. In global methods, one threshold value is set for the whole image, based on which it decides whether a pixel of a given brightness belongs to the text or the background. This is a relatively simple task in the case of clear, good-quality photos with high contrast between the text and the background. In more problematic cases, it is necessary to use the local methods or adaptive ones. In local methods, the threshold value is determined for a fixed-size window, while in adaptive ones, it is determined separately for each pixel.

Three binarization methods were examined (Eq. 2). The first one was the classic global Otsu method [7]. It is based on a discriminant analysis. The threshold is determined on the basis of minimizing intra-class variance and maximizing inter-class variance. The other two methods were adaptive methods [4]. In methods of this type, an important parameter is the distribution of the influence of neighbors in the observation window on the examined pixel. This is shown in the following equation:

$$Y'_{x,y} = \begin{cases} 1 & \text{if } Y_{x,y} > T_{x,y} - C \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $Y_{x,y}$ is a source pixel intensity, $Y'_{x,y}$ is destination intensity (zero-one value), $T_{x,y}$ is a threshold calculated individually for each pixel, and $C$ is a equivalent of the coefficient $k$ in simplified Niblack method [17]. For the first method, the threshold value $T$ is a mean of the pixel intensities in the observation window. For the second one, it is a weighted sum (cross-correlation with a Gaussian window) of this neighborhood.

The second parameter examined was the size of the observation window. The size of the window was dependent on the size of the input image and two versions were tested for each method: 1% and 3% of the width of the image being examined (larger windows were rejected due to blurring and merging of letters). In addition, as described earlier, versions with and without histogram equalization and smoothing were examined. In the case of smoothing with a Gauss filter, different values of the $\sigma$ sigma factor (determining the weight of subsequent neighbors in the mask) were considered. Selected results are presented in Figure 2.

Visual evaluation of the processed images led the authors to the conclusion that Otsu's global method copes well with clear, sharp images with a good lighting. Unfortunately, with the inferior input material, this method fails. In the adaptive methods

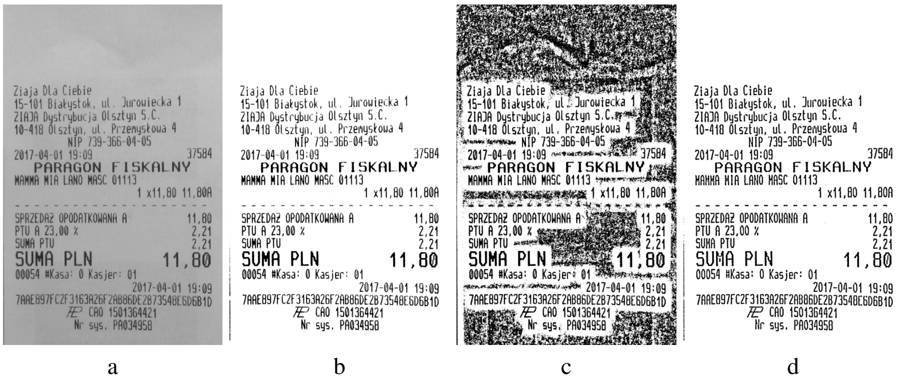


**Fig. 2.** Sample image (a) and same image after binarization: with Otsu method (b), adaptive method with 3% window (c), adaptive method with $C = 5$ (d)

it can be observed that along with the growing observation window, the text becomes less readable, which should be considered an undesirable situation, as it will negatively affect the ability to recognize the text.

Both alignment and smoothing of the histogram have a significant impact on improving the readability of the image for humans, but it does not equal the positive impact on the binarization or any further recognition process. In particular, the histogram equalization, due to the loss of some information, introduced disturbances and caused problems in the binarization. In the case of smoothing, only for low sigma values this has a positive effect on the result.

Because the observations showed that it is not possible to indicate the best solution for each type of photo, an additional factor $C$ was introduced in the adaptive methods (equation 2). Subsequent tests, carried out on the photographs with and without smoothing (Figure 2d), allowed to obtain the highest quality for the $C = 5$ and an observation window of a 1% width of the input image (in the adaptive method with a homogeneous neighbors weight distribution) and a 3% window (for the adaptive method with the Gaussian distribution). A larger window of observation reduces the impact of noise, but also blurs the text. In turn, too small a window, adversely affects the continuity of letters.

Comparing visually, the adaptive method with the equal weights in the observation window is the best, but full tests along with the subsequent stages will give the final verdict.

Several tests were carried out to evaluate the processing time and they showed that the differences between the individual methods are significant. The Otsu method usually dealt with image processing within 12-14 ms, the adaptive method with equal weights usually took 32-37 ms, while the most time-consuming method is the adaptive method with neighbors weights Gauss distribution, which needed over 100 ms for an observation window of 1% of the image width and more than 300 ms for a 3% wide window, so the time consuming increased with the increase of the number of neighboring pixels to be tested.

It should be noted that these tests were performed on a desktop computer (Intel Core i5-520M processor, 4GB RAM). Tests on a real mobile device will be carried out in the future. It will then be possible to assess whether the processing time should be the decisive factor when choosing the method. The mentioned processing time is always an additional overhead, perhaps, however, it will become an irrelevant factor in the duration of the entire process. It may also turn out that a properly selected preprocessing approach will shorten the time of image analysis by the OCR module. This will become the subject of further research.

### 3.2 Edge detection

The next tested process was edge detection. Its goal was to find the edge of the receipt, to be able to determine its rotation and determine the necessity of straightening (and also find the right angle).

It was decided to implement this with the help of Canny's well-known and proven method [3]. Attempts to process photos without prior processing showed that the resulting image is too noisy for further work, therefore pre-smoothing the image with a low-pass filter is necessary. Two different Gauss smoothing filters were adopted in the study, with coefficients such as in the examination of the binarization methods (observation window 1% of the image width and sigma $\sigma = 2$ and observation window 2% of the image width and sigma $\sigma = 3$). In addition, different threshold values in hysteresis were tested.

The best results were obtained with less smoothing (smaller window and sigma coefficient), for thresholds $TD = 100$ and $TH = 120$ (Figure 3b). At lower threshold values, the results are noisy, while at higher ones, information is lost.

For comparison, the method of the morphological gradient was also tested (Figure 3c). This is the difference in morphological dilatation and erosion [9]. Various shapes of the $3 \times 3$ structural element (square, round and cross) were used.

Unfortunately, taking into account the basic goal – determining the receipt angle – both of the tested methods proved to be insufficient. Therefore, it was decided to propose another solution for finding the contours of the receipt.

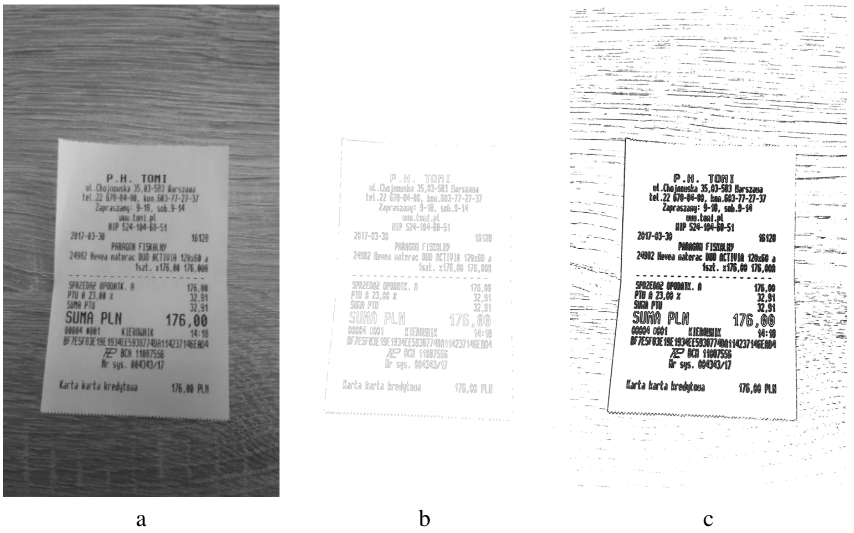


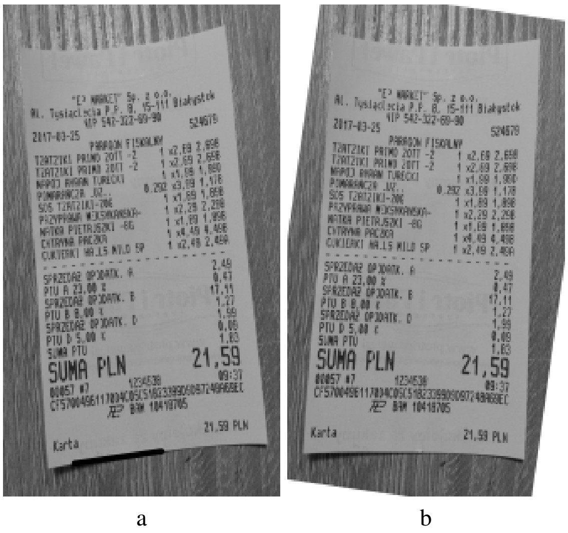
**Fig. 3.** Original image (a), Canny edge detector (b), morphological filter (c)

### 3.3 Rotation angle detection

Due to the unsatisfactory results of the edge detection algorithms, it was decided to test other approaches, this time with only the need to find the right angle, allowing for correction of receipt rotation.

The first of the approaches considered is based on the use of Hough transform. Hough transform allows you to find specific shapes in the image. Mostly these are lines, circles, but there are also solutions that allow you to search for more complex shapes. Its advantage is that it works even when the shapes do not maintain continuity. A detailed description of the method can be found in [8,4]. Its idea is to transfer individual pixels into the Hough space described with the shape parameters that we want to detect. In this way, for example, the line in the input image will be represented by a point in Hough space. When looking for the maximum values in Hough's space, we can find the location of the lines in the input image.

The prepared image had to be binarized in such a way that it contained as little noise as possible. From the point of view of this method, the sharpness and legibility of the detected elements were an irrelevant parameter. Thus, mathematical morphology (gradient method) was used. Then, in the iterative process, using the Hough transformation, lines that met the length criterion were searched for in the image. In the first step, a line that was going through the entire image was searched. If it was not detected, a line twice as short was searched, which was then repeated in the loop until one or more lines were found. For the found lines, their angle of inclination was calculated in relation to the vertical – their average was the angle at which the image should be rotated (Figure 4).



a b

**Fig. 4.** Input image (a), rotated image (b)

Experiments with images have shown that the above algorithm is perfectly suited for detecting the angle of rotation in situations where a photo of the entire receipt is in the frame, i.e. the outer edges of the receipt are visible. In a situation where there is only the content of the receipt in the frame, the algorithm is able to detect only the lines of the largest letters. The problem was also when the receipt was wrapped or folded.

Another tested algorithm was the original idea of looking for an angle based on the course of the text lines in the image. The first step was the binarization and denoising of the input image. The results obtained in the previous steps were used and the Gaussian filter with the size of 2% of the image width and $\sigma = 3$ was selected and the adaptive thresholding method with equal neighbors weighed, the observation window being 1% of the image width and the $C = 5$ coefficient. Then, in the loop for angles from $-10$ to 10 degrees, a vertical histogram of the image is calculated. For each angle tested, the sum of the pixel values of each line is calculated. On the basis of the average brightness, the lines referred to as "white" (with the highest brightness with a threshold of 30%) are selected. The angle for which the number of “white” lines was the highest is the sought angle.

Although the proposed method in many cases proved to be effective, we also found the classes of images for which it failed. The problem was the cases when a large part of the image was taken up by the background. It works best on images containing the text itself, where the angle was detected correctly. This created the need to develop a more effective method.

### 3.4 Text outline detection

In the next stage of the research, a method of universal, automated detection of the text outline was proposed based on existing solutions. It consists of three steps:

1. Pre-rotating the image
2. Detecting the entire text outline and marking it in the image
3. Cutting out the image of the smallest area containing the entire text

**Image rotation** The first step was to pre-rotate the picture to the correct position. For this purpose, the Hough transform was used as described in the previous chapter. Rotation is important because otherwise, even if the area with text is found correctly, it can be cut with the background.

**Outline detection** The second stage was to find the position of the text with the rejection of an unnecessary background. For this purpose, two algorithms were used.

The first one was based on Canny’s edge detection algorithm. Its course was consistent with the description in the previous part of the work – i.e. with binarization, the window of observation 5% of the width of the image and the coefficient $C = 5$. Then, the morphological operation of erosion with the element $3 \times 3$ was used. This led to merging the characters of the text into one blot (Figure 5).

The second of the considered algorithms was based on a high-pass filter based on edge detection using the Sobel operator. This is one of the convolution filters specializing in edge detection [12]. Due to its properties it is quite resistant to noise. The remaining steps were the same as above – until the text was merged. The only difference was that the erosion was carried out in the loop ten times, because once erosion resulted in insufficient merging.

**Image cropping** The third stage was the same, regardless of the previously chosen methods. It assumed an input of a with the text in the form of merged blots. At this

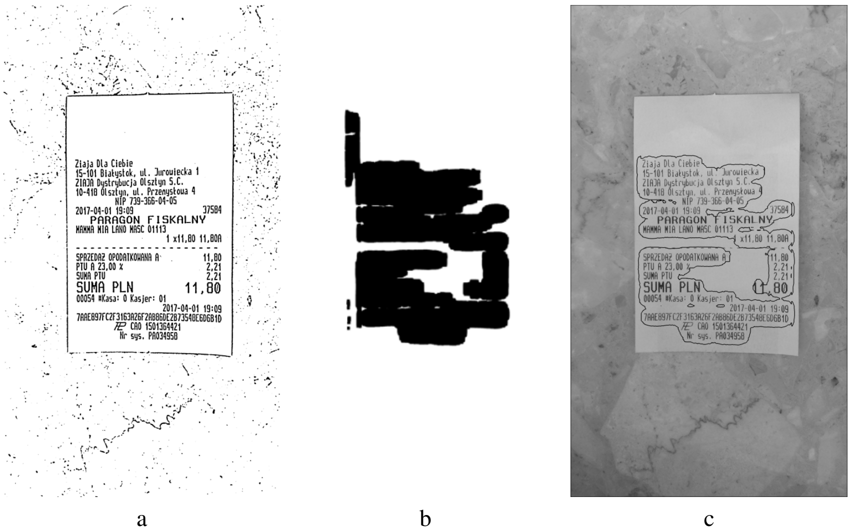


**Fig. 5.** Sample binarized image (a), after erosion (b), shape marked on original image (c)

stage, it was necessary to correctly find these blots and then cut out the smallest fragment containing them from the image.

The first step was to find all the contours based on the input image. Due to the optimization of operation, the simplified representation of contours was taken.

Then, the found outlines were filtered out. Because the algorithm finds the contours of even the smallest noise, the found outlines with an area of less than 0.1% of the area of the input image were discarded. Contours larger than 95% of the image area were also discarded (to eliminate the outline of the whole image). All contours whose surface area contained within this interval were taken into account as significant from the point of view of the described algorithm.

Then, the rectangles escribed on the given contours were found. Next, the rectangle escribed on the whole set of contours was searched for (Figure 6).

It was accomplished by simple algorithm that finds the minimum rectangle containing a set of rectangles. It was only necessary to make sure that the found rectangle did not go beyond the size of the original image. On this basis, a smaller image with dimensions matching the found contour of the text was cut from the original image.

The experiments on the collected samples allowed us to observe that better results are obtained by an approach based on the Canny algorithm, because it found a contour closer to the content of the receipt shown in the pictures.

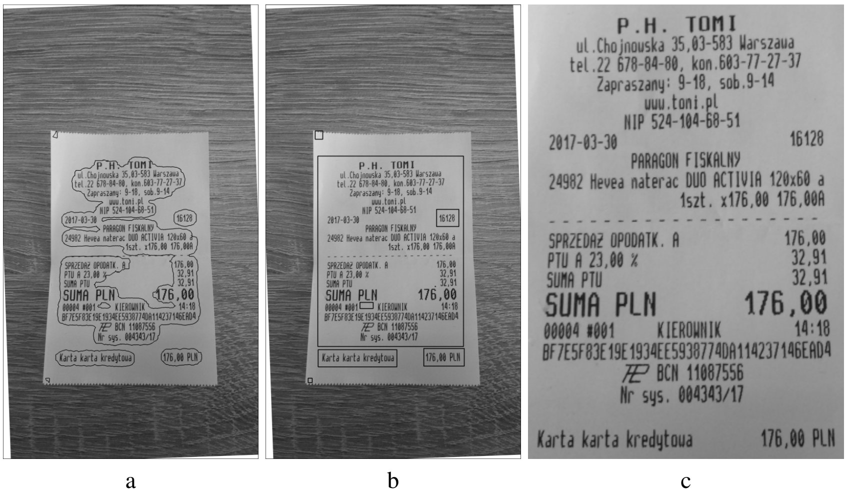


**Fig. 6.** Image with found shapes marked (a), bounding rectangles (b), cropped image (c)

### 3.5 Thinning

The task of the thinning stage is to represent the shapes shown in the image with the form of lines with a width of one pixel (Figure 7). This is helpful and sometimes even required in some text recognition methods. Some OCR software can do it on its own, while some can work better without it (if it is based on the full shape of the letter, not the lines themselves).

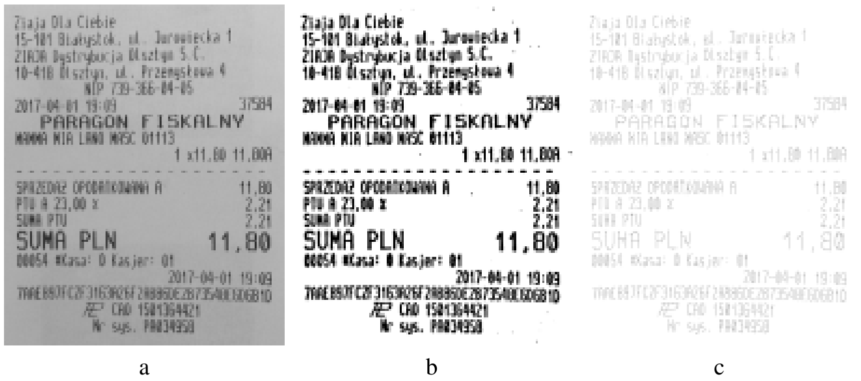


**Fig. 7.** Input image (a), image after binarization (b), image after skeletonization (c)

The work did not test different methods of thinning, however, based on previous experience of the authors, they propose the K3M skeletonization algorithm [10,16]. It is a fast and effective method that gives visually good results and has proven itself in many related applications. The influence of skeletonization on the OCR read will be discussed in the next chapter.

## 4. Optical Character Recognition

The final purpose of the procedure of processing a receipt photo is to recognize the text located in it. Therefore, it was necessary to test obtained images trying to identify the text located on them and in this way to assess the effectiveness of the proposed techniques. As part of this work, the original recognition system was not implemented, but available implementations were used. The first of them is the open-source Tess4j[1] library and the ITesseract class it offers. It is a simple and not flawless method, but it was decided to use it for comparative purposes.

The second application was the commercial ABBYY Fine Reader[2] application, one of the most popular applications of this type. The work uses the available trial version, which allowed for 50 readings, which was insufficient for the work presented. For this reason, it was decided to limit the size of the data set.

Comparison of these implementations on sample images allowed to conclude that the Fine Reader application gives much better results. Due to the fact that the program is intended for processing scanned documents, it worked best with images that were free from noise, shadows, distortions, and the text was clear and legible in them. It is worth noting that it did not require a thinned text – and even gave worse results after thinning. This clearly indicates that when using this program this stage should be omitted.

Another conclusion from the analysis of results is that more difficult cases of receipt images may require a dedicated solution, specially prepared for this type of image, and the image processing itself may be insufficient.

### 4.1 Comparing OCR results before and after pre-processing

The following results were obtained using the ABBYY Fine Reader OCR application. One representative of each of 9 the classes mentioned earlier (section 2) was selected

[1] http://tess4j.sourceforge.net/

[2] https://www.abbyy.com/en-ee/finereader/

randomly for study. The effectiveness of text recognition in the pictures before and after the pre-processing was compared.

Table 1 presents the results obtained. The correctly identified characters and words were calculated for individual samples. The results may seem low, as for the OCR system, but it is worth paying attention to several reasons. First of all, the selection of the samples – many of them were difficult and problematic. In addition, the method of assessing the correct diagnosis was quite rigorous and refused a character or word in the event of the slightest deviation. In the final solution, one should use the semantic analysis to increase the level of identification by introducing a dictionary or knowledge about what characters or words we expect in a given part of the receipt. Here it was not implemented. Many errors most probably resulted from the use of specific typefaces in part of cash registers, which apparently were not in the ABBYY Fine Reader data base, because some characters could not be identified in any instance. Here, it would certainly help to teach a system of specific fonts, used by the most popular cash registers in a given region. At the same time, these results are similar to the results reported by other authors (as described in section 1).

**Table 1.** A comparison of the effectiveness of character recognition in images before and after processing

| | average | minimum | maximum |
|---|---|---|---|
| Words recognized in original images | **24%** | 1% | 56% |
| Words recognized in pre-processed images | **33%** | 1% | 62% |
| Characters recognized in original images | **41%** | 1% | 76% |
| Characters recognized in pre-processed images | **51%** | 3% | 82% |

Careful analysis of the results for individual samples also revealed that for some of the examined photos the level of recognition was extremely low – around 3%. These were photos that were heavily corrupted (e.g. by uneven shade) and even the use of a processing algorithm had little effect on the result. These cases have been identified and will be analyzed more closely in subsequent work.

The results presented were made on a small data set, which is why they can not be considered as proving the effectiveness of the discussed techniques. For a proper assessment, it should be repeated on a full data set. However, it can be concluded that the approach is promising and it is worth continuing research. Text on pre-processed images has been recognized with a higher efficiency than on those before processing. The text recognition efficiency has been improved by an average of 25% in the case of characters and over 35% considering the entire words. Importantly – in each of the cases studied, the result improved by using the proposed processing algorithm. These results are comparable to the results of other authors (as mentioned in section 1).

## 5. Conclusions

After analyzing various pre-processing methods and examining the results obtained by OCR applications on the collected examples database, the following algorithms have been proposed:

- Straighten the image by locating the angle of rotation using the Hough transformation
- Finding a text outline using the described algorithm using the Canny method
- Image binarization by adaptive method

Its application allowed to obtain optimal and reproducible results on the tested sample. The vast majority of samples, regardless of the lighting conditions, cropping, angle of rotation or size, has been processed in a satisfactory manner. The output images are similar, comparable and reproducible. Each is aligned, the text takes up most of the frame and noise has been reduced.

Research using the OCR system was performed on a small database of samples and should be treated as preliminary. Nevertheless, they are promising and encouraging further research – character recognition efficiency has been improved by 25%. In the future, the authors plan to apply algorithms on a larger data set, in order to validate the approach. The direction of further work will include also the identified problematic receipt classes and try to address these problems. The solution may be to improve certain stages of the process, eg it is planned to use the binarization method proposed by Sauvola [11], intended to deal with noise, uneven lighting and other types of degradation of image quality. In addition, it is planned to use own implementation of the module recognizing the text, based on the authors' experience in the field of word recognition [15].

## References

[1] Wojciech Bieniecki, Szymon Grabowski, and Wojciech Rozenberg. Image preprocessing for improving ocr accuracy. In *2007 International Conference on Perspective Technologies and Methods in MEMS Design*, pages 75–80, May 2007.

[2] Matteo Brisinello, Ratko Grbić, Matija Pul, and Tihomir Anđelić. Improving optical character recognition performance for low quality images. In *2017 International Symposium ELMAR*, pages 167–171, Sep. 2017.

[3] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.

[4] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[5] Wojciech Korobacz. Przetwarzanie wstępne zdjęć paragonów do celów rozpoznawania: praca magisterska. Master's thesis, Wydział Informatyki PB, 2017. (In Polish).

[6] Witold Malina and Maciej Smiatacz. *Cyfrowe przetwarzanie obrazów*. Akademicka Oficyna Wydawnicza EXIT, 2008. (In Polish).

[7] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[8] William K. Pratt. *Digital Image Processing*. Addison-Wesley Publishing Company, 1991.

[9] John C. Russ and F. Brent Neal. *The Image Processing Handbook*. CRC Press, Inc., 2016.

[10] Khalid Saeed, Marek Tabedzki, Mariusz Rybnik, and Marcin Adamski. K3M: A universal algorithm for image skeletonization and a review of thinning techniques. *Applied Mathematics and Computer Science*, 20(2):317–335, 2010.

[11] Jaakko Sauvola and Matti Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[12] Linda G. Shapiro and George C Stockman. *Computer Vision*. Prentice Hall, 2001.

[13] Pooja Sharma and Shanu Sharma. Image processing based degraded camera captured document enhancement for improved ocr accuracy. In *2016 6th International Conference – Cloud System and Big Data Engineering (Confluence)*, pages 441–444, Jan 2016.

[14] Mande Shen and Hansheng Lei. Improving ocr performance with background image elimination. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 1566–1570, Aug 2015.

[15] Marek Tabedzki, Mariusz Rybnik, and Khalid Saeed. New results for view-based feature extraction method for handwritten words recognition without segmentation. In *1st International Conference on Image Processing & Communications, Poland*, pages 193–200, 2009.

[16] Marek Tabedzki, Khalid Saeed, and Adam Szczepanski. A modified K3M thinning algorithm. *Applied Mathematics and Computer Science*, 26(2):439–450, 2016.

[17] Øivind Due Trier and Anil K. Jain. Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1191–1201, Dec 1995.

[18] Christopher Wiraatmaja, Kartika Gunadi, and Iwan Njoto Sandjaja. The application of deep convolutional denoising autoencoder for optical character recognition preprocessing. In *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, pages 72–77, Sep. 2017.

# PRZETWARZANIE WSTĘPNE ZDJĘĆ PARAGONÓW DO CELÓW ROZPOZNAWANIA

**Streszczenie** Tematem tej pracy są metody przetwarzania wstępnego obrazów, zastosowane do zdjęć przedstawiajacych paragony. Celem jest poprawa ich jakości, pozwalająca zwiększyć skuteczność działania oprogramowania do rozpoznawania tekstu. Autorzy mieli na uwadze głównie trudne przypadki – zdjęć robionych „z ręki", przy słabym oświetleniu. Praca opisuje przeanalizowane metody filtrowania, binaryzacji, wyszukiwania krawędzi, prostowania obrazu, oznaczania obszaru zainteresowania, ścieniania. Przedstawiono również wstępne wyniki testów z oprogramowaniem OCR na niewielkiej bazie obrazów. Przetwarzanie wstępne pozwoliło na poprawę identyfikacji znaków o 25%. W końcowej części przedstawiono wnioski oraz plany przyszłej pracy.

**Słowa kluczowe:** cyfrowe przetwarzanie obrazów, rozpoznawanie znaków

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

# SOME IDEAS ABOUT CONNECTED GRAPHS ISOMORPHISM

Larisa Marchenko[1], Viktoria Podgornaya[2]

[1] Faculty of Mathematics and Programming Technologies, Francisk Scorina Gomel State University, Gomel, Belarus

[2] V.A.Belyi Metal-polymer Research Institute of National Academy of Sciences of Belarus, Gomel, Belarus

**Abstract:** In the paper we investigate the existence of graphs isomorphism and the search for invariants of connected graphs. A new graph invariant is formulated. It can be used to detect isomorphism of connected graphs. The vector space of all simple cycles of the graph and their edge-disjoint unions (cycle space) and the vector space of all cutting sets of the graph and their edge-disjoint unions (cut space) are constructed in the article for finding a new graph invariant. The authors investigate the method of constructing these vector spaces: cycle space and cut space. A new estimate of the dimensions of these vector spaces of the graph is given. The obtained invariant is demonstrated on a concrete example. A counterexample is constructed to confirm the fact that the proposed invariant can be used as a necessary but not sufficient condition for graphs isomorphism. A heuristic algorithm is proposed for constructing a one-to-one correspondence between sets of vertices of isomorphic graphs.

**Keywords:** graph, cycle, cutting set, vector space, invariant, isomorphism algorithm

## 1. Introduction

Isomorphism of graphs is the equivalence relation on the set of all graphs of the same order. Detection of graphs isomorphism is required in various fields of theoretical and applied knowledge [5]. A number of scientific studies of recent decades [1–4] are devoted to the problem of identifying isomorphic graphs as a class of equivalent objects, but the question still remains unresolved.

Currently, methods for detecting isomorphism for some types of graphs are known [6, 7]. For almost every particular algorithmic problem of detecting isomorphism of a special kind graphs, it was possible either to construct a polynomial algorithm or to prove its belonging to the class of NP-complete problems [5]. However,

Advances in Computer Science Research, vol. 14, pp. 105-123, 2018. DOI: 10.24427/acsr-2018-vol14-0007

detecting isomorphism of arbitrary graphs is a more complex task, which is still not finally resolved. One approach to identifying isomorphism of arbitrary graphs is the application of invariants.

In isomorphic graphs, by definition, all characteristics and properties are the same. A characteristic of a graph is called an invariant if it does not change with an isomorphic transformation of the graph. In many papers, an invariant is called complete if its coincidence for different graphs guarantees the existence of their isomorphism. You can require that the invariant does not change when you renumber the vertices of the graph. However, the latter requirement, in our opinion, is not essential, since it does not correspond to the notion of automorphism of the graph. Note that the graph automorphism is a special case of graphs isomorphism in a broad sense. On the other hand, one of the forms of graph representation is adjacency matrix, which allows to describe fully a given graph in a compact numerical form and depends on the numbering of vertices in it. It is obvious that the complete invariant of the graph remains unchanged under any numbering of the vertices and edges of the graph.

The main easily computable graph invariants are: the number of vertices and edges of the graph, the vector of local degrees of vertices, the number of connected components. Also, the number of vertices of the largest complete subgraph (density), the largest number of pairwise non-adjacent vertices of the graph (non-density), the chromatic number and the chromatic index of the graph, the Hadwigers number and others were used as invariants by a number of authors [8]. All the characteristics listed here are calculated from the original graph, but their values do not allow to restore the graph structure. Therefore, the requirement of equality of these characteristics is a necessary condition for graphs isomorphism, since counterexamples are known. That is, in the case of identical invariants, graphs may not be isomorphic [11]. For some classes of graphs, only a collection of several numerical characteristics helps to identify isomorphism. There are also a number of papers devoted to the detection of graph isomorphism using eigenvalues and the vectors of the adjacency matrix, the graph spectrum [11].

In the problem of detecting isomorphism, we have two main subtasks. First, the problem of identifying the isomorphism of two graphs without specifying the most bijective mapping between the sets of their vertices. Secondly, establishing a one-to-one correspondence between the sets of vertices of graphs that preserve the adjacency of the corresponding vertices and all other characteristic properties, i.e. construction of a map that is an automorphism on the set of vertices of the graph.

When detecting graphs isomorphism, the most important thing is to search for criteria. As noted above, the equality of known invariants gives only the necessary signs of isomorphism. The only currently known sufficient condition for graphs isomorphism is the equality of adjacency matrices converted to the same type by permutations of rows and corresponding columns. This transformation in the general case, as we know, requires of order n! of steps'.

We note that to establish a bijective map, the Lux approach [9] is considered to be the most promising, which reduces the number of permutations using block structures. Based on this approach, Leslie Babay proposes a heuristic method for constructing graphs isomorphism [10].

Thus, the search for a sufficient condition for graphs isomorphism, which can be computed in polynomial time or at least almost polynomial, is relevant. Similarly, with respect to establishing a bijective map of vertex sets of isomorphic graphs. In this article new necessary condition for detection of graphs isomorphism are constructed.

## 2. Preliminary information

We present the necessary definitions and properties.

*Definition.* An abstract graph (graph) $G = (V,E)$ is a pair, which consist of vertices set $V = \{v\}$ and edges set $E = \{e = (u,v) | u,v \in V\}$. The graph $G$ has an order $n$ if $|V| = n$.

A pair of vertices can be connected by two or more edges, such edges are called multiples. An edge can begin and end at the same vertex, that is, $e = (v,v) \in E$. In this case, the edge is called a loop. A graph is called *simple* if it does not contain multiple edges and loops.

*Definition.* Graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ are called isomorphic if there is a one-to-one correspondence $h : V_1 \to V_2$, preserving the adjacency of the corresponding vertices:

$$e_1 = (u,v) \in E_1 \Leftrightarrow e_2 = (h(u),h(v)) \in E_2.$$

Isomorphic graphs are naturally identified. And they can be represented in the same way. If the graphs $G$ and $H$ are isomorphic, then write $G \cong H$ (so $H \cong G$).

An isomorphic map of a graph onto itself is called an automorphism.

*Definition.* Let $f$ be a function that associates with each graph $G$ an element $f(G)$ from some set of $M$. The function $f$ is called an invariant if its values on isomorphic graphs coincide:

$$G \cong H \Rightarrow f(G) = f(H)$$

for any graphs $G$ and $H$.

The invariant $f$ is called complete if, for any graphs $G$ and $H$ the equality of $f(G)=f(H)$ implies an isomorphism of graphs $G$ and $H$.

As "conditional" full invariants, many authors consider the adjacency, incidence and Kirchhoffs matrices [12], since after transformation to the same form, they allow us to construct the required one-to-one correspondence between sets of graph vertices. The following theorem is known.

**Theorem 1** [13]**.** Graphs are isomorphic if and only if their adjacency matrices (incidents, Kirchhoff) are permutationally similar, that is, they can be obtained one from the other by permutations of the rows and the corresponding columns.

Moreover, a uniform computational algorithm has not yet been formulated for an arbitrary graph without enumerating the elements of the set of vertices that has complexity of order less than $n!$.

For the considered vectors and matrices, addition and multiplication operations act in the Galois field $GF(2)$ modulo 2, unless otherwise specified.

*The route* in the graph $G=(V,E)$ is the finite sequence of its edges of the form $(v_0,v_1)$, $(v_1,v_2)$,..., $(v_{k-1},v_k)$. The number of edges in the route (with repetitions) is called *the length of the route*.

At the same time we have:

1) if $v_0=v_k$, then the route is called *closed*, otherwise *open*;

2) if all edges of the route are different, then the route is called *a chain* and is denoted by $[v_0;v_k]$, if all vertices are different — *a simple chain*;

3) a closed chain $(v_k=v_0)$ is called *a cycle*, a closed simple chain is *a simple cycl*e.

The graph $G'=(V',E')$ is called the subgraph of the graph $G=(V,E)$, if $V'\subseteq V$ and $E'\subseteq E$. If $V'=V$, then the graph $G'$ is called *the spanning subgraph* of the graph $G$.

*The spanning tree* of the graph $G=(V,E)$, $|V|=n$, is called a spanning subgraph without cycles.

The proof of Theorem 2.2 [12] implies that if a graph is connected, then it has at least one spanning tree. The converse is also true (Theorem 2.3, [12]).

Vertices $u$ and $v$ are called *reachable* if there is a route from $u$ to $v$. A graph in which any two vertices are reachable is called *connected*. Any vertex $v$ is connected with itself by a trivial route.

A connected subgraph $G_1$ of a graph $G$ is called *a connected component* of a graph $G$ if $G_1$ is maximal in the sense that no other connected subgraph $G_2$ of the graph $G$ contains the subgraph $G_1$. The number of connected components of a graph is denoted by $k(G)$. For a connected graph, we have $k(G)=1$.

The cutting set or cutset $S$ of a connected graph $G$ is the minimal set of edges whose removal makes the graph $G$ disconnected.

Let $G$ be a connected graph and $V = \{V_1, V_2\}$ be a partition of the set of its vertices: $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. The set of edges of the graph $G$, one end of which belongs to $V_1$, and the other to $V_2$, is called a cut. The operation to delete the edges of the cut divides the graph into two connected components and makes it disconnected.

*Definition*[12]. Let $T$ be an arbitrary spanning tree, $T^* = G \setminus T$ be the corresponding counting spanning tree of the graph $G = (V, E)$, $|V| = n$, $|E| = m$. And let $e_i^*$ be the chord (edge) of counting spanning tree $T^*$. Since $T$ is an acyclic graph, the graph $T \cup e_i^*$ contains exactly one cycle $C_i$. The cycle $C_i$ consists of the chord $e_i^*$ and those edges of the spanning tree $T$, which form a single simple chain between the terminal vertices of the chord $e_i^*$. The cycle $C_i$ is called *a basic cycle* with respect to the chord $e_i^*$ and the spanning tree $T$. The number of all basic cycles in an arbitrary graph is equal to the cyclomatic number of the graph $\nu(G) = m - n + k$, where $k$ is the number of connected components.

The set of all basic cycles is called *the fundamental system of cycles* or *cycle basis* relative to the fixed spanning tree of $T$. The fundamental system of cycles is associated with a specific spanning tree. The number of spanning trees is equal to the algebraic complement of any element of the Kirchhoff's matrix. If we take another spanning tree, then it will correspond to a different set of cycles that form the *cycle basis*.

Removing the edge $e_j$ from the spanning tree $T$ breaks it into 2 components of connectivity $T_1$ and $T_2$. Let $V_1$ and $V_2$ be the sets of vertices of the components $T_1$ and $T_2$, respectively. And $G_1$ and $G_2$ be the subgraphs of the graph $G$, which are generated by the sets of the vertices $V_1$ and $V_2$. Obviously, $T_1$ is the spanning tree of the subgraph $G_1$, and $T_2$ is the spanning tree of the subgraph $G_2$. Consequently, the subgraphs $G_1$ and $G_2$ are connected. The separating cut $V_1$ and $V_2$ is the cutting set of the graph $G$. The cutting set $S_j$ made up of edges connecting the vertices of the components $T_1$ and $T_2$ of the spanning tree is called *the base cut* of $G$.

The set of all basic cuts is called *the fundamental system of cuts* or *the fundamental cutsets* of the graph $G$ with respect to the spanning tree $T$. The number of the basic cuts in an arbitrary graph is equal to *the rank* of the graph $\nu^*(G) = n - k$.

Properties of basic cycles and basic cuts:

1. The base cycle $C_i$ with respect to the chord $e_i^*$ of the $T^*$ of the connected graph $G$ includes exactly those edges of the spanning tree $T$, which correspond to the basic cuts which include this chord.

2. The base cut $S_j$ with respect to the edge $e_j$ of the spanning tree $T$ of the connected graph $G$ includes exactly those chords of the counting spanning trees $T^*$, which correspond to the basic cycles which are including this edge.

Let $(e_1, e_2, \ldots, e_m)$ be a sequence of all edges of the graph $G = (V, E)$, $|V| = n$, $|E| = m$.

The base cycle $C_i$, $i = 1, \ldots, \nu$, determines the vector $(c_{i1}, c_{i2}, \ldots, c_{im})$, where $c_{ij} = 1$, if $e_j \in C_i$, and $c_{ij} = 0$, if $e_j \notin C_i$. The fundamental system of cycles corresponds to *the matrix of cycles* $C(G) = [c_{ij}]$, $i = 1, \ldots, \nu$, $j = 1, \ldots, m$. Since each basic cycle $C_i$ contains exactly one chord, the matrix $C(G)$ can be transformed to canonical form by rearranging the columns

$$\hat{C}(G)_{\nu\times m} \sim \begin{pmatrix} 1 & 0 & \ldots & 0 & a_{1\nu+1} & \ldots & a_{1m-\nu} \\ 0 & 1 & \ldots & 0 & a_{2\nu+1} & \ldots & a_{2m-\nu} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & a_{\nu\nu+1} & \ldots & a_{m-\nu} \end{pmatrix} = [\mathbf{E}_\nu \mid \mathbf{C}^*], \tag{1}$$

where $\mathbf{C}^* = \begin{pmatrix} a_{1\nu+1} & \cdots & a_{1m-\nu} \\ a_{2\nu+1} & \cdots & a_{2m-\nu} \\ \cdots & \cdots & \cdots \\ a_{\nu\nu+1} & \cdots & a_{\nu\, m-\nu} \end{pmatrix}$, $a_{ij} \in \{0, 1\}$, for $i = 1, \ldots, \nu$, $j = 1, \ldots, m-\nu$.

In contrast to [12], here we construct the cycle matrix for the graph, and not for the digraph.

In the matrix $C(G)_{\nu\times m}$, the columns of the unit submatrix $\mathbf{E}_\nu$ correspond to the edges of the counting spanning tree $T^*$, followed by the columns corresponding to the edges of the spanning tree $T$. Here we note that the cycle matrix does not define the entire graph up to isomorphism. For example, vertices of a graph of degree 1 will not be present in it.

Similarly to the basic cut $S_i$, for $i = 1, \ldots, v^*$, there is a vector $(s_{i1}, s_{i2}, \ldots, s_{im})$, where $s_{ij} = 1$, if the edge is $e_j \in S_i$, and $s_{ij} = 0$, if $e_j \notin S_i$. For the fundamental system of cuts, we can write the *matrix of cuts* $S(G) = [s_{ij}]$, where $i = 1, \ldots, \nu^*$, $j = 1, \ldots, m$, which, by rearranging the columns, also converts to canonical form

$$\hat{S}(G)_{\nu^*} \sim \begin{pmatrix} b_{11} & \ldots & b_{1\nu} & 1 & 0 & \ldots & 0 \\ b_{21} & \ldots & b_{2\nu} & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{\nu^*1} & \ldots & b_{\nu^*\nu} & 0 & 0 & \ldots & 1 \end{pmatrix} = [\mathbf{S}^* | \mathbf{E}_{v^*}], \tag{2}$$

where $S^* = \begin{pmatrix} b_{11} & \dots & b_{1\nu} \\ b_{21} & \dots & b_{2\nu} \\ \dots & \dots & \dots \\ b_{\nu^*1} & \dots & b_{\nu^*\nu} \end{pmatrix}$, $b_{ij} \in \{0,1\}$, for $i = 1, \dots, \nu^*$, $j = 1, \dots, \nu$.

Similarly, here we build a matrix of cuts for a graph, and not for a digraph. Note that the matrix of cuts is also not a unique representation of the graph.

The rows of the matrix of cycles are called *the cycle vectors* of the graph $G$, the rows of the matrix of cuts are called *the vectors of the cuts*.

In [16] algorithms of construction of basic cycles of minimum length are given.

## 3. Main part

We will consider abstract connected graphs with number of vertices $n \geq 3$.

For the graph $G = (V,E)$, the Boolean (the set of all subsets) of the set $E$, including the empty set $\emptyset$, is denoted by $W_G$. The set $W_G$ forms an Abelian group with operation of addition modulo 2, provided that all elements are represented as rows of 0 and 1 length $|E| = m$ by the following rule. If the edge $e_i$ belongs to a subset, then the $i$-th coordinate is 1, if it does not belong to — 0. Addition is performed by coordinate. Multiplying by 0 gives the zero line corresponding to the empty set. If we add the operation of multiplying rows by elements of the Galois field $GF(2) = \{0,1\}$, then all the axioms of the linear space for the set $W_G$ will be fulfilled. In this case, the dimension of this space is equal to $|E| = m$, and as one of the bases we can take the rows corresponding to the edges of the graph (Theorem 4.2, [12]).

The set of all simple cycles, including the null-graph and the union of edge-disjoint simple cycles of the graph over the field $GF(2)$, forms a linear subspace $W_C$ of dimension $\nu = m - n + k$ of the space $W_G$ (Theorem 4.3, [12]). We will call it the cycle space. Similarly, the set of all cuts corresponding to the selected spanning tree of the graph and their edge-disjoint unions of cuts over the field $GF(2)$ is a linear subspace $W_S$ of dimension $\nu^* = n - k$ of the space $W_G$ (Theorems 4.4, 4.5, [12]). We will call it the cut space. Moreover, the set of basic cycles and the set of basic cuts with respect to some spanning tree of a connected graph are bases, respectively, of the space of all simple cycles of the graph and their edge-disjoint unions, as well as the space of all cuts and their edge-disjoint unions, respectively. They are called cycle basis and fundamental cutsets accordingly (or cut basis).

Linear subspaces of graph: the cycle space $W_C$ and the cut space $W_S$ are orthogonal, moreover, they are orthogonal complements of each other (theorems 4.9 and 4.10, [12]). And any graph can be represented as a direct sum of the cycle space and the cut space (theorem 4.11, [12]).

According to the process of construction of basic cycles, it can be seen that only simple cycles can be basic. Since each of them will contain only one chord (theorem 2.10, [12]) and edges of the graph spanning tree. Moreover, each chord will be included only in one basic cycle. Therefore, the fundamental system of cycles (cycle basis) is a linearly independent system of vectors. Similarly, each cut from the fundamental cutsets of graph contains only one edge of the spanning tree (theorem 2.9, [12]). The specified edge belongs to a single cut from the fundamental cutset of graph, which ensures their linear independence.

The elements of the linear subspaces $W_C$ and $W_S$ are found as linear combinations of the vectors of these linearly independent systems of vectors, therefore the cycle basis and the fundamental cutsets form the bases of the corresponding vector subspaces (Theorem 4.6, [12]).

**Theorem 2** [14]. For a simple graph $G$, any row of the matrix $C = C(G)_{\nu \times m}$ is orthogonal to any row of the matrix $S = S(G)_{\nu^* \times m}$

$$C \cdot S^T = S \cdot C^T = \mathbf{0},$$

where $C^T, S^T$ are transposed matrices, $\mathbf{0}$ is the zero matrix of the corresponding dimension.

A similar theorem for a digraph was proved in [12] by Theorem 6.6.

**Theorem 3** [14]**.** Let for some spanning tree T of a connected graph $G = (V, E)$, $|V| = n$, $|E| = m$, $k = 1$, $n \geq 3$, the matrix of cycles $C(G)$ is constructed and transformed to the form $C(G) = [\mathbf{E}_{m-n+k} \mid \mathbf{C}^* ]$. Then the canonical form of the matrix of cuts can be defined as $S(G) = [\mathbf{S}^* | \mathbf{E}_{n-k}]$, where $\mathbf{S}^* = (\mathbf{C}^*)^T$.

**Theorem 4** [15]**.** 1. The order of the linear subspace $W_C$ of all simple cycles of the graph, including the null-graph, and their edge-disjoint unions (cycle space), is $2^{m - n + k}$.

2. The order of the linear subspace $W_S$ (cut space) of all cutsets of the graph and their edge-disjoint unions (cut space) is $2^{n-k}$.

**Corollary 1.** The number of all non-zero simple cycles and their edge-disjoint graph associations is $2^{m-n+k} - 1$. The number of all non-zero cutsets of a graph and their edge-non-intersection associations is $2^{n-k} - 1$.

Thus, using the matrix of cycles $C(G)$, we construct the matrix of fundamental cutsets of the graph $S(G)$. Using the matrices $C(G)$ and $S(G)$, we can construct the cycle space $W_C$ of all simple cycles and their edge-disjoint unions and the cut space $W_S$ of all cuts and their edge-disjoint unions of the graph $G$ as the set of all possible linear combinations of rows of matrices by adding them modulo 2 over the Galois field $GF(2)$.

**Theorem 5.** The isomorphic graphs are the same:

1) ordered numerical sequences of lengths of all simple cycles and their edge-disjoint unions,

2) ordered numerical sequences of lengths of all cutsets and their edge-disjoint unions.

The proposed numerical sequences can be used as two invariants upon detection of graphs isomorphism. Using only the sequence of lengths of all simple cycles and their edge-disjoint unions does not work on graphs with vertices of degree 1.

Example 1. We consider the use of the proposed invariants on a known pair of non-isomorphic graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ (figure 1).

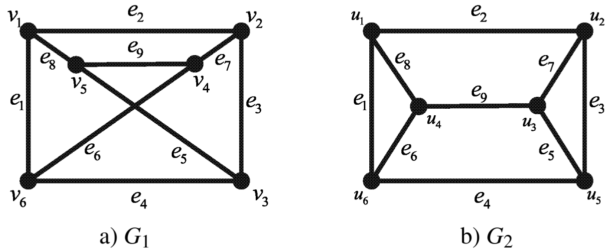


**Fig. 1.** Non-isomorphic graphs $G_1$ and $G_2$

For graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ we have $|V_1| = |V_2| = 6$, $|E_1| = |E_2| = 9$, $k_1 = k_2 = 1$. Note that the graphs $G_1$ and $G_2$ are regular of degree 3, that is, the vectors of the vertices powers of the graphs are $(3; 3; 3; 3; 3; 3)$. These invariants do not give an answer to the question about the isomorphism of these graphs.

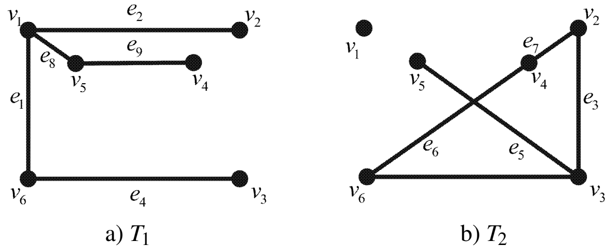


**Fig. 2.** The spanning tree $T_1$ and the spanning co-tree $T_1^*$ for the graph $G_1$

We construct the cycle space $W_C(G_1)$ for the graph $G_1$. The cyclomatic rank of the graph $G_1$ is $\nu(G_1) = 9 - 6 + 1 = 4$. Consequently, any spanning tree of a graph contains 5 edges, and spanning co-tree — 4 chords (edges). Take an arbitrary spanning tree of the graph $T_1$ (figure 2, a) and appropriate spanning co-tree $T_1^*$ (figure 2, b). Note that the spanning tree $T_1$ is presented in the form of a connected tree, and the spanning co-tree of $T_1^*$ is forest.

Attaching the chord (edge) $e_3$ of the spanning co-tree $T_1^*$ to the spanning tree $T_1$, we obtain the cycle $C_1 = \{e_1,\ e_2,\ e_3,\ e_4\} = (111100000)$, whose length is 4, $l(C_1) = 4$. Similarly for the chord $e_5$ cycle is $C_2 = \{e_5,\ e_4,\ e_1, e_8\} = (100110010)$, $l(C_2) = 4$, for the chord $e_6$ cycle is $C_3 = \{e_6,\ e_1,\ e_8,\ e_9\} = (100001011)$, $l(C_3) = 4$, for the chord $e_7$ cycle is $C_4 = \{e_2,\ e_7,\ e_9,\ e_8\} = (010000111)$, $l(C_4) = 4$.

The constructed basic cycles relative to the spanning tree $T_1$ form a fundamental system of cycles (cycle basis). Rewrite them in the form of vectors that form the rows of the matrix of cycles:

$$C(G_1)_{4\times 9} = \begin{array}{c} \begin{array}{ccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}.$$

Transform the matrix $C(G_1)_{4\times 9}$ to the canonical form, first writing down the columns corresponding to the spanning co-tree chords $T_1^*$:

$$\hat{C}(G_1)_{4\times 9} = \begin{array}{c} \begin{array}{ccccccccc} e_3 & e_5 & e_6 & e_7 & e_1 & e_2 & e_4 & e_8 & e_9 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array} = [\mathbf{E}_4|\mathbf{C}^*].$$

According to Theorem 3, the matrix of basic cuts (fundamental cutsets) in the canonical form is equal to

$$S(G_1)_{5\times 9} = \begin{array}{c} \begin{array}{ccccccccc} e_3 & e_5 & e_6 & e_7 & e_1 & e_2 & e_4 & e_8 & e_9 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} = [(\mathbf{C}^*)^T|\mathbf{E}_5].$$

The dimension of the space $W_C(G_1)$ of simple cycles and their edge-disjoint unions for the $G_1$ graph is $2^4$, and the elements of the space are represented by vectors obtained from the matrix $C(G_1)_{4\times 9}$ by modulo-2 addition of all possible combinations of the matrix rows:

$C_5 = (011010010), C_6 = (011101011), C_7 = (101100111),$
$C_8 = (000111001), C_9 = (110110101), C_{10} = (110011100),$
$C_{11} = (111011001), C_{12} = (001010101), C_{13} = (010111110),$
$C_{14} = (001101100), C_{15} = (101011110).$

In this case, we take into account the trivial zero cycle $C_{16} = (000000000)$.

The ordered sequence of lengths of all constructed simple cycles and their edge disjoint unions of the space $W_C(G_1)$ has the form:

$$(0,4,4,4,4,4,4,4,4,6,6,6,6,6,6,6).$$

Similarly, for the graph $G_2$ the matrix of cycles $C(G_2)$ and the matrix of cuts $S(G_2)$ in the canonical form are constructed on the spanning tree $T_2$ and on the spanning co-tree $T_2^*$ (Figure 3):

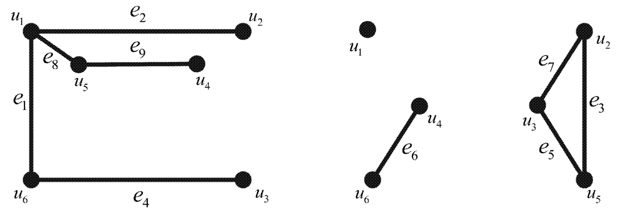
$$\hat{C}(G_2) = \begin{array}{c} \begin{matrix} e_3 & e_5 & e_6 & e_7 & e_1 & e_2 & e_4 & e_8 & e_9 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array},$$

$$\hat{S}(G_2) = \begin{array}{c} \begin{matrix} e_3 & e_5 & e_6 & e_7 & e_1 & e_2 & e_4 & e_8 & e_9 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

The ordered sequence of lengths of all constructed simple cycles and their edge disjoint unions of the cycle space $W_C(G_2)$ has the form:

$$(0,3,3,4,4,4,4,5,5,5,5,6,6,6,6,6).$$

At this step, it can be noted that the ordered sequences of the lengths of all simple cycles and their edge-disjoint unions of the cycle spaces $W_C(G_1)$ and $W_C(G_2)$ do not coincide. Therefore, there is no need to find the space of all cuts of the graph. And on the basis of the necessary condition (Theorem 5), these graphs are not isomorphic.



a) $T_2$

b) $T_2^*$

**Fig. 3.** The spanning tree $T_2$ and the spanning co-tree $T_2^*$ for the graph $G_2$

Thus, to detect the isomorphism of graphs, one should construct a matrix of cycles of the graph along one of the spanning trees of the graph. With its help, you can get in the graph the whole set of simple cycles and their edge-disjoint unions. To do this, you need to find all possible sums of rows of the graph matrix of cycles according to the definition of the basis of the linear space of $W_C$. Such nonzero sums will be $2^{\nu-1}$, where $\nu = m - n + k$ is the cyclomatic rank of the graph. Of course, the representations of these cycles will depend on the numbering of the vertices of the graph and the order in which they are considered. But we can first determine the lengths of all simple cycles in a graph and their edge-disjoint unions. Of course, for isomorphic graphs, ordered by non-decreasing sequences of the lengths of all simple cycles and their edge-disjoint unions must coincide completely. The discrepancy allows you to immediately answer the absence of isomorphism between the graphs.

Naturally, all vertices of degree 1 will not participate in the recording of cycles of a graph, since the hanging vertices of a connected graph do not enter into one cycle and will not be represented by units in the matrix of cycles. Therefore, it is also advisable here to consider the matrix of basic cuts, since the hanging vertices will necessarily fall into the set of all cutsets of the graph and their edge-disjoint unions. Since all such cutsets and their unions form a linear space of $W_S$, then using the matrix of basic cuts in $2^{n-k} - 1$ steps (summation modulo 2 of all possible variants of rows of the matrix $S$), all cutsets of the graph and their edge-disjoint unions can be obtained.

Like the case with cycles, isomorphic graphs organized by non-decreasing sequences of all cutsets and their edge-disjoint unions must coincide completely. The discrepancy allows you to immediately answer the absence of isomorphism between the graphs.

Note that this invariant, like many well-known invariants, does not allow us to establish the absence of isomorphism for the graphs $G_3$ and $G_4$ in Figure 4.

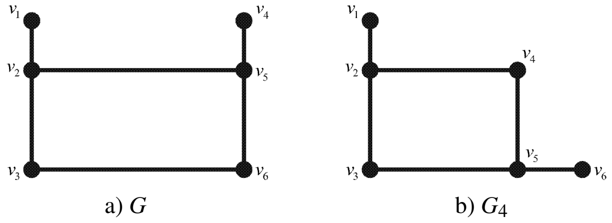


**Fig. 4.** Non-isomorphic graphs $G_3$ and $G_4$

The graphs $G_3$ and $G_4$ have the same number of vertices, edges, connected components, the same ordered power series of vertices, equal densities and non-densities, chromatic numbers, Hadwigers numbers. Also, both graphs have a single simple cycle of length 4 and 15 non-zero cutsets, the ordered sequences of lengths of which coincide. This counterexample suggests that in case of coincidence of ordered sequences of lengths of vectors of spaces $W_C$ and $W_S$, additional studies are required. Therefore, the coincidence of these new invariants is also only a necessary condition of graph isomorphism.

Thus, as invariants of a graph, it is recommended to consider the vector of degrees of the vertices of the graph, together with the ordered sequences of lengths of simple cycles and cutsets of the graph. However, with the full coincidence of these quantities, it is possible to make an assumption about the isomorphism of the graphs under consideration, but additional research will be required.

It is possible for graphs to establish an isomorphism using the following heuristic algorithm.

Let $G$ and $G^*$ are isomorphic graphs with different numbering of the vertices. It is necessary to construct vertex classification trees for establishing a one-to-one correspondence between the vertex sets of graphs $G$ and $G^*$. Construction of the classification tree of the graph $G = (V,E)$, $V = \{v_1,\ v_2,\ \ldots,\ v_n\}$, $|V| = n$, $|E| = m$, $k = 1$, consists of the following steps.

(1) construction of spaces $W_C$ of simple cycles (and their edge-disjoint unions) and $W_S$ of simple cuts (and their edge-disjoint unions) of graph $G$;

(2) establishing the incidence property of each vertex of the graph to some edges forming cycle of the space $W_C$ or a cut of the space $W_S$;

(3) construction of matrix $X = [x_{ij}]$ "vertex-property". The cycles and the cuts of the spaces $W_C$ and $W_S$ respectively are considered as properties forming the rows of the matrix. The element $x_{ij}$ of the matrix takes the value 1 if the vertex $v_j$ is incident

to some edge of the cycle or cut located in the $i$-th row, and 0 otherwise, where $i = 1, \dots, p$, $j = 1, \dots, n$, $p = 2^{m-n+k} + 2^{n-k} - 2$;

(4) calculation of the sum of elements in each column of the matrix $X$: $Q_j = \sum_{i=1}^{p} x_{ij}$, $j = 1, \dots, n$;

(5) sorting the columns of the matrix $X$ in non-decreasing order of vertex degrees $deg\ v_j$, $j = 1, \dots, n$. Columns corresponding to vertices with equal degrees, it is also desirable to sort in non-decreasing order of the sums of $Q_j$. Let's redefine the obtained sequence of vertices as $x_1,\ x_2, \dots, x_n$;

(6) construction of the similarity matrix $H = [h_{jk}]$ of vertices $x_1,\ x_2, \dots, x_n$, where the element $h_{jk}$ is the "distance" between vertices $x_j$ and $x_k$, calculated by the formula $h_{jk} = \sum_{i=1}^{p} |x_{ij} - x_{ik}|$ for $j,k = 1, \dots, n$;

(7) dividing the set of vertices using the similarity matrix, assuming that initially all vertices belong to the same class: $K_1 = \{x_1,\ x_2, \dots, x_n\}$. To do this, two vertices $x_l$ and $x_q$ are defined, the difference between them is the greatest, that is, $h_{jk} = max\{h_{ij}\}$. The resulting vertices are considered to be the centers of two new classes $K_l$ and $K_q$. Other vertices are divided into these classes by the degree of proximity to their centers. If the centers of the classes are the vertices $x_l$ and $x_q$ forming the classes $K_l$ and $K_q$ respectively, then the vertex $x_r \in K_l$ if $h_{rl} < h_{rq}$, or $x_r \in K_q$ otherwise. As a result, all vertices will be distributed between two classes $K_l$ and $K_q$. Further separation of vertices classes $K_l$ and $K_q$ takes place by a similar procedure. If at some step it turned out that there are several vertices with equal distances to the centers in the classes, then the degree of vertex deg $x_j$ and $Q_j$ should be taken into account when dividing them.

The split process ends when there is only one vertex in each class.

This procedure allows us to construct a classification tree $K^*$, whose hanging vertices correspond to the vertices $x_1,\ x_2, \dots, x_n$ and, consequently, to the vertices $v_1,\ v_2,\ \dots,\ v_n$ of the original graph $G$.

Similarly, we construct a classification tree $K^*$ of the graph $G^* = (U, E^*)$, $U = \{u_1,\ u_2,\ \dots,\ u_n\}$, $|U| = n$, $|E| = m$, $k = 1$, whose hanging vertices are $u_1,\ u_2,\ \dots,\ u_n$.

With the help of known algorithms for establishing isomorphism of trees, a one-to-one mapping between sets of graph vertices is constructed.

The work of the proposed algorithm is considered by an example.

Example 2. Establish a one-to-one correspondence between vertices of isomorphic graphs $G_5 = (V,\ E_5)$ and $G_6 = (U,\ E_6)$, $|V| = |U| = 5$, $|E_5| = |E_6| = 6$, $k(G_5) = k(G_6) = 1$ (figure 5) using vector spaces of simple cycles (and their edge-disjoint unions) and cuts (and their edge-disjoint unions) $W_C$ and $W_S$.

Choosing the spanning trees $T_5$ and $T_6$ of graphs $G_5$ and $G_6$ (figure 5), we obtain the following matrix of basis cycles

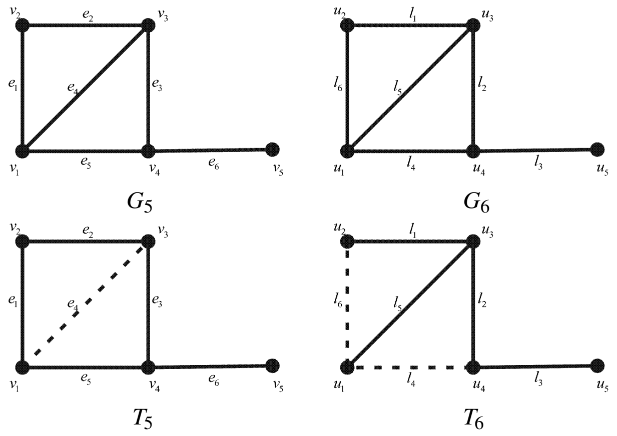


**Fig. 5.** Isomorphic graphs $G_5$ and $G_6$ and their spanning trees $T_5$ and $T_6$

$$C(G_5) = \begin{pmatrix} 1\,1\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0 \end{pmatrix},$$

$$C(G_6) = \begin{pmatrix} 0\,1\,0\,1\,1\,0 \\ 1\,0\,0\,0\,1\,1 \end{pmatrix}.$$

Next, we find the corresponding matrix of basic cuts

$$S(G_5) = \begin{pmatrix} 1\,1\,1\,0\,0\,0 \\ 1\,1\,0\,1\,0\,0 \\ 1\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1 \end{pmatrix},$$

$$S(G_6) = \begin{pmatrix} 0\,1\,1\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 1\,1\,0\,0\,0\,1 \end{pmatrix}.$$

Using matrices $C(G_5)$, $S(G_5)$ the corresponding spaces $W_C(G_5)$ and $W_S(G_5)$ are constructed. Let's form a matrix "vertex-property" $X(G_5)$ in accordance with point (3) of the algorithm

| X($G_5$) | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $C_1$ | 1 | 1 | 1 | 1 | 0 |
| $C_2$ | 1 | 1 | 1 | 0 | 0 |
| $C_3$ | 1 | 0 | 1 | 1 | 0 |
| $S_1$ | 1 | 1 | 1 | 1 | 0 |
| $S_2$ | 1 | 1 | 1 | 1 | 0 |
| $S_3$ | 1 | 0 | 1 | 1 | 0 |
| $S_4$ | 0 | 0 | 0 | 1 | 1 |
| $S_5$ | 1 | 1 | 1 | 0 | 0 |
| $S_6$ | 1 | 1 | 1 | 1 | 0 |
| $S_7$ | 1 | 1 | 1 | 1 | 1 |
| $S_8$ | 1 | 1 | 1 | 1 | 0 |
| $S_9$ | 1 | 1 | 1 | 1 | 1 |
| $S_{10}$ | 1 | 0 | 1 | 1 | 1 |
| $S_{11}$ | 1 | 1 | 1 | 1 | 0 |
| $S_{12}$ | 1 | 1 | 1 | 1 | 1 |
| $S_{13}$ | 1 | 1 | 1 | 1 | 1 |
| $S_{14}$ | 1 | 1 | 1 | 1 | 1 |
| $S_{15}$ | 1 | 1 | 1 | 1 | 1 |
| $Q_j$ | 17 | 14 | 17 | 16 | 8 |

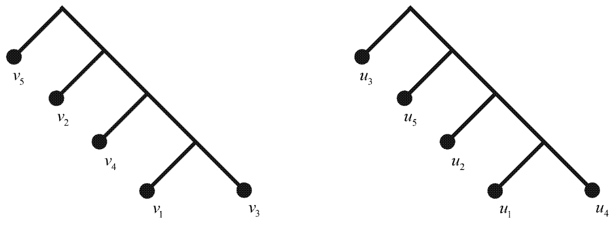
After ordering the vertices by non-decreasing degrees, we find the matrix of similarity of the vertices of the graph $G_5$

$$H(G_5) = \hat{S}(G_2) = \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 10 & 11 & 11 & 8 \\ 10 & 0 & 3 & 3 & 6 \\ 11 & 3 & 0 & 0 & 3 \\ 11 & 3 & 0 & 0 & 3 \\ 8 & 6 & 3 & 3 & 0 \end{pmatrix} \end{array},$$

where $x_1 \to v_5, x_2 \to v_2, x_3 \to v_1, x_4 \to v_3, x_5 \to x_4$.

Using the matrix $H(G_5)$, taking into account the values of $Q_j$, we construct a classification tree of vertices of the graph $G_5$ (figure 6, a). Similarly, we obtain the vertex classification tree of the graph $G_6$ (figure 6, b).

Comparing the classification trees $K(G_5)$ and $K(G_6)$ taking into account the automorphism of graph vertices, we obtain the following one-to-one mapping between the vertex sets of graphs $G_5$ and $G_6$ preserving adjacency: $v_1 \leftrightarrow u_1$, $v_2 \leftrightarrow u_5$, $v_3 \leftrightarrow u_4$, $v_4 \leftrightarrow u_1$, $v_5 \leftrightarrow u_3$.



a) $K(G_5)$ b) $K(G_6)$

**Fig. 6.** Classification tree of the vertices in isomorphic graphs $G_5$ and $G_6$

The developed new invariant can be used in the analysis of graph structures of big data. The proposed heuristic algorithm makes it possible to establish a bijection between sets of vertices of isomorphic graphs when solving applied problems.

## 4. Conclusions

The article describes the following new statements.

1. The linear spaces of all simple cycles and their edge disjoint unions of a simple graph (cycle space), as well as of all cuts of the graph and their edge disjoint unions (cut space) are constructed. A estimate of the dimensions of these spaces is given.

2. The canonical form of matrices of basis cycles and basis cuts of the graph is described. The method of finding these matrices is given.

3. A new invariant of detecting isomorphism of graphs in the form of ordered by non-decreasing sequences of lengths of all simple cycles of the graph (and their edge-disjoint unions) and all sections of the graph (and their edge-disjoint unions) is proposed. It is shown that the coincidence of this invariants gives a new necessary condition for isomorphism of graphs.

4. We propose a heuristic algorithm for constructing a one-to-one correspondence between sets of vertices of isomorphic graphs using the linear spaces of all simple cycles and their edge disjoint unions of a simple graph (cycle space), as well as of all cuts of the graph and their edge disjoint unions (cut space).

## References

[1] Нечепуренко М.И.: Алгоритмы и программы решения задач на графах и сетях Новосибирск, Наука, 1990.

[2] Карелин, В.П.: Теория и средства поддержки принятия решений в организационно-технологических системах: дис. . . . д-ра техн. наука, Таганрог, ТРТУ, 1995.

[3] Бернштейн Л.С., Карелин В.П., Целых А.Н.: Модели и методы принятия решений в интегрированных интеллектуальных системах, Ростов/Д, Изд-во РГУ, 1999.

[4] Пинчук В.П.: Табличные инварианты на графах и их применение // Кибернетика и системный анализ, Институт кибернетики НАН Украины, 2001, № 4.

[5] Карелин В.П.: Задача распознавания изоморфизма графов. Прикладное значение и подходы к решению, Вестник Таганрогского института управления и экономики, 2015, № 1, С. 102–106.

[6] Пономаренко И.Н.: Проблема изоморфизма графов: Алгоритмические аспекты, СПб., 2010, 57 с.

[7] Погребной Ан.В., Погребной В.К.: Метод дифференциации вершин графа и решение проблемы изоморфизма // Известия Томского политехнического университета. Инжиниринг георесурсов. – 2015. – Т. 326. – № 6. – С. 34–45.

[8] Батенков К.А. Числовые характеристики структур сетей связи. Труды СПИИРАН, 2017, Вып. 53, С. 5–28.

[9] L. Babai, Eugene M. Luks: Canonical labeling of graphs. Proc. 15th ACM Symp. on Theory of Computing, 1983, pp. 171–183.

[10] Leszly Babai: Graph Isomorphism in Quasipolynomial Time . Submitted 19 January, 2016. –V. 1 submitted 11 December, 2015; originally announced December 2015. [https://arxiv.org/search/cs?searchtype=author&query=Babai%2C+L].

[11] Ландо С.К.: Графы и топология. М., ВШЭ, 2018., 78 с.

[12] Свами М., Тхуласираман К.: Графы, сети и алгоритмы. М., Книга по Требованию, 2013, 450 с.

[13] Носов В.И., Бернштейн Т.В., Носкова Н.В., Храмова Т.В. Элементы теории графов: Учебное пособие под редакцией В.И. Носова, СибГУТИ, Новосибирск, 2008, 107 с.

[14] Кристофидес Н. Теория графов. Алгоритмический подход. Издательство: Мир, 1978. 432 с.

[15] Горбатов В.А. Фундаментальные основы дискретной математики. Учебное пособие. М.: Наука. Физматлит, 544 с.

[16] Kolasinska E. On a Minimum Cycle Basis of a Graph, Zastosowania Matematyki, 16, 4 (1980), p. 631–639.

# KILKA POMYSŁÓW NA TEMAT IZOMORFIZMU POŁĄCZONYCH WYKRESÓW

**Streszczenie** W artykule badamy istnienie izomorfizmów między grafami oraz poszukujemy niezmienników grafów spójnych. Tworzony jest nowy niezmienniczy graf. Metoda może słuïyć do wykrywania izomorfizmów między grafami spójnymi. W pracy użyto pojęcia przestrzeni wektorowej wszystkich prostych cykli grafu i ich sum względem rozłącznych krawędzi oraz przestrzeni wektorowej wszystkich zbiorów grafów uciętych i ich rozłącznych krawędziowo sum. Zbadano metodę konstruowania takich przestrzeni wektorowych: przestrzeni cyklicznej i przestrzeni cięcia. Podano nowe oszacowanie wymiarów tych tego typu przestrzeni wektorowych grafów. Otrzymany niezmiennik jest pokazany na konkretnym przykładzie. W pracy podano kontrprzykład, aby potwierdzić fakt, że zaproponowany niezmiennik może być użyty jako warunek konieczny, ale niewystarczający dla izomorfizmu grafów.

**Słowa kluczowe:** wykres, cykl, zestaw tnący, przestrzeń wektorowa, niezmienny, algorytm izomorfizmu

# MUSIC GENRE RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS

Mateusz Matocha, Sławomir K. Zieliński

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

**Abstract:** The aim of this study was to develop a music genre classifier using convolutional neural networks and to compare its performance with a traditional algorithm based on support vector machines. A distinct feature of the proposed approach was to utilize two-channel stereo signals at the input of the convolutional network. The proposed method yielded similar results compared to those obtained with the traditional approach, demonstrating the potential of the proposed method and indicating the need for its further optimization. Using two-channel stereo signals at the input of the algorithm showed no improvements over the baseline method exploiting single-channel recordings, suggesting that monaural signals fed to the convolutional network might be sufficient to undertake the task of music genre recognition. According to the results, the network 'prioritized' the temporal changes over the frequency variations of the signals. This observation tentatively implies that the classifiers specifically designed to account for temporal changes might potentially better serve the task of music genre recognition than the convolutional neural networks.

**Keywords:** automatic music genre recognition, convolutional neural networks, music information retrieval

## 1. Introduction

The growing popularity of music-on-demand services in the Internet gave rise to the situation where manual labelling of audio recordings according to their styles is no longer practical. There is also a need to automatically organize music content and to make intelligent recommendations to the listeners based on their preferences [10]. Therefore, automatic music genre recognition has recently become one of the most prominent research topics within a broader field of music information retrieval [27], [31], [9].

The remarkable success of the machine learning algorithms based on the convolutional neural networks (CNNs) in the field of image classification [17] suggests that

Advances in Computer Science Research, vol. 14, pp. 125-142, 2018. DOI: 10.24427/acsr-2018-vol14-0008

such algorithms might also exhibit competitive performance when applied to the task of music genre recognition. On the other hand, inherent properties of the convolutional neural networks, outlined in more detail in the next section, may prevent them from achieving as good results as those obtained using 'sound-oriented' algorithms, that is the algorithms which better account for temporal variations of sound, e.g. the recurrent neural networks [5], [12]. Recently, Muraurer and Specht [22] demonstrated that even a traditional algorithm employing hand-engineered features and XGBoost classifier performed better than the CNN. Hence, more research is needed in order to evaluate the applicability of the CNNs in the area of automatic music genre recognition.

The way in which stereophonic recordings are produced is genre specific. For example, in the case of classical music recordings the stereophonic panorama (distribution of audio objects in space) is determined by the position of the musicians and the microphone technique used by a sound engineer (e.g. XY, ORTF or Decca Tree). By contrast, for pop music recordings, stereophonic panorama is typically 'created' artificially, using amplitude panning algorithms in mixing consoles (see [26] for a comprehensive review of audio recording techniques). Hence, it was hypothesized that exploiting two-channel stereophonic signals, as opposed to monophonic ones, could enhance the performance of a music genre classification method.

The purpose of this study is twofold. First, we want to validate the suitability of convolutional neural networks to undertake the task of automatic music genre recognition by comparing its performance with a traditional algorithm based on the hand-crafted features and support vector machines (SVM). Second, we want to check whether there is any merit in using two-channel stereo sound at the input of the convolutional networks, compared to the standard approach exploiting single-channel monaural signals.

## 2. Related Work

The studies in the area of automatic music genre recognition were pioneered in 1995 by Matityaho and Furst [18] and then followed, among other researchers, by Tzanetakis and Cook [32], McKay and Fujinaga [20], Silla et al. [28], and Bhalke et al. [3]. An interested reader is referred to [27], [31] for a comprehensive literature review in this field. Most of the methods developed so far involved a two-stage approach. Music signals were subject to a procedure of hand-crafted feature extraction first, and then the extracted data were fed to the input of a classifier, such as a *k*-NN algorithm, random forest, logistic regression or support vector machines. A meaningful comparison of the results across the studies was impeded by the fact that the researchers used a different number of genre categories, they applied diverse classification metrics, and employed various music corpora. Therefore, to increase the comparability of research, several datasets with manually labelled music recordings were developed and made publically available, most notably GTZAN [32], LMD [29], MSD [2], and FMA [8]. The number of songs in these databases ranged from 1 000 to 1000 000.

As mentioned above, the researchers in the area of music genre recognition use diverse classification metrics, which hinders a consistent comparison of the results across the studies. The average classification accuracy $A$ and the $F1$ metric appear to constitute the most commonly exploited measures. They can be defined using the following two equations, respectively [30]:

$$A = \sum_{i=1}^{L} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i} / L, \tag{1}$$

$$F1 = 2\frac{pr}{p+r}. \tag{2}$$

The parameter $L$ in Eq. (1) denotes the number of music genres (classes) used in a given study. The variables $p$ and $r$ in Eq. (2) represent classification precision and recall [30], respectively.

Convolutional neural networks are artificial neural networks intended for processing data that have a grid-like topology [13]. As already mentioned, they proved to be particularly effective in image classification [17]. Therefore, in order to apply them to the task of classification of audio, most of the researchers convert sound signals into images first and then feed them to the input of the CNN. This conversion is typically accomplished by calculating two-dimensional spectrograms [6], although other forms of visual representation of sound, such as tonnetz-plots and tempograms, could also be used [12].

One of the first attempts to apply convolutional neural networks to the task of automatic classification of music genres was undertaken by Gwardys and Grzywczak [14]. In contrast to a traditional procedure of hand-engineered extraction of signal features, they used the convolutional neural network to automatically generate the features of music recordings. These features were subsequently used as input data of the support vector machines. When applied to the GTZAN [32] dataset, their method yielded a classification accuracy approaching 78%. A year later Rajanna et al. [24] reported another attempt to employ a deep learning technique to the task of music genre classification. They compared a range of traditional procedures of feature extraction followed by two hidden layered feed-forward neural network, yielding rather poor classification results with an accuracy below 39%. Since that time, a markedly increased interest of the research community in convolutional networks and deep learning could be observed. For example, Kim et al. [16] reported that CNNs might be employed to classify music genres with an $F1$ metric of 0.6571, which was demonstrated using the FMA [8] database. Similarly, promising results were also reported by Ghosal and Kolekar [12], Costa et al. [7] as well as by Bahuleyan [1].

Despite a growing body of research univocally supporting a high potential of the convolutional neural networks in the area of music genre recognition, there are some properties of the convolutional neural networks theoretically inhibiting their performance when applied to the classification of audio signals. CNNs are known to be '*approximately invariant to small translations*' [13]. This property is advantageous when the networks are applied to two-dimensional images which have the same interpretation of both dimensions. However, the dimensions of the spectrograms of audio signals have a different physical interpretation (time and frequency). Therefore the above property may be detrimental in terms of music genre classification, as highlighted by Medhat et al. [21]. Moreover, other classification algorithms, such as the recurrent convolutional networks, may be better at 'capturing' the temporal changes of the music signals compared to CNNs [12], [5].

To the best of the authors' knowledge, all of the algorithms used in the area of music genre recognition exploit single-channel signals at their input. While most of the publically available datasets of music excerpts contain the two-channel stereo recordings, they are typically down-mixed to mono (by averaging the stereo signals), before being fed to the input of the classification algorithms. Consequently, potentially important information is discarded. In their work regarding the acoustic scene classification (a different field of research compared to music genre recognition), Ham et al. [15] have recently demonstrated that exploiting two-channel stereo signals could enhance the performance of the audio classification algorithms. Hence, it can be hypothesized that including spatial information conveyed by the two-channel stereo signals could improve the performance of the music genre classification algorithms. This hypothesis was verified in this study.

## 3. Experiments

A 'small' version of the recently developed FMA [8] corpus of music recordings was used as a basis for the research described in this paper. It contained a set of 8 000 music excerpts of 30 seconds in duration, representing the following eight genres: electronic music, experimental, folk, hip-hop, instrumental, international, pop, and rock.

In the first and the second experiment 1999 music recordings were selected from the FMA database. Then, 70% of the recordings were used for **training** purposes whereas the remaining 30% of the excerpts were exploited for **testing**. The following metrics were used to evaluate the classification performance: accuracy, $F1$ metric, and the area under the curve (AUC). In the third experiment, 7994 recordings were selected from the FMA database. They were split in the proportion of 75% to 25% for the training and testing purposes, respectively. For the test employing SVM, the division between the training and testing datasets was slightly different. Namely, 70% of the recordings were used for training and 30% for testing. This difference was taken into consideration in the statistical test comparing the results between the methods. Due to long computational time, a hold-out validation technique was employed. In order to detect a potential problem with over-fitting, the classification learning curves were visually inspected. They were plotted as a function of training epochs both for the training and the testing datasets, respectively.

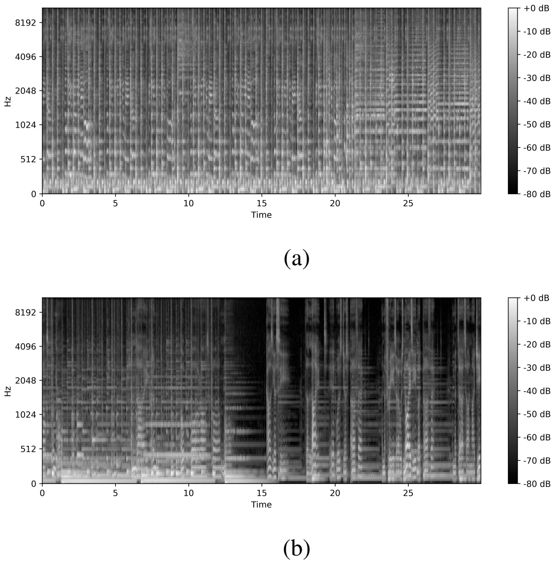
Due to the inconsistencies in sampling rate between the recordings, all the excerpts were down-sampled to 22.05 kHz. The pilot tests and the literature reports [32], [1], [12] confirmed that such sampling rate was adequate for the purpose of automatic genre recognition. The recordings were processed using 1024-samples long time frames with a 50% overlap. A Hanning window was applied to each frame.

Prior to exploiting the CNN, the musical signals were converted into the standard spectrograms (images) using a bank of 128 Mel-frequency filters. Example spectrograms obtained for the selected hip-hop and folk recordings were depicted in Fig. 1.

The proposed architecture of the CNN was implemented in Python programming language using the *keras* and *tensorflow-gpu* libraries. Moreover, the *sklrean* package was used to standardize the data, to split them into the training and testing datasets as well as to run the SVM classifier. The *librosa* package was used in order to generate the required spectrograms. The simulations were accelerated using the NVIDIA graphical processing unit GTX 1080Ti with 11 GB of memory.

### 3.1 Experiment 1 – Initial selection of the network topology

The purpose of the first experiment was to establish the number and the shape of the convolutional filters. The proposed CNN model consisted of three convolutional layers and two fully-connected layers. The three convolutional layers were interleaved by three average pooling layers, as shown in Fig. 2a. Due to the restrictions of available memory space in the graphical processing unit, the number of filters in the first convolutional layer was limited to 32. To reduce the risk of overfitting, a dropout technique (0.5 rate) was applied to the fully-connected layers. A stochastic gradient



(a)

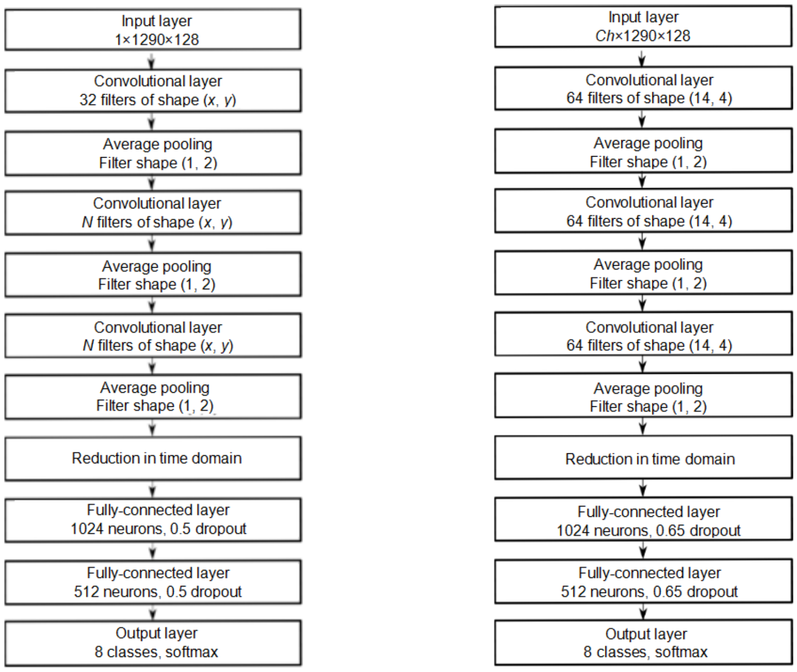
(b)

Fig. 1: Example spectrograms of the selected recordings: (a) hip-hop and (b) folk.

descent (SGD) optimization algorithm was used. In order to reduce the data size at the output of the last average pooling layer, a technique of data reduction in time-domain was performed by calculated average values along the time-axis. A similar approach was proposed by Dieleman [10]. Table 1 shows a set of parameter values used during the optimization in Experiment 1. The obtained results were summarized in Table 2. Since accuracy $A$ and $F1$ metric are commonly quoted classification performance measures in the field of music genre recognition, they were used together to evaluate the models developed in this study.

The best performance was seen with the network topology using 128 filters of a shape of $14 \times 4$ ($length \times height$). In this case, the $F1$ metric was equal to 0.345 which constitutes a mediocre outcome compared to the state-of-the-art algorithms [9]. Nevertheless, in line with the present results, the filters of a size of $14 \times 4$ were employed in the next experiment (see the next section).

The common feature of the best models was an irregular shape of the filters, with a shorter height $y$, and a longer length $x$. This outcome indicates that information conveyed by temporal changes of the signals (horizontal axis of the spectrograms) was more important than information represented by frequency changes (vertical axis of the spectrograms).



(a) (b)

Fig. 2: A model of a neural network used in: (a) Experiment 1, (b) Experiment 2. A symbol *Ch* denotes the number of channels (*Ch* = 1 for monaural signals and *Ch* = 2 for stereophonic signals).

### 3.2 Experiment 2 – Utilizing stereophonic sound

The aim of the second experiment was to assess the merit of using two-channel stereophonic signals at the input of the CNN. To this end two networks were designed, one for the monaural signals and another one for the two-channel stereophonic signals. Their topology was similar to each other, the only difference being the number of the input layers *Ch* (see Fig. 2b). Monaural signals used in this experiment were obtained by averaging the two-channel stereophonic signals.

The number of the convolutional filters was set to 64, preserving the best shape identified in the first experiment $(14 \times 4)$. The reason for using 64 convolutional filters instead of 32 ones, as in the previous experiment, was due to a redesigned architecture of the network and graphical processing unit memory capacity limitations. In the first experiment, the usage of more than 32 filters required more memory than

Table 1: The parameters under optimization in Experiment 1.

| Optimized parameters | A set of tested values |
|---|---|
| Number of filters ($N$) | 32, 64, 96, 129 |
| Filter height ($y$) | 1, 2, 3, 4, 5 |
| Filter length ($x$) | 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 |

Table 2: Overview of the best results obtained in Experiment 1.

| Number of filters | Shape of filters (x, y) | $F1$ metric (Accuracy) |
|---|---|---|
| 32 | (20, 2) | 0.255 (0.337) |
| 32 | (14, 3) | 0.305 (0.353) |
| 64 | (20, 2) | 0.313 (0.363) |
| 128 | (18, 3) | 0.329 (0.378) |
| 64 | (14, 3) | 0.332 (0.375) |
| **128** | **(14, 4)** | **0.345 (0.389)** |

it was available during the tests of the largest filters (e.g. 20 × 4). In the second experiment, the network was redesigned according to the previously obtained results. The change of the filters' shape allowed to increase the number of filters in the first layer from 32 to 64. Due to the encountered problems with overfitting, the dropout rate was increased to 0.65, compared to the previous experiments. The following optimization algorithms were trialled in the pilot test (not reported in the paper): SGD, ADAM, and ADADELTA [25]. Since the ADAM algorithm produced the best results, it was employed in this experiment.

In this experiment the corpus was split into four datasets (quarters), each containing the equal number of tracks per genre. The training process was repeated four times, once per each quarter. Then, the average results were computed. Every model was trained for 50 epochs.

Contrary to the expectation, the performance level of the network exploiting the two-channel stereophonic recordings was slightly worse compared to that obtained using the monaural signals. The average $F1$ metrics achieved for the monaural and stereophonic algorithms were equal to 0.431 and 0.408, respectively.

### 3.3 Experiment 3 – Network optimization and comparison with the SVM classifier

The aim of the last experiment was to undertake the final optimization of the network and to compare its performance with the traditional method based on the support vector machine (SVM). Since the classical network layout, exploiting monaural signals at its input, outperformed the algorithm using the stereophonic signals, the standard topology employing a single-channel input was adopted in this experiment (see Fig. 3). The values of the parameters taken into account during the optimization procedure were presented in Table 3. Due to a large number of all possible combinations of the values (over $4.7 \times 10^{10}$) and limited computational resources, it was not practical to run an exhaustive search. Instead, a heuristic method was used in which a grid search algorithm was executed iteratively. In each iteration, the parameter values which gave rise to a significant deterioration in model accuracy were removed from the parameter grid.

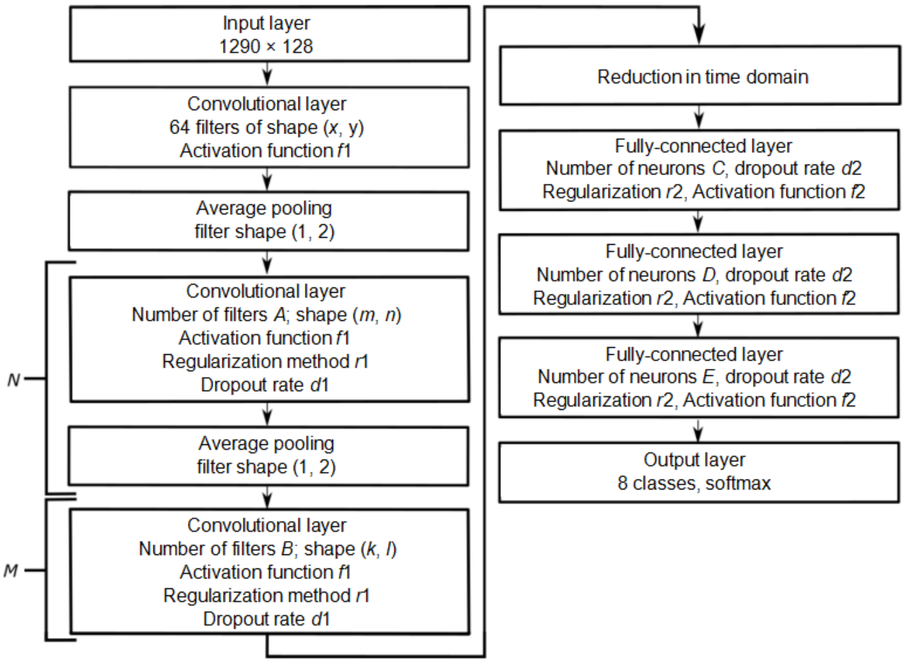


Fig. 3: A model of the neural network used in Experiment 3.

Table 3: The parameters taken into account during the final optimization. See Fig. 3 for explanation of the symbols.

| Parameters | A set of tested values |
|---|---|
| Shape of filters (*x*, *y*) | (20, 2); (14, 3); (14, 4); (5, 1); (8, 1); (10, 1); (13, 1); (15, 1) |
| Activation function ($f1$) | Tanh; ReLU; Exponential Linear Unit (ELU) |
| Number of filters (*A*) | 256; 128; 96; 64; 32 |
| Shape of filters (*m, n*) | (20, 2); (14, 3); (14, 4); (3, 1); (5, 1); (8, 1); (10, 1) |
| Regularization type ($r1$) | L1; L2 |
| Regularization value ($r1$) | 1e-7; 1e-6; 1e-5; 1e-4 |
| Dropout rate ($d1$) | 0; 0.01; 0.05; 0.1; 0.2; 0.5 |
| Number of filters (*B*) | 256; 128; 96; 64; 32 |
| Shape of a filter (*k, l*) | (14, 3); (14, 4); (3, 1); (5, 1); (8, 1); (10, 1) |
| Number of neurons (*C*) | 2048; 1024; 512 |
| Dropout rate ($d2$) | 0.2; 0.3; 0.5; 0.6; 0.7 |
| Regularization type ($r2$) | L1; L2 |
| Regularization value ($r2$) | 0.025; 0.03; 0.035; 0.04 |
| Activation function ($f2$) | Tanh; ReLU; ELU |
| Number of neurons (*D*) | 2048; 1024; 512 |
| Number of neurons (*E*) | 2048; 1024; 512 |
| Number of layers (*N*) | 4; 3; 2; 1 |
| Number of layers (*M*) | 4; 3; 2 |

The optimized version of the network yielded the best results compared to the previous experiments. The $F1$ metric and the accuracy of the best model reached the values of 0.6 and 0.605, respectively.

The hyper-parameters yielding the best performance of the network were gathered in Table 4. It was interesting to observe that the shape of the filter in the first convolutional layer of the best model was ‘reduced’ to a single dimension (10, 1). This outcome indicates that the initial layers of the network tended to ignore frequency information and to prioritize temporal information. Note, that a similar effect, although less pronounced, was also observed in the first experiment. In that case a shape of the best filter was $14 \times 4$ (see Table 2). Hence, it might be tentatively concluded that in order to obtain the best results in sound classification, irregular shapes of the convolutional layers, prioritizing time-axis of the spectrograms, might be beneficial. This supposition is supported by the recent study of Ghosal and Kolekar [12] who deliber-

Table 4: Overview of the best parameters in Experiment 3. See Fig. 3 for explanation of the symbols.

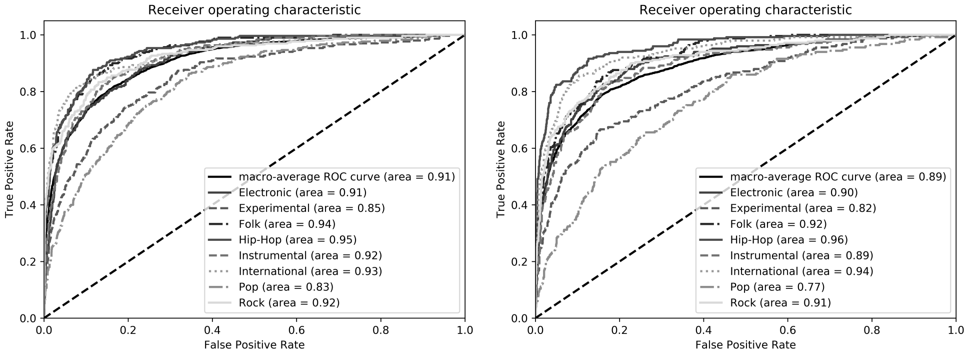
| Optimized parameter | Value | Optimized parameter | Value |
|---|---|---|---|
| Shape of filters ($x$, $y$) | (10,1) | Number of neurons ($C$) | 2048 |
| Activation function ($f1$) | ELU | Dropout rate ($d2$) | 0.5 |
| Number of filters ($A$) | 64 | Regularization type ($r2$) | L2 |
| Shape of filters ($m$, $n$) | (8, 1) | Regularization value ($r2$) | 0.04 |
| Regularization type ($r1$) | L1 | Activation function ($f2$) | ReLU |
| Regularization value ($r1$) | 1e-6 | Number of neurons ($D$) | 2048 |
| Dropout rate ($d1$) | 0 | Number of neurons ($E$) | 512 |
| Number of filters ($B$) | 64 | Number of layers ($N$) | 2 |
| Shape of a filter ($k$, $l$) | (5, 1) | Number of layers ($M$) | 3 |

ately restricted the filters to ‘1D convolution’ along the time-axis in their music genre classification algorithm.

In order to compare the performance of the optimized CNN with a traditional method, it was decided to use a set of 518 hand-engineered features and to feed them to the input of the support vector machine (SVM). The rationale for choosing the SVM classifier was related to its well-known generalizability property. The above-mentioned features were calculated by the authors of the FMA music dataset [8]. They contained the standard metrics commonly used in music information retrieval applications, such as zero crossing, spectral centroid or spectral bandwidth. Several kernels were trailed in the classifier (not reported in the paper) indicating that the one employing the radial basis function (RBF) produced the best results. The hyper-parameters $C$ and gamma of the support vector machine were equal to 10 and 0.3, respectively.

According to the obtained results, the classifier based on the hand-engineered features yielded an increase in accuracy by 0.1%. However, according to the binomial test of proportions, the above increment was statistically insignificant. Hence, it cannot be concluded that the traditional method outperformed the CNN.

The receiver operating curves (ROC) obtained for the traditional method and the one based on the CNN were illustrated in Fig. 4. It can be seen that the overall performance of the CNN and the traditional algorithm was similar. However, while the traditional algorithm was better at classification of pop, electronic and experimental music, the CNN-based algorithm was a ‘winner’ in terms of hip-hop and international music classification. Therefore, these two algorithms seem to complement each other

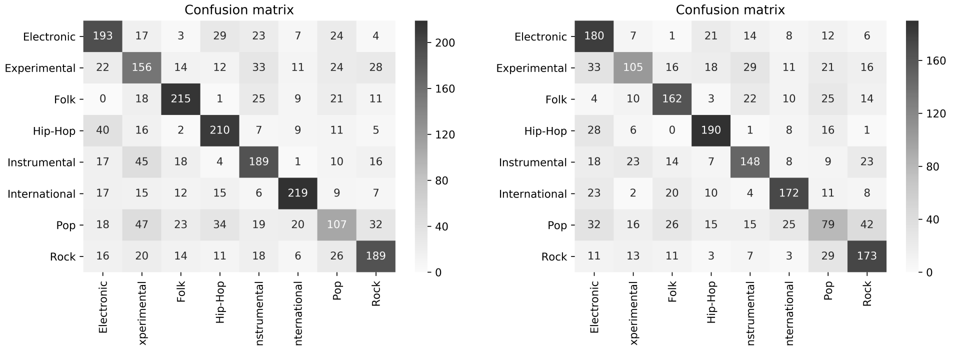


(a) (b)

Fig. 4: ROC curves: (a) SVM with RBF kernel; (b) CNN

and theoretically might be used in parallel in an ensemble of classifiers. Verification of this conclusions was beyond the scope of the study and was left for future work.

A confusion matrix obtained for the SVM-based method was presented in Fig. 5a. It can be seen that prediction accuracy varied between genres. The worst results were obtained for pop music which was often misclassified as rock or electronic music. A method incorporating CNN also exhibited difficulty with the classification of pop music recordings (see Fig. 5b).



(a) (b)

Fig. 5: Confusion matrices: (a) SVM with RBF kernel; (b) CNN. Predicted classes are listed at the bottom of the chart.

Table 5 illustrates how the obtained results compare to those obtained by other researchers. The data presented in the table were limited to the studies involving CNNs and has to be treated with some caution, as the direct comparison between the studies is hindered by the inconsistencies in terms of the music datasets and the evaluation metrics used by various authors. While the method proposed in this paper did not match the state-of-the-art algorithms described in the most recent literature, the obtained results indicate a high potential of the CNNs in the area of music genre recognition. If the results gathered in the table were further limited to those employing the FMA dataset, the CNN-based algorithm proposed in the present paper was only marginally worse compared to the one developed by Kim et al. [16].

Table 5: Comparison of the results in the literature (limited to the studies using CNNs for automated music genre recognition).

| Authors | Source | Music dataset | Accuracy | $F1$ | AUC |
|---|---|---|---|---|---|
| Ghosal and Kolekar (2018) | Tab. 2 in [12] | GTZAN [32] | 0.942 | — | — |
| Costa et al. (2017) | CNN in Tab. 3 in [7] | LMD [29] | 0.83 | 0.836 | — |
| Costa et al. (2017) | CNN in Tab. 8 in [7] | ISMIR [4] | 0.859 | 0.863 | — |
| Gwardys and Grzywczak (2014) | [14] | GTZAN [32] | 0.78 | — | — |
| Kim et al. (2018) | Tab. 4 in [16] | FMA [8] | — | 0.6571 | — |
| Bahuleyan (2018) | VGG-16 CNN F. Tun. in Tab. 2 in [1] | Audio Set [11] | 0.64 | 0.61 | 0.889 |
| **Matocha and Zieliński** | **Present study** | **FMA [8]** | **0.605** | **0.6** | **0.89** |
| Murauer and Specht (2018) | CNN in Tab. 5 in [22] | FMA [8] | — | 0.48 | — |
| Oramas et al. (2018) | Audio (A) in Tab. 6 in [23] | MuMu [19] | — | — | 0.888 |
| Oramas et al. (2018) | CNN_Audio in Tab. 2 in [23] | MSD [2] | — | 0.336 | — |
| Choi et al. (2017) | Fig. 3 in [5] blue dashed line | MSD [2] | — | — | 0.83 |

## 4. Conclusions

The aim of this paper was to develop a music genre classifier using a convolutional neural network (CNN) and to compare its performance with a traditional algorithm based on hand-engineered audio metrics and support vector machines. A novel feature of the proposed approach was to utilize two-channel stereo signals at the input of the convolutional network. To the best of the authors' knowledge, no-one has attempted to employ stereophonic signals at the input of CNN to classify music genres yet. According to the results, using two-channel stereo signals showed no improvements over the baseline method exploiting single-channel recordings. On one hand, this outcome tentatively suggests that monaural signals fed to the convolutional network might be sufficient to undertake the task of music genre recognition. On the other hand, one may not exclude a possibility that the best topology of a network, handling two-channel stereo sounds, has not been identified yet. The latter conclusion is supported by the recent work of Ham et al. [15] in the area of acoustic scene classification. Similarly to this study, they observed that the algorithm exploiting the two-channel stereo sounds at the input of the CNN, used in isolation, produced worse results than the standard method. However, when the 'stereophonic' algorithm was employed as a part of an ensemble of classifiers, the overall performance of the method markedly improved. Therefore, further work is required before one could dismiss the usefulness of spatial information conveyed by two-channel stereo sounds in the area of deep learning and automatic classification of music genres.

Another contribution of this study was to quantify the accuracy yielded by the CNN when applied to the task of music genre recognition. Only a few papers devoted to this research domain have been published so far. To the best of the authors' knowledge, there are only two papers in which the researchers applied the CNN to the new FMA (2018) corpus [8]. While the results obtained in this study were 5% worse than those achieved by Kim et al. [16], they proved to be 12% better than the results published by Murauer and Specht [22]. The classification outcomes obtained in the area of machine learning do not always depend on the type of a classification method employed but also on "the match" between the database characteristics and the properties of a chosen classifier. Hence, the results presented in this paper could help other researchers to make an informed choice regarding a classification method for the task of the music genre recognition, particularly in relation to the new FMA dataset [8].

The proposed method yielded similar results compared to those obtained with the traditional approach, indicating that these methods could be used interchangeably or, which constitutes the subject of a future verification, in an ensemble of two algorithms working in parallel. While the method proposed in this paper did not match the state-of-the-art algorithms described in the most recent literature, the obtained results demonstrated a high potential of the CNNs in the area of automatic music genre recognition.

The implemented CNN put higher importance to information carried by the temporal changes rather than to the frequency variations of the processed signals. This observation implies that the classifiers specifically designed to account for temporal changes, such as the recurrent neural networks, might better serve the task of music genre recognition than the convolutional neural networks. This conclusion is in accordance with the preliminary results obtained recently by Choi et al. [5] as well as by Ghosal and Kolekar [12].

## Acknowledgments

This work was supported by a grant S/WI/3/2018 from Białystok University of Technology and funded from the resources for research by Ministry of Science and Higher Education.

## References

[1] H. Bahuleyan: Music Genre Classification using Machine Learning Techniques, arXiv preprint arXiv:1804.01149, [https://arxiv.org/abs/1804.01149v1], Access time: November 5, 2018.

[2] T. Bertin-Mahieux, D. Ellis, B. Whitman and P. Lamere: The Milion Song Dataset, In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011.

[3] D.G. Bhalke, B. Rajesh and D.S. Bormane: Automatic Genre Classification Using Fractional Fourier Transform Based Mel Frequency Cepstral Coefficient and Timbral Features, Archives of Acoustics, vol. 42(2), pp. 213–222, 2017.

[4] P. Cano, E. Gomez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S.Streich and N. Wack: ISMIR 2004 audio description contest, Technical report, Music Technology Group – Universitat Pompeu Fabra, 2006.

[5] K. Choi, G. Fazekas and K. Cho: Convolutional recurrent neural networks for music classification, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017.

[6] Y.M.G. Costa, L.S. Oliveira, A.L. Koericb and F. Gouyon: Music genre recognition using spectrograms, In Proceedings of the 18th International Conference on Systems, Signals and Image Processing, 2011.

[7] Y.M.G. Costa, L.S. Oliveira and C.N. Silla Jr.: An evaluation of convolutional neural networks for music classification using spectrograms, Applied Soft Computing, vol. 52, pp. 28–38, 2017.

[8] M. Defferrard, K. Benzi, P. Vandergheynst and X. Bresson: FMA: A Dataset for Music Analysis, In Proceedings of the 18th International Society for Music Information Retrievel Converence (ISMIR), 2017.

[9] M. Defferrard, S.P. Mohanty, S.F. Carroll and M. Salathé: Learning to Recognize Musical Genre from Audio: Challenge Overview, In Companion of the Web Conference 2018. Lyon, France, April 23–27, 2018.

[10] S. Dieleman: Recommending music on Spotify with deep learning, Site [http://benanne.github.io/2014/08/05/spotify-cnns.html], Access time: November 6, 2018

[11] J.F. Gemmeke, D.P.W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R.Ch. Moore, M. Plakal and M. Ritter: Audio set: An ontology and human-labeled dataset for audio events, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 776–780, 2017.

[12] D. Ghosal and M.H. Kolekar: Music Genre Recognition using Deep Neural Networks and Transfer Learning, In Proceedings of Interspeech, September, 2018.

[13] I. Goodfellow, Y. Bengio, and A. Courville: Deep Learning, MIT Press, 2016.

[14] G. Gwardys and D. Grzywczak: Deep Image Features in Music Information Retrieval, Intl Journal of Electronics and Telecommunications, vol. 60(4), pp. 321–326, 2014.

[15] Y. Ham, J. Park and K. Lee: Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification, Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop, 2017.

[16] J. Kim, M. Won, X. Serra and C.C.S. Liem: Transfer Learning of Artist Group Factors to Musical Genre Classification, In Companion of the the Web Conference 2018, Lyon, France, April 23–27, 2018.

[17] A. Krizhevsky, I. Sutskever and G.E. Hinton: ImageNet classification with deep convolutional neural networks, In Advances in neural information processing systems, vol. 25(2), pp. 1097–110, 2012.

[18] B. Matityaho and M. Furst: Neural network based model for classification of music type, In Proceedings of the Convention of Electrical and Electronics Engineers in Israel, pp. 1–5, March, 1995.

[19] J. McAuley, C. Targett, Q. Shi and A. Van Den Hengel: Image-based recommendations on styles and substitutes, In Proceedings of the 38th Interna-

tional ACM SIGIR Conference on Research and Development in Information Retrieval, 2015.

[20] C. McKay and I. Fujinaga: Music genre classification: is it worth pursuing and how can it be improved?, In Proceedings of the ISMIR, Victoria, Canada, October, 2006.

[21] F. Medhat, D. Chesmore and J. Robinson: Automatic Classification of Music Genre using Masked Conditional Neural Networks, In Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 979–984, 2017.

[22] B. Murauer and G. Specht: Detecting Music Genre Using Extreme Gradient Boosting, In Companion of the Web Conference 2018. Lyon, France, April 23–27, 2018.

[23] S. Oramas, F. Barbieri, O. Nieto and X. Serra: Multimodal Deep Learning for Music Genre Classification, Transactions of the International Society for Music Information Retrieval, 1(1), pp. 4–21, 2018.

[24] A.R. Rajanna, K. Aryafar, A. Shokoufandeh and R. Ptucha: Deep Neural Networks: A Case Study for Music Genre Classification, In Proceedings of the 14th International Conference on Machine Learning and Applications, 2015.

[25] S. Ruder: An overview of gradient descent optimization algorithms, Site: `[https://arxiv.org/pdf/1609.04747.pdf]`, 2017. Access time: November 6, 2018.

[26] F. Rumsey and T. McCormick: Sound and Recording, Focal Press, 2014.

[27] N. Scaringella, G. Zoia and D. Mlynek: Automatic Genre Classification of Music Content. A survey, IEEE Signal Process. Mag., vol. 23(2), pp. 133141, 2006.

[28] C. Silla, C. Kaestner and A. Koerich: Automatic Music Genre Classification Using Ensemble of Classifiers, IEEE International Conference on Systems, Man, and Cybernetics, pp. 1687–1692, 2007.

[29] C. Silla, A. Koerich and C. Kaestner: The Latin Music Database, In Proc. of the 9th International Conference on Music Information Retrieval (ISMIR), 2008.

[30] M. Sokolova and G. Lapalme: A systematic analysis of performance measures for classification tasks, Information Processing and Management, vol. 45, pp. 427–437, 2009.

[31] B.L. Sturm: A Survey of Evaluation in Music Genre Recognition, In A. Nürnberger et.al. (eds) Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation. AMR 2012. Lecture Notes in Computer Science, vol. 8382. Springer, Cham. 2014

[32] G. Tzanetakis and P. Cook: Musical genre classification of audio signals, IEEE Trans. Speech Audio Processing, vol. 10, no. 5, pp. 293–302, 2002.

*Mateusz Matocha, Sławomir K. Zieliński*

# ROZPOZNAWANIE GATUNKÓW MUZYCZNYCH Z UŻYCIEM SPLOTOWYCH SIECI NEURONOWYCH

**Streszczenie** Celem niniejszej pracy było opracowanie klasyfikatora gatunków muzycznych z użyciem splotowych sieci neuronowych i porównanie go z tradycyjnym algorytmem opartym na maszynie wektorów wspierających. Wyróżniającą cechą zaproponowanego podejścia było wykorzystanie dwu-kanałowego dźwięku stereofonicznego na wejściu sieci splotowej. Zaproponowana metoda dała podobne wyniki do rezultatów otrzymanych z użyciem podejścia tradycyjnego, demonstrując potencjał zaproponowanej metody oraz wskazując na potrzebę jej dalszej optymalizacji. Wykorzystanie dwu-kanałowego dźwięku stereofonicznego na wejściu algorytmu nie poprawiło wyników w porównaniu z metodą bazową wykorzystującą nagrania jednokanałowe, sugerując, iż zastosowanie dźwięków monofonicznych na wejściu splotowej sieci neuronowej jest adekwatne do celów rozpoznawania gatunków muzycznych. Zgodnie z uzyskanymi wynikami, sieć 'potraktowała priorytetowo' zmiany czasowe w porównaniu ze zmianami częstotliwościowymi sygnałów. Obserwacja ta pozwala wstępnie przypuszczać, że klasyfikatory specjalnie zaprojektowane, by uwzględnić zmiany czasowe, potencjalnie mogłyby lepiej służyć celom rozpoznawania gatunków muzycznych niż neuronowe sieci splotowe.

**Słowa kluczowe:** automatyczne rozpoznawanie gatunków muzycznych, splotowe sieci neuronowe, pozyskiwanie informacji w muzyce

Badania zostały zrealizowane w ramach pracy S/WI/3/2018 sfinansowanej ze środków na naukę MNiSW.

# DIFFERENT APPROACHES TO INFEASIBLE SOLUTIONS IN EVOLUTIONARY ALGORITHMS FOR THE ORIENTEERING PROBLEM

Krzysztof Ostrowski

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

**Abstract:** The Orienteering Problem (OP) is a combinatorial optimization problem defined on weighted graphs. The purpose of the OP is to find a path of limited length which maximizes total profit (collected in vertices). This paper presents comparison of different approaches to infeasible solutions (too long paths) in evolutionary algorithms solving the OP. A group of evolutionary algorithms (varying in crossover and selection operators) was tested in different configurations: with and without infeasible solutions in populations. Parameters for all algorithm configurations were obtained from automatic tuning procedure (ParamILS). Results show that presence of too long paths in a population can improve quality of resulting solutions. The presented metaheuristic generated optimal or close to optimal solutions for the tested benchmark networks.

**Keywords:** infeasible solutions, evolutionary algorithms, Orienteering Problem

## 1. Introduction

The Orienteering Problem (OP) is a combinatorial optimization problem defined on graphs. Its practical applications include tourist trip planning, transport logistics, DNA sequencing problem and others [1] [2]. The OP is an NP-hard problem [3] and computing exact solutions for larger graphs can be very time-consuming. For this reason most of proposed approaches to solve the OP were heuristic. Evolutionary algorithms (EAs) are among most popular metaheuristics for solving optimization problems. Performance of EAs is determined by various factors like choice of recombination operators and parameters values. For optimization problems with constraints another important aspect is a way of dealing with infeasible solutions in a population. The paper presents comparison of two groups of EAs solving the OP: one without infeasible solutions (presented in [4]) and another with infeasible solutions (adaptive

Advances in Computer Science Research, vol. 14, pp. 143-161, 2018. DOI: 10.24427/acsr-2018-vol14-0009

penalty). The results show superiority of EAs with infeasible solutions in populations. The article is organized as follows. Section 2 presents mathematical definition of the OP. Section 3 presents a review of the literature on the OP. In Section 4 the proposed EA is described. Experimental results are included in section 5 and conclusions are drawn in Section 6.

## 2. Mathematical definition of the Orienteering Problem

The OP is defined on a weighted graph $G = (V,E)$. Each edge has associated a nonnegative cost and each vertex has a nonnegative profit. The purpose of the OP is to find a path (or cycle) between a given pair of vertices ($s$ - start vertex, $e$ - end vertex) which maximizes total collected profit (sum of profits of visited vertices) and its total cost (sum of costs of visited edges) is limited by a given constraint ($C_{max}$). Each vertex can be included in the path at most once (except the situation when $s = e$). Let $w_{ij}$ be a cost of edge $(i,\ j)$ and $p_i$ be a profit of vertex $i$. Let $x_{ij}$ be a binary variable equal to 1 if a solution contains edge $(i,\ j)$ and 0 otherwise. Let $r_i$ be a position of vertex $i$ in a solution - it is defined only for vertices included in a path. The OP can be formulated mathematically:

$$max \sum_{i \in V} \sum_{j \in V} (p_i \cdot x_{ij}) \tag{1}$$

$$\sum_{i \in V} \sum_{j \in V} w_{ij} \cdot x_{ij} \leq C_{max} \tag{2}$$

$$\sum_{j \in V} x_{sj} = \sum_{i \in V} x_{ie} = 1 \tag{3}$$

$$\underset{k \in V \setminus \{s,e\}}{\forall} (\sum_{i \in V} x_{ik} = \sum_{j \in V} x_{kj} \leq 1) \tag{4}$$

$$r_s = 1 \tag{5}$$

$$\underset{i \in V,\ j \in V \setminus \{s\},}{\forall} (x_{ij} = 1 \Rightarrow r_j = r_i + 1) \tag{6}$$

Objective function maximization is described by formula 1. Constraint 2 relates to maximal path cost, which cannot exceed $C_{max}$. Constraint 3 indicates that the path starts in vertex $s$ and ends in vertex $e$ while formula 4 guarantees that all other vertices are included at most once in the path. Formulas 5-6 assure that the solution is a continuous path without sub-cycles.

## 3. Literature review of the Orienteering Problem

The OP was introduced in 1984 [5] and since then various approaches have been proposed to solve the problem. Branch-and-cut and branch-and-bound methods are among few exact algorithms applied for the OP [6] [7]. These methods usually needed long time to solve larger OP instances. Therefore most researchers concentrated on heuristic methods solving the OP.

Tsiligirides [5] proposed first method for the OP (S-algorithm), which based on the Monte Carlo method and heuristic procedure. Golden et al. [3] proposed another approach (use of greedy route construction and a centre-of-gravity heuristic). Chao et al. [8] introduced a two-step iterative heuristic, which obtained results of highest quality at that time.

Tasgetiren [9] was the first to present a genetic algorithm for the OP. The genetic operators used were: tournament selection, injection crossover and mutation with local search methods (add, omit, replace and swap operators). A tabu search heuristic for the OP was presented by Gendreau et al. [10]. High-quality solutions were obtained by the algorithm on randomly generated test instances (up to 300 nodes).

Vansteenwegen et al. [11] applied the guided local search method (GLS) for the Team Orienteering Problem (TOP). This method uses local search heuristics (like insert, replace, 2-opt) and reduces the likelihood of becoming trapped in a local optimum (thanks to disturb operator). The GLS meta-heuristic method yields satisfactory results for small-sized networks and is used in the Mobile Tourist Guide [1]. Heuristics for the TOP generate $m$ disjoint routes (except start/end vertices) and total collected profit is maximized. For $m = 1$ the GLS solves classic Orienteering Problem.

In 2009 Schilde et al. [12] published two metaheuristics: Variable Neighbourhood Search (VNS) and Ant Colony Optimization (ACO). For standard benchmark instances both algorithms achieved better average results than Chao's method [8] and GLS [11].

Souffriau et al. [13] proposed Greedy Randomized Adaptive Search Procedure (GRASP) for the TOP. Later the method was adapted to the classic OP by Campos et al. [14]. The authors also added a path relinking (PR) method to GRASP. Construction of initial solutions is partially greedy and partially random: ratio between greediness and randomness is used when inserting new vertices (four different insertion methods are used). Afterwards, local search procedures (exchange and insert) are applied in order to reduce path length and increase its profit. GRASPwPR metaheuristic has an additional step (path relinking), which is performed for each pair of solutions $P_1$ and $P_2$ generated by GRASP: the $P_1$ path is gradually transformed into the $P_2$ by a sequence of vertex insertions and deletions. The best intermediate paths are then improved by local search operators. GRASPwPR was tested on many benchmark instances and generated very high-quality solutions (one of the best among metaheuristics).

The author's research is focused on metaheuristic solutions for various problems from the OP family (mainly evolutionary algorithms and their composition with local search methods). Apart from the classic OP [15] [4] [16] [17] the author also proposed metaheuristics for the Time-Dependent Orienteering Problem (costs of edges vary with time) [18] [19] [20] focusing also on practical aspects of the problem (trip planning in public transport networks). Obtained results showed advantage of evolutionary algorithms over other known meta-heuristics (like GLS, GRASP, GRASPwPR), especially for larger test instances.

## 4. Description of the proposed evolutionary algorithm

The author proposed evolutionary algorithm (EA) with embedded local search operators to solve the Orienteering Problem. Path representation is used in the EA: successive genes in a chromosome are equivalent to successive vertices included in a solution path. At first, an initial population of $P_{size}$ random routes is created. Afterwards, evolutionary phase takes place - operators of selection, crossover, mutation and disturb are applied repeatedly. Mutation, crossover and disturb operators can be either random or heuristic (local search) and frequency of using random and heuristic operators is determined by algorithm parameters (heuristic coefficients). Evolutionary phase terminates after a fixed number $N_g$ of generations, or earlier if there have been no improvements in the last $C_g$ generations. Finally the population undergo local improvement procedure. The best feasible path (with highest total profit) obtained during algorithm run is EA final result. Three different crossover operators and three different selection methods were tested. The algorithm is derived from [4]. However, in the previous publication only feasible solutions were present in populations while in this paper a comparison between different ways of dealing with solutions infeasibility is made. A short description of the algorithm is given below (more details about the operators are presented in [4]).

### 4.1 Selection

Three different selection procedures were tested:
1) Unbiased tournament parent selection [21]

2) Fitness proportionate parent selection with stochastic universal sampling [22]
3) Deterministic crowding survivor selection [23]
First two methods (parent selections) are executed at the beginning of each EA iteration and they create intermediate population, which undergo crossover and mutation. The last method (deterministic crowding) works differently: it is executed after crossover procedure and consists of competitions of child-parent pairs for a place in the next generation.

### 4.2 Crossover

Initially, pairs of individuals (parents) are randomly chosen from the population (number of pairs depends on crossover probability). Afterwards, each pair undergo crossover procedure. Crossover methods tested in the evolutionary algorithms are: 2-point crossover [4], injection crossover [9], and path relinking crossover [14]. Each crossover variant has two versions: heuristic and random. Probability of using heuristic version is determined by an algorithm parameter (crossover heuristic coefficient).

### 4.3 Mutation

First, a group of individuals is randomly chosen from a population (size of the group depends on mutation probability). Afterwards, each of them undergoes mutation, which consists of 2-opt procedure (reversing one path fragment to shorten the path) as well as one vertex insertion or deletion. Afterwards, a small group of individuals undergo disturb procedure. This is another type of mutation which removes some path fragment. This procedure helps to escape from local optima but it is destructive and should be executed rarely. Insert/delete/disturb procedures have two types: heuristic (local search) and random. Frequency of these two types of mutation is determined by algorithm parameters (heuristic coefficients) in an analogical way to crossover procedure. For more details regarding mutation see [4].

### 4.4 Approaches to infeasibility

In this paper two different approaches were tested.

**No infeasible solutions allowed:** This approach was described in [4]. For feasible solutions the fitness function was equal to path profit but for infeasible solutions the fitness was 0. Genetic operators weren't allowed to create too long paths.

**Adaptive penalty of infeasible solutions:** Too long paths are allowed in populations (they can be created by genetic operators) but they are penalized by fitness function. Let $S$ be a path with total profit $p_S$ and total cost $c_S$. Fitness function of a feasible path (not exceeding $C_{max}$) is equal to its profit (as in previous subsection). If path $S$ is too long its fitness is expressed by the following formula:

$$fitness(S) = p_S \cdot (\frac{C_{max}}{c_S})^k \tag{7}$$

Fitness function decreases as solution is farther from feasibility boarder $C_{max}$. Parameter $k$ represents penalty severity. For $k = 1$ fitness function can be interpreted as expected profit of a path after shortening it to $C_{max}$ limit. Larger $k$ values mean stronger penalties. Fitness function is adaptive and depends on number of infeasible solutions in a population. At the beginning $k = 1$ and every 10 generations number of infeasible solutions is checked. If more than $\alpha$ percent of individuals are infeasible then $k$ is increased by 0.1. It enables to control number of infeasible solutions in a population.

### 4.5 Algorithm parameters

Algorithm parameters are described in table 1 and table 2. Probability parameters $p$ describe percentage of population which is chosen for mutation/crossover/disturb procedures in each generation. Heuristic coefficient parameters $z$ determine the probability of using heuristic operator version (1-$z$ is the probability of using random version). Values of population parameters from table 2 were set by the author. Population size is a compromise between exploration ability and computation time. Parameters associated with generations number were determined during earlier experiments and should not stop EA prematurely. The remaining parameters values (table 1) were determined by automatic tuning procedure - Parameters Iterated Local Search (ParamsILS) [24]. This meta-algorithm searches multi-dimensional parameter space using local search procedures. Vectors of parameter values (meta-solutions) processed by ParamsILS are evaluated by averaging results from multiple runs of the tuned EA. More details about the tuning procedure can be found in [4].

## 5. Experimental results

All experiments (calibration and testing) were carried out on a computer with Intel i7 3.6 GHz processor and 4GB of RAM. Programs were implemented in C++ and executed on Linux operating system.

**Table 1.** EA parameters tuned automatically

| parameter | description |
|---|---|
| $p_k$ | crossover probability |
| $p_m$ | mutation probability |
| $p_z$ | disturb probability |
| $z_k$ | crossover heuristic coefficient |
| $z_m$ | mutation heuristic coefficient |
| $z_z$ | disturb heuristic coefficient |
| $\alpha$ | max. percentage of infeasible solutions in a population |

**Table 2.** EA parameters set by the author

| parameter | description | value |
|---|---|---|
| $P_{size}$ | population size | 100 |
| $Ng$ | maximum number of generations | 5000 |
| $Cg$ | maximum number of generations without improvement | 500 |

## 5.1 Problem instances

In table 3 there are all OP instances used in this experiment. Class I tests come from TSPLIB library (distance matrices in XML) and were adapted to the OP by Fischetti et al. [6]. Profits of vertices were generated according to the formula:

$$p_i = 1 + (7141 \cdot i + 73)(mod\,100) \tag{8}$$

where $p_i$ is profit of vertex $i$. $C_{max}$ values for class I instances were set as 50 percent of shortest hamiltionian cycles. Distances between vertices were truncated to integer numbers.

Class II tests were Vehicle Routing Problem (VRP) instances adapted to OP by Fischetti et al [6]. In these instances customer demands from VRP were interpreted as vertices profits in the OP. $C_{max}$ values were set as 25, 50 and 75 percent of shortest hamiltonian cycles. Distances between vertices were rounded to nearest integers.

Class III tests were created by the author and base on a network of 908 cities in Poland [25]. Profits are equal to numbers of inhabitants (expressed in thousands) and graph weights were calculated as big circle distances between cities (in km). For all instances of all classes paths start and end in vertex 1.

**Table 3.** All problem instances with $C_{max}$ values

| Class | Instance name | $C_{max}$ | Class | Instance name | $C_{max}$ |
|---|---|---|---|---|---|
| I | kroA100 | 10641 | II | eil101A | 158 |
| | kroB100 | 11071 | | cmt121A | 137 |
| | kroC100 | 10375 | | cmt151A | 175 |
| | kroD100 | 10647 | | cmt200A | 191 |
| | kroE100 | 11034 | | gil262A | 595 |
| | rd100 | 3955 | | eil101B | 315 |
| | eil101 | 315 | | cmt121B | 273 |
| | lin105 | 7190 | | cmt151B | 350 |
| | pr107 | 22152 | | cmt200B | 382 |
| | gr120 | 3471 | | gil262B | 1189 |
| | pr124 | 29515 | | eil101C | 472 |
| | bier127 | 59141 | | cmt121C | 409 |
| | pr136 | 48386 | | cmt151C | 525 |
| | gr137 | 34927 | | cmt200C | 573 |
| | pr144 | 29269 | | gil262C | 1784 |
| | kroA150 | 13262 | III | pl500 | 500 |
| | kroB150 | 13065 | | pl1000 | 1000 |
| | pr152 | 36841 | | pl1500 | 1500 |
| | u159 | 21040 | | pl2000 | 2000 |
| | rat195 | 1162 | | pl2500 | 2500 |
| | d198 | 7890 | | pl3000 | 3000 |
| | kroA200 | 14684 | | | |
| | kroB200 | 14719 | | | |
| | ts225 | 63322 | | | |
| | pr226 | 40185 | | | |
| | gil262 | 1189 | | | |
| | pr264 | 24568 | | | |
| | pr299 | 24096 | | | |
| | lin318 | 21015 | | | |
| | rd400 | 7641 | | | |

### 5.2 Tuning results

Calibration process was carried out on three problem instances: pr299 and rd400 (from class I) and gil262 (from class II). These instances are among largest with known optimal solutions. To assess quality of the algorithm with given parameter values it was run 10 times for each calibration network and its results were averaged. Gaps are expressed in percent and were obtained according to formula $100 \cdot (1 - \frac{P_{alg}}{P_{opt}})$ where $P_{alg}$ is average route profit obtained by EA while $P_{opt}$ is profit of an optimal route. The experiment was conducted for 18 different algorithm versions (all combinations of 3 selection procedures, 3 crossovers and 2 approaches to infeasible solutions).

The calibration results are presented in table 4. They were divided into two groups (varying in a way of dealing with infeasible solutions) and results of the best configurations of each group are in bold. It can be seen that for all 9 combinations (varying in crossover and selection methods) EAs with infeasible solutions obtain better average results than those without infeasible solutions (the difference is about 0.4 percent). The difference between EAs with and without infeasible solutions is biggest for rd400 network (0.6 percent on average), which suggests that searching both sides or feasibility boarder is more important for instances with larger solution space.

In almost all cases the best results were obtained by a combination of 2-point crossover and deterministic crowding selection. Solutions of the highest quality (only 0.13 percent of average gap to optimal solutions) were produced by EA version with adaptive penalty - they are 0.5 percent better than those obtained by the best EA configuration without infeasible solutions.

Calibration results show the importance of using local search operators in EAs (heuristic coefficients between 0.6 and 1 in most cases). One can see that disturb procedures (which can be destructive when overused) are executed rarely compared to mutation and crossover operators. It can also be seen that penalty parameter ($\alpha \geq 70$) is not severe in most cases - a lot of infeasible solutions are allowed in populations.

**Table 4.** Tuning results for different EA configurations with best found sets of parameter values and average gaps to optimal results for calibration networks. Crossover types: 2P - two point crossover, INJ - injection crossover, PR - path relinking crossover. Selection types: TUR - unbiased tournament selection, SUS - fitness proportionate selection with stochastic universal sampling, CRO - deterministic crowding. The result of the best EA configuration from each group in bold.

| Infeasible | Crossover | Selection | Calibrated parameters | | | rd400 | pr299 | gil262C | All 3 networks |
|---|---|---|---|---|---|---|---|---|---|
| solutions | | | $p_k$ $p_m$ $p_z$ | $z_k$ $z_m$ $z_z$ | α | gap (%) | gap (%) | gap (%) | avg. gap (%) |
| No infeasible solutions | 2-P | TUR | 0.6 1.0 0.70 | 0.4 0.6 0.4 | - | 3.47 | 3.32 | 1.14 | 2.64 |
| | | SUS | 0.6 1.0 0.10 | 0.8 0.8 0.6 | - | 2.37 | 1.42 | 0.81 | 1.53 |
| | | CRO | 1.0 1.0 0.10 | 0.6 0.8 1.0 | - | 0.63 | 0.91 | 0.38 | **0.64** |
| | INJ | TUR | 0.8 1.0 0.50 | 0.6 0.6 0.6 | - | 4.84 | 2.40 | 1.53 | 2.92 |
| | | SUS | 0.6 1.0 0.00 | 0.8 0.8 - | - | 3.20 | 2.37 | 0.74 | 2.10 |
| | | CRO | 1.0 1.0 0.00 | 0.4 0.6 - | - | 5.96 | 2.34 | 2.54 | 3.61 |
| | PR | TUR | 0.6 1.0 0.70 | 1.0 0.6 0.6 | - | 3.86 | 1.90 | 0.63 | 2.13 |
| | | SUS | 0.4 1.0 0.10 | 1.0 0.8 0.6 | - | 2.09 | 2.63 | 0.41 | 1.71 |
| | | CRO | 0.8 1.0 0.03 | 0.4 0.8 0.2 | - | 2.36 | 0.93 | 0.45 | 1.25 |
| Adaptive penalty | 2-P | TUR | 0.8 1.0 1.0 | 0.8 0.6 0.2 | 50 | 3.98 | 2.20 | 1.12 | 2.43 |
| | | SUS | 0.4 1.0 0.03 | 1.0 0.8 1.0 | 90 | 1.45 | 1.15 | 0.46 | 1.02 |
| | | CRO | 1.0 1.0 0.01 | 1.0 0.8 0.8 | 90 | 0.18 | 0.14 | 0.07 | **0.13** |
| | INJ | TUR | 0.2 1.0 0.3 | 0.4 0.6 1.0 | 90 | 3.49 | 2.11 | 1.34 | 2.32 |
| | | SUS | 0.2 1.0 0.03 | 1.0 0.8 0.6 | 50 | 2.04 | 2.45 | 0.54 | 1.68 |
| | | CRO | 1.0 1.0 0.00 | 0.0 0.6 - | 30 | 5.01 | 2.56 | 2.35 | 3.31 |
| | PR | TUR | 0.8 0.8 0.5 | 0.6 0.6 1.0 | 70 | 3.49 | 1.03 | 0.77 | 1.76 |
| | | SUS | 0.6 1.0 0.3 | 1.0 1.0 0.2 | 90 | 2.06 | 0.99 | 0.35 | 1.13 |
| | | CRO | 0.6 1.0 0.03 | 0.4 1.0 1.0 | 70 | 1.89 | 0.45 | 0.55 | 0.96 |

**Table 5.** Collective average results for best EA configurations (2-point crossover + deterministic crowding selection) of both groups for all instances from all classes (* indicates that optimal solutions are unknown and gap to the best solution found by the EA is given). 95-percent confidence intervals for average gaps are given in brackets.

| Solutions feasibility | calibration networks | class I networks | class II networks | class III networks |
|---|---|---|---|---|
| | avg. gap (%) | avg. gap (%) | avg. gap (%) | avg. gap (%) |
| No infeasible solutions | **0.64** (±0.07) | **0.41** (±0.03) | **0.34** (±0.02) | **0.55*** (±0.08) |
| Adaptive penalty function | **0.13** (±0.07) | **0.08** (±0.01) | **0.18** (±0.02) | **0.06*** (±0.03) |

## 5.3 Results for all test instances

For each problem instance EA was executed 30 times and average gap was computed. In table 5 collective average results of best tuned EA configurations for all instances from all classes are displayed and compared to results from calibration phase. Results obtained during tuning procedures are consistent with those obtained for all test cases: better calibration results imply better overall results. EA configuration with adaptive penalty function produces extremely good overall results (average gaps of 0.08 and 0.19 percent) and is 0.2-0.5 percent better than EA without infeasible solutions. It can be seen that overall average gaps between algorithms for class I and II are smaller than gaps obtained for calibrated networks. It results from the fact that calibration was performed on larger instances (probably the hardest to obtain high-quality results) with 262-400 vertices while most test networks from class I and II had 100-200 nodes. Average gap between EAs is biggest for class III (large network of 908 cities). Average differences between two EA versions are statistically significant (confidence intervals don't overlap).

In table 6 results for all instances from class I are given for best EA configurations (with and without infeasible solutions). EA version with adaptive penalty is on average more than 0.3 percent better than EA without infeasible paths (results generated in similar execution times). The biggest difference between EAs was about 2 percent (pr144 network). The algorithm with adaptive penalty achieves optimal or nearly optimal solutions in all test cases and clearly outperforms GRASP and GRASPwPR metaheuristics (by 2.4 and 1.2 percent respectively).

In table 7 analogical results are given for instances from class II. EA with adaptive penalty performs better than its version without infeasible solutions (by almost 0.2 percent) and its execution times are shorter. The biggest difference between EAs is about 1.1 percent (cmt200B network). Both EAs clearly outperform GRASP and GRASPwPR (respectively by 1.7-1.9 and 3.1-3.3 percent).

In table 8 results for class III instances are presented. EA with penalty function is on average 0.5 percent better than EA without infeasible solutions and the biggest difference (1 percent) is obtained for longest paths (3000 km). EAs are on average 3-5 percent better than other metaheuristics. It can be seen that bigger gaps between algorithms are associated with larger test network (908 vertices).

**Table 6.** Detailed comparison of results of the best tuned EA configurations (2-point crossover + deterministic crowding) with results of GRASP. GRASPwPR and best known solutions (class I). Execution time is given in seconds.

| Instance | $EA_{AdaptivePenalty}$ | | | $EA_{NoInfeasibleSolutions}$ | | | GRASP | | GRASP PR | | Best solution |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | profit | gap (%) | time | profit | gap (%) | time | profit | gap (%) | profit | gap (%) | |
| kroA100 | 3180 | 0.03 | 1.1 | 3177.8 | 0.10 | 1.3 | 3135 | 1.45 | 3181 | 0.00 | 3181 |
| kroB100 | 3187.1 | 0.25 | 1.1 | 3191 | 0.13 | 1.3 | 3183 | 0.38 | 3191 | 0.13 | 3195 |
| kroC100 | 3043.3 | 0.02 | 1.5 | 3025.7 | 0.60 | 1.5 | 3044 | 0.00 | 3044 | 0.00 | 3044 |
| kroD100 | 3223.3 | 0.08 | 1.3 | 3222.3 | 0.11 | 1.7 | 3152 | 2.29 | 3212 | 0.43 | 3226 |
| kroE100 | 3310 | 0.00 | 1 | 3303.9 | 0.18 | 1.2 | 3260 | 1.51 | 3310 | 0.00 | 3310 |
| rd100 | 3470 | 0.00 | 1.2 | 3448.7 | 0.61 | 1.1 | 3449 | 0.61 | 3453 | 0.49 | 3470 |
| eil101 | 3668 | 0.00 | 1.2 | 3667.4 | 0.02 | 1.1 | 3596 | 1.96 | 3645 | 0.63 | 3668 |
| lin105 | 3577 | 0.00 | 1.6 | 3576.7 | 0.01 | 1.4 | 3577 | 0.00 | 3577 | 0.00 | 3577 |
| pr107 | 2681 | 0.00 | 1.2 | 2681 | 0.00 | 0.8 | 2681 | 0.00 | 2681 | 0.00 | 2681 |
| gr120 | 4223 | 0.00 | 1.3 | 4198.5 | 0.58 | 1.4 | 4138 | 2.01 | 4201 | 0.52 | 4223 |
| pr124 | 3840 | 0.00 | 2 | 3840 | 0.00 | 1.8 | 3840 | 0.00 | 3840 | 0.00 | 3840 |
| bier127 | 5375 | 0.02 | 3.4 | 5374.7 | 0.02 | 4.1 | 5154 | 4.13 | 5254 | 2.27 | 5376 |
| pr136 | 4221.2 | 0.04 | 2 | 4214.2 | 0.21 | 1.7 | 4170 | 1.26 | 4213 | 0.24 | 4223 |
| gr137 | 4274.7 | 0.38 | 2.2 | 4272.1 | 0.44 | 1.7 | 4255 | 0.84 | 4284 | 0.16 | 4291 |
| pr144 | 3989.1 | 0.12 | 2.5 | 3911.5 | 2.07 | 2 | 3902 | 2.30 | 3994 | 0.00 | 3994 |
| kroA150 | 4919 | 0.00 | 2 | 4916.8 | 0.04 | 2.3 | 4768 | 3.07 | 4915 | 0.08 | 4919 |
| kroB150 | 5017 | 0.00 | 1.8 | 5014.5 | 0.05 | 2.1 | 4967 | 1.00 | 5001 | 0.32 | 5017 |
| pr152 | 4193.4 | 0.06 | 1.9 | 4192.4 | 0.09 | 1.9 | 4094 | 2.43 | 4175 | 0.50 | 4196 |
| u159 | 5044 | 0.00 | 2.2 | 5028.3 | 0.31 | 2.3 | 4809 | 4.66 | 4987 | 1.13 | 5044 |
| rat195 | 5933 | 0.05 | 2.5 | 5895.1 | 0.69 | 2.9 | 5693 | 4.09 | 5693 | 4.09 | 5936 |
| d198 | 6537.9 | 0.02 | 3.9 | 6507.9 | 0.48 | 3.3 | 6347 | 2.94 | 6476 | 0.96 | 6539 |
| kroA200 | 6611.9 | 0.06 | 4 | 6583.3 | 0.49 | 3.3 | 6447 | 2.55 | 6551 | 0.98 | 6616 |
| kroB200 | 6596.2 | 0.01 | 2.5 | 6581.6 | 0.23 | 3.5 | 6357 | 3.64 | 6409 | 2.85 | 6597 |
| ts225 | 6807.5 | 0.07 | 3.6 | 6732.1 | 1.17 | 4.2 | 6701 | 1.63 | 6784 | 0.41 | 6812 |
| pr226 | 6690.7 | 0.00 | 6.5 | 6685 | 0.09 | 6.5 | 6375 | 4.72 | 6614 | 1.15 | 6691 |
| gil262 | 9146.8 | 0.13 | 6.4 | 9135.8 | 0.25 | 5.6 | 8847 | 3.41 | 8941 | 2.38 | 9159 |
| pr264 | 6657.8 | 0.12 | 3.9 | 6666 | 0.00 | 3.8 | 6666 | 0.00 | 6666 | 0.00 | 6666 |
| pr299 | 9091.9 | 0.17 | 6 | 9010 | 1.07 | 6.3 | 8645 | 5.07 | 8689 | 4.59 | 9107 |
| lin318 | 10902 | 0.55 | 8.4 | 10795.6 | 1.52 | 8.4 | 10074 | 8.10 | 10339 | 5.68 | 10962 |
| rd400 | 13534.4 | 0.15 | 14.6 | 13461.3 | 0.69 | 16.2 | 12365 | 8.78 | 12365 | 8.78 | 13555 |
| **Avg.** | **5431.5** | **0.08** | **3.2** | **5410.4** | **0.41** | **3.2** | **5256.4** | **2.49** | **5322.8** | **1.29** | **5437.2** |

**Table 7.** Detailed comparison of results of the best tuned EA configurations (2-point crossover + deterministic crowding) with results of GRASP, GRASPwPR and best known solutions (class II). Execution time is given in seconds.

| Instance | $EA_{AdaptivePenalty}$ | | | $EA_{NoInfeasibleSolutions}$ | | | GRASP | | GRASPwPR | | Best solution |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | profit | gap (%) | time | profit | gap (%) | time | profit | gap (%) | profit | gap (%) | |
| eil101A | 572 | 0.00 | 0.6 | 571.9 | 0.02 | 0.7 | 566 | 1.05 | 572 | 0 | 572 |
| cmt121A | 406.9 | 1.24 | 1 | 408.9 | 0.75 | 0.7 | 412 | 0 | 412 | 0 | 412 |
| cmt151A | 824 | 0.00 | 0.9 | 824 | 0.00 | 0.8 | 815 | 1.09 | 824 | 0 | 824 |
| cmt200A | 1202.8 | 0.18 | 1.7 | 1204.7 | 0.02 | 2 | 1145 | 4.98 | 1181 | 1.99 | 1205 |
| gil262A | 4508.9 | 0.02 | 1.9 | 4492.9 | 0.38 | 2.4 | 3916 | 13.17 | 4050 | 10.2 | 4510 |
| eil101B | 1049 | 0.00 | 1 | 1047.1 | 0.18 | 1.2 | 1024 | 2.38 | 1032 | 1.62 | 1049 |
| cmt121B | 708.9 | 0.85 | 1.3 | 712.7 | 0.32 | 1.5 | 699 | 2.24 | 707 | 1.12 | 715 |
| cmt151B | 1536.5 | 0.03 | 1.9 | 1535.7 | 0.08 | 2.1 | 1482 | 3.58 | 1528 | 0.59 | 1537 |
| cmt200B | 2196.5 | 0.07 | 5.5 | 2172.4 | 1.16 | 5.2 | 2073 | 5.69 | 2105 | 4.23 | 2198 |
| gil262B | 8449.6 | 0.08 | 6.3 | 8420.6 | 0.42 | 5.7 | 7946 | 6.03 | 8074 | 4.52 | 8456 |
| eil101C | 1336 | 0.00 | 1.9 | 1333.6 | 0.18 | 2.7 | 1295 | 3.07 | 1302 | 2.54 | 1336 |
| cmt121C | 1133.8 | 0.02 | 2.6 | 1129.5 | 0.40 | 2.1 | 1120 | 1.23 | 1125 | 0.79 | 1134 |
| cmt151C | 2001.9 | 0.05 | 4.3 | 1993.3 | 0.48 | 6 | 1965 | 1.9 | 1996 | 0.35 | 2003 |
| cmt200C | 2876.1 | 0.17 | 9 | 2873.3 | 0.27 | 10.9 | 2791 | 3.12 | 2824 | 1.98 | 2881 |
| gil262C | 11185.4 | 0.09 | 10.1 | 11153.6 | 0.37 | 13.3 | 10938 | 2.3 | 11046 | 1.33 | 11195 |
| **Avg.** | **2665.9** | **0.18** | **3.3** | **2658.3** | **0.34** | **3.8** | **2545.8** | **3.46** | **2585.2** | **2.08** | **2668.5** |

**Table 8.** Detailed comparison of results of the best tuned EA configurations (2-point crossover + deterministic crowding) with results of GRASP, GRASPwPR and best known solutions (class III). Execution time is given in seconds.

| Instance | $EA_{AdaptivePenalty}$ | | | $EA_{NoInfeasibleSolutions}$ | | | GRASP | | GRASPwPR | | Best solution |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | profit | gap | time | profit | gap(%) | time | profit | gap (%) | profit | gap (%) | |
| pl500 | 3735 | 0.11 | 1.2 | 3711.6 | 0.74 | 1 | 3637 | 2.73 | 3696 | 1.15 | 3739 |
| pl1000 | 7936.2 | 0.06 | 3.8 | 7927.6 | 0.17 | 3.3 | 7598 | 4.32 | 7801 | 1.76 | 7941 |
| pl1500 | 10379.4 | 0.04 | 5.5 | 10336.8 | 0.45 | 5.1 | 9576 | 7.78 | 9974 | 3.95 | 10384 |
| pl2000 | 12119.7 | 0.05 | 7.1 | 12092.1 | 0.28 | 7.4 | 11739 | 3.19 | 11739 | 3.19 | 12126 |
| pl2500 | 13574.2 | 0.04 | 11 | 13500.1 | 0.59 | 11.7 | 12654 | 6.82 | 12774 | 5.94 | 13580 |
| pl3000 | 14956.6 | 0.06 | 16.6 | 14805.1 | 1.07 | 17 | 14022 | 6.31 | 14222 | 4.97 | 14966 |
| **Avg.** | **10450.2** | **0.06** | **7.5** | **10395.6** | **0.55** | **7.6** | **9871.0** | **5.19** | **10034.3** | **3.49** | **10456** |

In figure 1 the best path generated for a network of Polish cities is illustrated. All major cities (except Bialystok) are included in the path. A comparison between runs of EAs with static penalty (parameter *k* is always equal to 1) and adaptive penalty is shown in figures 2 and 3 (network pr144). Without adaptation all paths in the population exceeds length limit in further generations. This is rare among tested benchmark instances and usually signals convergence to very fit solutions violating infeasibility border. On the other hand, EA version with adaptive penalty reduces number of infeasible solutions allowing for further improvement (*k* parameter increased to 1.6).

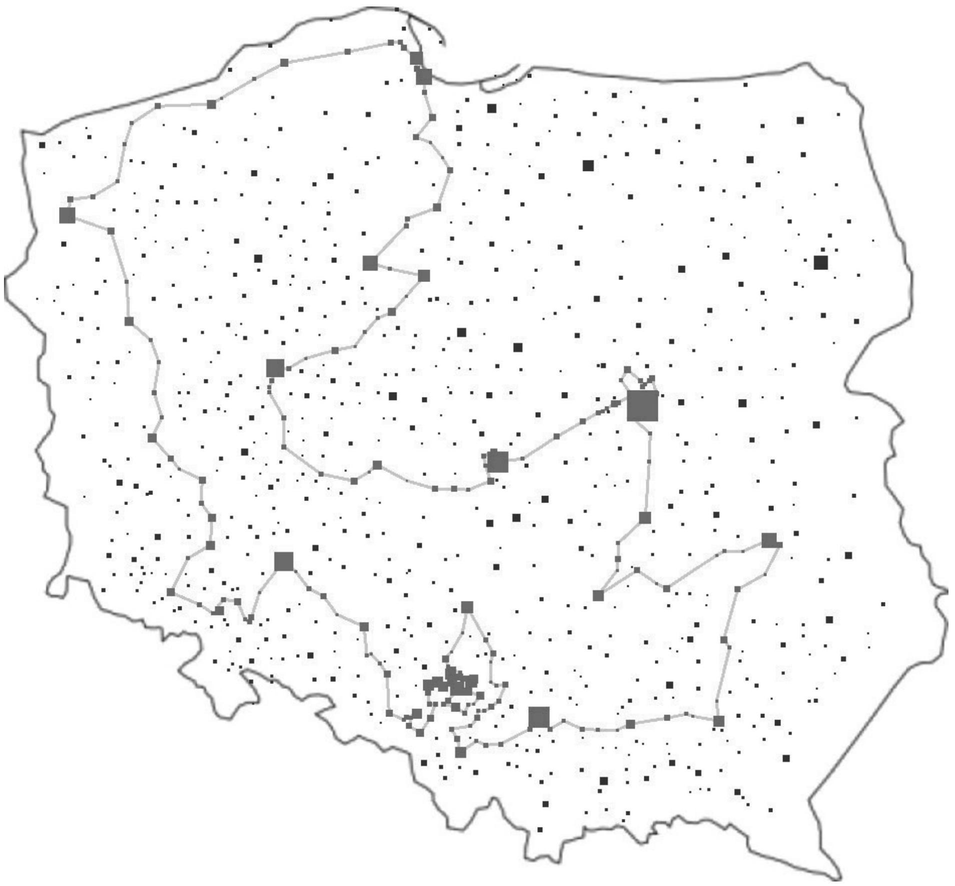


**Fig. 1.** The best path found for a network of Poland ($C_{max} = 3000$ km). The route starts and ends in Warsaw, it's length is 2999.81 km and profit is 14966 (almost 15 million of inhabitants in visited cities).

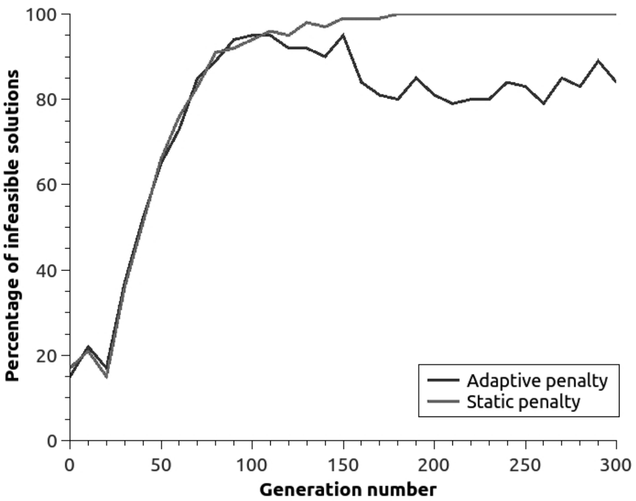


**Fig. 2.** EA runs with static and adaptive penalties (percentage of infeasible solutions in the population).

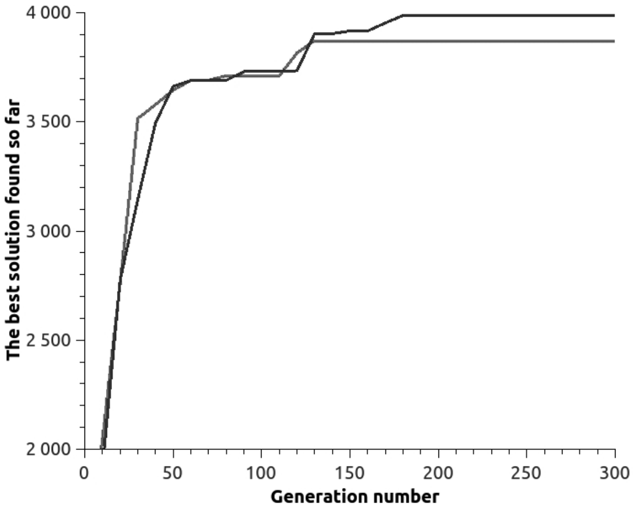


**Fig. 3.** EA runs with static and adaptive penalties (profit of best solution found).

## 6. Conclusions and further research

In the paper two strategies of dealing with solutions infeasiblity were compared for evolutionary algorithms solving the Orienteering Problem. Nine different algorithm configurations (varying in selection and crossover phases) were tuned and tested in two versions: with infeasible solutions (too long paths) in populations (adaptive penalty) and without them. Parameters of EAs were tuned with ParamsILS local search algorithm. It was found that all EA configurations with infeasible solutions outperformed their counterparts without too long paths in populations. The best EA configurations from both groups used 2-point crossover and deterministic crowding and the difference between them were on average about 0.2-0.5 percent (up to 1-2 percent in some cases). Best EA configuration with adaptive penalty obtained optimal or nearly optimal results for all test networks clearly outperforming other compared metehauristics (GLS, GRASP, GRASPwPR).

The author's further research is concentrated on OP versions more applicable to trip planning: the Time-Dependent Team Orienteering Problem with Time Windows (finding a multi-day tour in time-dependent graps i.e. public transport networks) and the Orienteering Problem with Hotel Selection (finding a tour divided into stages with acommodation in hotels).

### Acknowledgment

The author gratefully acknowledges support from the Polish Ministry of Science and Higher Education at the Bialystok University of Technology (grant S/WI/1/2014).

## References

[1] Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: The City Trip Planner: An expert system for tourists. Expert Systems with Applications, vol. 38(6), 6540-6546, 2011.

[2] Caserta, M., Voss, S.: A hybrid algorithm for the DNA sequencing problem. Discrete Applied Mathematics, vol. 163(1), 87-99, 2014.

[3] Golden, B., Levy, L., Vohra, R.: The orienteering problem. Naval Research Logistics, vol. 34, 307-318, 1987.

[4] Ostrowski, K.: Parameters tuning of evolutionary algorithm for the Orienteering Problem. Advances in Computer Science Research, vol. 12, 53-78, 2015.

[5] Tsiligirides, T.: Heuristic methods applied to orienteering. Journal of the Operational Research Society, vol. 35 (9), 797-809, 1984.

[6] Fischetti, M., Salazar, J., Toth, P.: Solving the orienteering problem through branch-and-cut. INFORMS Journal on Computing, vol. 10, 133-148, 1998.

[7] Gendreau, M., Laporte, G., Semet, F.: A branch-and-cut algorithm for the undirected selective traveling salesman problem. Networks, vol. 32(4), 263-273, 1998.

[8] Chao, I., Golden, B., Wasil, E.: Theory and methodology - a fast and effective heuristic for the orienteering problem. European Journal of Operational Research, vol. 88, 475-489, 1996.

[9] Tasgetiren, M.: A genetic algorithm with an adaptive penalty function for the orienteering problem. Journal of Economic and Social Research, vol. 4 (2), 1-26. 2001.

[10] Gendreau, M., Laporte, G., Semet, F.: A tabu search heuristic for the undirected selective travelling salesman problem. European Journal of Operational Research, vol. 106, 539-545, 1998.

[11] Vansteenwegen, P., Souffriau, W., Vanden Berghe, G. and Oudheusden, D.V.: A guided local search metaheuristic for the team orienteering problem. European Journal of the Operational Research, vol. 196(1), 118-127, 2009.

[12] Schilde, M., Doerner, K., Hartl, R., Kiechle, G.: Metaheuristics for the biobjective orienteering problem. Swarm Intelligence, vol. 3, 179-201, 2009.

[13] Souffriau, W., Vansteenwegen, P., Vanden Berghe, G. and Oudheusden, D.V.: A path relinking approach for the team orienteering problem. Computers and Operations Research, vol. 37, 1853-1859, 2010.

[14] Campos, V., Marti, R.,Sanchez-Oro, J., Duarte, A.: Grasp with Path Relinking for the Orienteering Problem. Journal of the Operational Research Society, vol. 156, 1-14, 2013.

[15] Koszelew. J., Ostrowski. K.: A Genetic Algorithm with Multiple Mutation which Solves Orienteering Problem in Large Networks. Computational Collective Intelligence. Technologies and Applications, vol. 8083, 356-366, 2013.

[16] Zabielski. P., Karbowska-Chilinska, J., Koszelew, J., Ostrowski, K.: A Genetic Algorithm with Grouping Selection and Searching Operators for the Orienteering Problem. Lecture Notes in Artificial Intelligence, vol. 9012, 31-40, 2015.

[17] Ostrowski, K., Karbowska-Chilinska, J., Koszelew, J., Zabielski, P.: Evolution-inspired local improvement algorithm solving orienteering problem. Annals of Operations Research, vol. 253(1), 519-543, 2017.

[18] Ostrowski, K.: Comparison of Different Graph Weights Representations Used to Solve the Time-Dependent Orienteering Problem. Trends in Contemporary Computer Science, Podlasie 2014, Bialystok University of Technology Publishing Office, 144-154, 2014.

[19] Ostrowski, K.: Evolutionary Algorithm for the Time-Dependent Orienteering Problem. Proceedings from the Computer Information Systems and Industrial Management : 16th IFIP TC8 International Conference : CISIM 2017 (eds. Khalid Saeed, Wladyslaw Homenda, Rituparna Chaki). Lecture Notes in Computer Science, vol. 10244, 50-62, 2017.

[20] Ostrowski, K.: An Effective Metaheuristic for Tourist Trip Planning in Public Transport Networks. Applied Computer Science, vol. 14(2), 2018.

[21] Sokolov, A., Whitley, D.: Unbiased tournament selection. Proceedings of Genetic and Evolutionary Computation Conference. ACM Press, 1131-1138, 2005.

[22] Baker, J.E.: Reducing Bias and Inefficiency in the Selection Algorithm. Proceedings of the Second International Conference on Genetic Algorithms and their Application, Hillsdale, New Jersey: L. Erlbaum Associates, 14-21, 1987.

[23] Mahfoud, S.W.: Crowding and preselection revisited. Proceedings of the 2nd International Conference on Parallel Problem Solving from Nature (PPSN II), Brussels, Belgium, 1992. Elsevier, Amsterdam, The Netherlands, 27-36, 1992.

[24] Hutter, F., Hoos, H.H., Leyton-Brown, K., Stutzle, T.: ParamILS: an automatic algorithm configuration framework. Journal of Artificial Intelligence Research, vol. 36, 267-306, 2009.

[25] Network of 908 cities in Poland. http://p.wi.pb.edu.pl/sites/default/files/krzysztof-ostrowski/files/polska908.txt. Accessed 1 November 2018.

# RÓŻNE METODY TRAKTOWANIA ROZWIĄZAŃ NIEDOPUSZCZALNYCH W ALGORYTMACH EWOLUCYJNYCH ROZWIĄZUJĄCYCH ORIENTEERING PROBLEM

**Streszczenie** Orienteering Problem (OP) należy do problemów optymalizacji kombinatorycznej i jest zdefiniowany na grafach ważonych. Celem OP jest znalezienie ścieżki o ograniczonej długości i maksymalnym łącznym proficie (zbieranym w wierzchołkach). Artykuł prezentuje porównanie różnych metod radzenia z rozwiązaniami niedopuszczalnymi (zbyt długimi ścieżkami) w algorytmach ewolucyjnych rozwiązujących OP. Grupa algorytmów ewolucyjnych (różniących się operatorami selekcji i krzyżowania) została przetestowana w dwóch konfiguracjach: z osobnikami dopuszczalnymi w populacji oraz bez nich. Wartości parametrów algorytmów zostały ustawione za pomocą automatycznej procedury kalibracji

(ParamILS). Wyniki wskazują, że obecność zbyt długich ścieżek w populacji może poprawić jakość rozwiązań. Prezentowana meta-heurystyka uzyskiwała rozwiązania optymalne lub bliskie optymalnym dla sieci testowych.

**Słowa kluczowe:** rozwiązania niedopuszczlne, algorytmy ewolucyjne, Orienteering Problem

# PREPROCESSING TECHNIQUES FOR ONLINE SIGNATURE VERIFICATION AND IDENTIFICATION

Arkadiusz Pień, Marcin Adamski

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

**Abstract:** Handwritten signature is a behavioral biometric that can be used for automatic signer verification and identification. Online signature, in addition to visual shape, incorporates dynamics of the writing process such as trajectory, velocity and additional characteristics such as pen pressure and angles. While there are many approaches to online signature verification proposed in the literature, only few works related to preprocessing and its effect on the system performance. In this work selected preprocessing techniques were investigated such as: normalization, noise filtering and resampling. The evaluation was carried out in verification and identification tasks based on DTW distance measure and signatures from SVC2004 database.

**Keywords:** online signature, signature preprocessing

## 1. Introduction

Handwritten signature is one of the behavioral biometric traits that is widely used in all parts of the world. Compared with physical traits such as finger veins or iris image it has drawbacks that include low permanence and ease of producing a forgery. However, due to its widespread usage it has been a subject of intensive research and gained a lot of interest in commercial institutions. The ongoing development of new algorithms and methods allowed to lower the error rates of automatic signature verification to levels comparable with the results obtained for physical biometrics and opened the way for potential applications [4].

During data acquisition handwritten signatures are collected for further processing. There are two ways in which data can be acquired: offline – where the input of the system are static images of handwritten signatures; online – registers the act of signing that includes both the image and dynamics of writing. Due to specified nature

Advances in Computer Science Research, vol. 14, pp. 163-176, 2018. DOI: 10.24427/acsr-2018-vol14-0010

of data, offline and online signatures usually require different methods at each stage of the biometric system, however the stages of the system are similar.

The architecture of signature recognition system does not differ much from a typical biometric system. Its main stages are: data acquisition, preprocessing, feature extraction and classification (Fig. 1). Data acquisition is the process of registering data using particular type of input device . The preprocessing is responsible for preparing raw input to feature extraction process. Methods used at this stage may perform tasks such as: normalization, resampling, noise filtering.

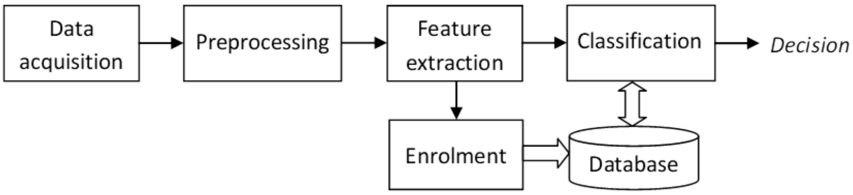


**Fig. 1.** Biometric signature verification and identification system

The feature extraction methods are responsible for constructing the feature vector that is passed to classification stage. The classification is used for one of the two tasks: identification and verification. The aim of verification is to decide whether the given biometric sample is genuine or forged. During the identification the systems finds individual whose signature best matches given sample.

While are there many approaches to automatic signature verification and identification proposed in the literature, few works related to signature preprocessing and its effect on system performance. To address it, this investigation is focused on preprocessing techniques and evaluation of their usefulness in signature verification and identification tasks. The evaluation is carried out using signature verification and identification system based on Dynamic Time Warping distance measure [9].

## 2. Online signature acquisition

In order to register online characteristics of the signing process a special input device is necessary. Dynamic data can be recorded by means of PC tablets [1], specialized signature pads [2] or cameras [8] where the trajectory of the signature is traced in video sequence. Some recent works investigate possibility of signature registration with mobile devices such as smartphones and mobile tables [6,7]. Widespread usage of these devices makes them interesting alternative for specialized signature pads. However, the lower quality of acquired data and lack of important features (ex. pressure) may lead to increased error rates [6] and needs to be addressed.

Basic dynamic data gathered at the time of signing can contain the following parameters (Fig. 2): X, Y coordinates, pressure (P), altitude (AL) and azimuth (AZ). The coordinates X and Y determine the position of the pen tip inside the controlled area where the signing process is being traced. The pressure parameter describes the pen pressure inflicted on the tablet surface. The altitude is the angle between the pen and the surface. The azimuth denotes the angle between the projection of the pen onto the writing surface and the X coordinate axis.

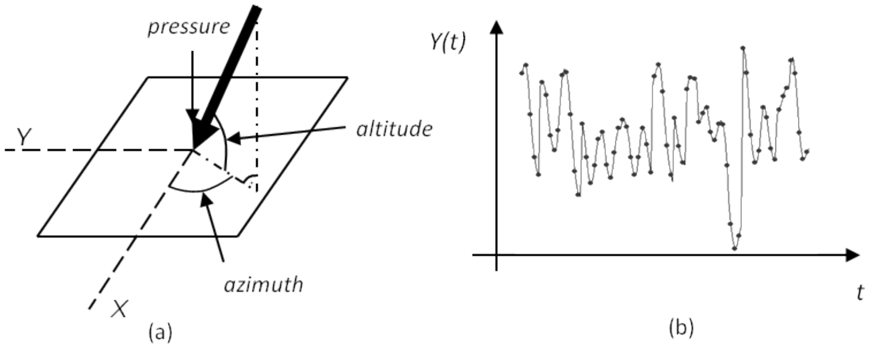


**Fig. 2.** Online signature parameters (a) and example of Y samples collected during the signing process (b)

Dynamic information acquired during online registration is very important because it allows to increase the system resistance to forgeries. Imitation of pen pressure, pen angles and dynamics of drawing signature is much more difficult than just copying a signature image. These dynamic parameters are also called hidden because it is impossible to precisely reconstruct their characteristics given only the image of a genuine signature. Another advantage when using online data compared to offline is that it is much easier to analyze – there is no need to extract a signature from complex background or deal with artifacts resulted from poor quality of scans.

## 3. Preprocessing techniques

There are various preprocessing techniques that may be used for online data. In this study the following types of preprocessing methods were investigated: normalization, filtering, resampling and component merging

### 3.1 Normalization

Raw data from acquisition device usually has range and precision dependent on a particular hardware. In addition, different characteristics have distinct units and scale. Signatures may be also given at arbitrary or fixed positions on writable surface depending on the constraints imposed by the application. In order to standardize ranges of values the normalization is applied to input data. However, the normalization process, may also result in loosing information important to distinguish genuine from forged signatures. In this work we experimented with the following techniques

**Position scaling** scales X and Y values according to equation (1).

$$x_i^N = K\frac{x_i}{M},\ y_i^N = K\frac{y_i}{M},\ M = \sqrt{\sum_{i=1}^{n} x_i^2 + y_i^2} \tag{1}$$

**Position centroid translation** translates centroid of a signature to the origin of coordinate system (2).

$$x_i^N = x_i - \bar{x},\ y_i^N = y - \bar{y},\ \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,\ \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2}$$

**Position standarization** translates centroid of a signature to the origin of coordinate system (3) and scales by standard deviation.

$$x_i^N = \frac{x_i - \bar{x}}{\sigma_x},\ y_i^N = \frac{y_i - \bar{y}}{\sigma_x} \tag{3}$$

During experiments we also performed centroid translation and standardization for complete set of input characteristics, namely for: X, Y, P, AL, AZ.

### 3.2 Filtering

The aim of filtering is to remove noise present in the signal. Main sources of such noise are instability of device during signing and noise introduced by input device. In this work we used three techniques for noise filtering [3].

**Median filter** replaces each sample with median value computed over window of length W as given by (4).

$$c_i^N = median(i - \lfloor W/2 \rfloor, ..., i + \lfloor W/2 \rfloor + 1) \tag{4}$$

**Average filter** replaces each value with the average computed over window of size W.

$$c_i^N = \frac{\sum\limits_{j=i-\lfloor W/2 \rfloor}^{i+\lfloor W/2 \rfloor} c_j}{W},\ i = \lfloor W/2 \rfloor + 1 ... n - \lfloor W/2 \rfloor \tag{5}$$

**Gaussian filter** replaces each value with weighted sum where weight coefficients are computed based on Gaussian distribution (6)

$$c_i^N = \sum_{j=-2\sigma}^{2\sigma} w_j c_{i+j},\ w_k = \frac{e^{-\frac{k^2}{2\sigma}}}{\sum\limits_{i=-2\sigma}^{2\sigma} e^{-\frac{i^2}{2\sigma}}} \tag{6}$$

### 3.3 Resampling

Resampling results in increasing or decreasing number of samples acquired by input device. Downsampling procedure may be implemented by selecting every k-th sample from input signal, resulting in reduction of frequency k times. The need for downsampling my may arise from Nyquist criteria. According to studies on dynamics of handwriting [5] the cut-off temporal frequency of the signing process is below 20Hz. This reduces the number of samples without losing information. Nyquist frequency in this case requires only 40 samples per second to retain all important components of writing parameters. Upsampling requires "inventing" new samples using interpolation. Upsamling may considered when the acquisition frequency of device is too low. For upsampling two interpolation techniques have been investigated.

**Linear interpolation in time domain** with samples interpolated at equal intervals in time to preserve time characteristics of signal (7).

$$L = \left\lfloor \frac{f_{req}}{f_{act}} \right\rfloor - 1 \tag{7}$$

where L is number of points to be added to the signal between consecutive samples, $f_{req}$ is required frequncy, $f_{act}$ is actual frequency.

**Linear interpolation in space domain** adds additional points based on position. It does not preserve time characteristics (8).

$$LP = \left\lfloor \frac{d}{D} \right\rfloor \tag{8}$$

where LP is number of points required to be added to signal between consecutive samples, d is distance between those points, D is threshold distance above which proportional number of points will be added.

### 3.4 Component merging

The signature may be splitted into separate curves that are separated by pen-up and pen-down events. This happens due to the signer lifting pen between drawing different parts of the signature. Similarly to upsampling techniques, two methods for adding artificial samples between up/down events were investigated – *merging in time domain* and *merging in space domain*. The interpolation method is the same as previously described approach, but in this case it is only used to add points between samples corresponding to pen-up/down events.

## 4. Classification

In order to evaluate different preprocessing techniques a signature recognition system was implemented. In the literature one may find many approaches to signature classification, among them some of the most successful techniques are based on Dynamic Time Warping (DTW) distance measure [4]

DTW method allows for modeling the time-axis fluctuation with nonlinear warping function [9]. The timing differences are eliminated by warping the characteristics of the signatures in such a way that the optimal alignment is achieved. DTW algorithm defines a measure $D'(R', S')$ between two sequences $R^{'}$ and $S^{'}$ (9):

$$R^{'} =< r^{'}_1, r^{'}_2, ..., r^{'}_M >,\ S^{'} =< s^{'}_1, s^{'}_2, ..., s^{'}_N > \tag{9}$$

The distance $D'$ is defined using the following (10) and (11):

$$D'(R', S') = D^R(N, M) \tag{10}$$

$$D^R(i, j) = \min \left\{ \begin{array}{c} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{array} \right\} + d(r^{'}_i, s^{'}_j) \tag{11}$$

where $i = 1..M, j = 1..N$. $d(r^{'}_i, s^{'}_j)$ can be distance measure such as Euclidian.

The calculations are carried out using dynamic programming. The key part of this algorithm is forming the so called cost matrix g. Its elements are cumulative distances computed as the sum of distances with one of the cumulative distances being found in earlier iterations (12).

$$g(i, j) = d(r^{'}_i, s^{'}_j) + \min\{g(i-1, j), g(i, j-1), g(i-1, j-1)\} \tag{12}$$

The cost matrix enables to find a warping path that represents the best alignment and minimizes the overall distance given by the recursive function of (11).

## 5. Evaluation procedure

In this investigation we used SVC2004 signatures database [10] publicly available for research. There two datasets available:

- Task1 – samples in this category have only trajectory data : X,Y, pen up/down state.
- Task2 – samples contain all characteristics: X,Y,P,AL,AZ, pen up/down state.

Both sets contain genuine and forged signatures of 40 individuals. All signatures were collected using Intuos tablet with sampling frequency of 100Hz. In this study the signatures from Task2 dataset were used.

During experiments two classification tasks were considered: identification and verification. The experiments were repeated 10 times using cross-validation scheme. The final results are the average values over 10 trials. During each repetition both the training and testing sets were selected at random.

Identification task was carried out using DTW distance measure with k-NN algorithm. The training set contained three genuine signatures per individual, the test set consisted of 12 genuine signatures per individual. Final evaluation was based on percent of properly classified signatures.

In the verification, the minimal distance between tested and reference signatures from training set for particular individual was computed, and the final decision (accept or reject) was based on comparison with threshold value. The training (reference) set consisted of four genuine examples per individual. The test set was constructed from 10 genuine and forged examples per subject. The verification was performed separately for simple forgeries (signatures of other users used as forgeries) and skilled forgeries (attempts to forge user signature by individuals who had access to genuine samples and time to train). Equal Error Rate (EER) was used as a performance measure.

## 6. Results

As a baseline for comparison, first the verification and identification performance was computed without using preprocessing techniques. Table 1. presents obtained results.

**Table 1.** Baseline results without preprocessing

| Verification EER [%] | | Identification [%] |
|---|---|---|
| Skilled forgeries | Simple forgerie | |
| 17.82 | 8.45 | 91.5 |

As could be expected, verification of skilled signatures has higher error than simple forgeries. It is important to note that in this investigation our aim was not to develop complete system with lowest error, but to compare different preprocessing techniques.

### 6.1 Normalization

Table 2 presents impact of selected normalization techniques on system performance. As can be seen normalization techniques may significantly improve verification and identification rates (improvement over baseline given in Table 1 is grayed out). For identification the highest result was achieved for position centroid translation. This method also resulted in lower error for skilled and simple forgeries verification as compared to baseline system. Best performance for skilled signature was obtained for position standarization, however for other task this technique performed worse than baseline.

**Table 2.** Results of normalization preprocessing techniques

| Method | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| Position scaling | 12.86 | 10.56 | 86.75 |
| Position centroid translation | 16.38 | **5.70** | **99.81** |
| Centroid translation (for all parameters) | 22.23 | 6.02 | 99.65 |
| Position standarization | **12.69** | 10.61 | 86.73 |
| Standarization (for all parameters) | 18.26 | 9.03 | 98.06 |

## 6.2 Filtration

The results of preprocessing using filters are given in Tables 3, 4 and 5.

**Table 3.** Results of median filtering

| Window size | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| 3 | 17.62 | **8.18** | 90.38 |
| 4 | 17.71 | 8.73 | 90.56 |
| 6 | 17.36 | 9.00 | 90.25 |
| 8 | **16.69** | 9.85 | 88.87 |
| 10 | 17.00 | 10.59 | 87.96 |
| 20 | 17.62 | 9.76 | 86.85 |
| 40 | 19.82 | 9.46 | 88.42 |

As can be seen from presented data, improvement over unfiltered input has been achieved mostly for skilled forgery verification but is less significant compared to normalization. Best result with median filtering was obtained for window of size 8. The average filter reported highest improvement with window size equal 6. In Gaussian the lowest error occurred for mask size of 14. In case of simple forgeries minor improvement occurred only with median at filter size of 3 and average filter of sizes 5 and 6. There were no enhancement for identification task under any of the investigated configurations. As a conclusion the application of filtering should be considered carefully, because it may decrease system performance depending on the type of task.

**Table 4.** Results of average filtering

| Window size | Verification EER % | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| 2 | 17.08 | 8.86 | 90.35 |
| 3 | 17.71 | 9.88 | 90.23 |
| 4 | 17.13 | 9.10 | 90.60 |
| 5 | 16.80 | 8.44 | 88.67 |
| 6 | **15.26** | **7.71** | 86.92 |
| 8 | 16.96 | 9.60 | 88.65 |
| 20 | 17.86 | 9.78 | 88.71 |
| 40 | 20.10 | 9.98 | 88.77 |

**Table 5.** Results of Gaussian filtering

| Window size | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| 2 | 17.14 | 9.08 | 90.04 |
| 3 | 16.90 | 9.09 | 90.73 |
| 4 | 17.05 | 9.78 | 88.38 |
| 5 | 16.80 | 9.80 | 87.10 |
| 6 | 16.94 | 9.27 | 87.29 |
| 7 | 16.24 | 10.36 | 87.35 |
| 14 | **15.80** | 11.58 | 81.00 |
| 15 | 16.49 | 10.84 | 79.54 |
| 20 | 17.07 | 11.52 | 78.71 |
| 30 | 18.03 | 11.85 | 72.23 |

### 6.3 Resampling

The effects of resampling in time domain by downsampling and upsampling using linear interpolation are shown in Table 6. The input frequency of acquired data is

**Table 6.** Results of resampling in time domain

| Requested frequency [Hz] | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| 50 (downsampling) | **16.75** | 8.80 | 91.08 |
| 25 (downsampling) | 18.63 | 8.37 | **92.00** |
| 20 (downsampling) | 18.58 | 9.16 | 91.67 |
| 10 (downsampling) | 21.80 | 10.39 | 89.36 |
| 200 (upsampling) | 17.53 | **8.07** | 89.40 |

100Hz. As can be seen, downsampling to 50 or 25 Hz does not significantly decrease system performance. Moreover, for identification and verification slight improvement over raw input can be noticed. This may be attributed to the fact that 40Hz sampling is sufficient to reconstruct most of the frequencies present in handwriting, therefore part of the samples at higher frequencies are redundant. Reduction of frequency also decreases amount of computations required to process a signature. However, as can be seen in Table 6, reducing frequency below certain limit increases verification and identification errors. Upsampling to 200Hz gives slight improvement in verification process but also increases computational cost.

**Table 7.** Results of resampling in space domain

| Method | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| interpolation with D=50 | 21.01 | 9.34 | **92.19** |
| interpolation with D=100 | 19.74 | **8.36** | 90.79 |
| downsampling to 50Hz and interpolation with D=100 | 20.49 | 8.68 | 91.69 |
| downsampling to 25Hz and interpolation with D=100 | 20.90 | 9.57 | 90.25 |

Table 7 shows the results of resampling that is based on distance between points that removes time dependencies between samples. In the last two rows the interpolation process was preceded by downsampling to replace removed samples with distance based interpolated points. By comparison with baseline one can notice small improvement in identification and simple forgeries verification. However, the verification of skilled forgeries worsened in all cases.

### 6.4 Component merging

Evaluation of component merging method with interpolation in space domain is given by Table 8. As can be seen, both the verification and identification may benefit from this technique. The parameter D denotes required distance between interpolated points, therefore controls the number of synthetic points added through interpolation. If D is small (number of added points is larger) the system performance decreases in all of the tasks. However starting from D=200, improvements start to be visible. Most of the positive effects happen in identification and skilled forgery verification.

**Table 8.** Results of component merging in space domain

| D | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| 25 | 22.53 | 12.64 | 87.41 |
| 50 | 19.49 | 9.64 | 89.41 |
| 100 | 18.80 | 9.21 | 90.85 |
| 200 | 18.38 | 8.65 | 91.52 |
| 300 | 17.73 | 8.26 | 91.42 |
| 600 | **17.34** | 8.80 | 91.52 |
| 650 | 17.80 | 8.91 | 90.60 |
| 800 | 17.58 | 8.60 | **92.44** |
| 900 | 17.50 | **8.07** | 91.79 |
| 1000 | 17.83 | 8.50 | 91.29 |

**Table 9.** Results of component merging in time domain

| Requested frequency [Hz] | Verification EER [%] | | Identification [%] |
|---|---|---|---|
| | Skilled forgeries | Simple forgeries | |
| 200 | 20.01 | 12.34 | 90.71 |
| 100 | 18.37 | 10.15 | 91.08 |
| 50 | 17.85 | 8.49 | 90.92 |
| 33 | 17.49 | 9.20 | 91.73 |
| 20 | 17.22 | 8.77 | 91.06 |
| 16.67 | **17.07** | 8.53 | **92.35** |
| 14.26 | 17.47 | **8.37** | 92.02 |
| 11.11 | 17.20 | 8.73 | 90.83 |

Table 9 shows output of the system for component merging based on interpolation in the time domain. In this case the number of added points is controlled through requested frequency. Similarly to previous method if the number of generated points is high (frequency is high) the performance drops in both tasks. For lower interpolation frequencies the system performance may improve.

## 7. Conclusions

In this work we have assessed selected preprocessing techniques in online signature verification and identification tasks. As the results show, the preprocessing techniques can have significant influence on verification error and identification accuracy. The normalization techniques that may be recommend are position scaling, position centroid substraction and standarization. As for the filtering the investigated methods performed slightly better for skilled forgeries verification, however in other tasks they had negative effect. One may also consider downsampling data to 50Hz or even 25Hz to benefit from reduced computational cost. Merging components also seem to be promising. However, if too many samples will be interpolated the system accuracy may decrease. The results have been obtained for DTW based system. Further investigation may include assessment with different classifiers and combining different preprocessing techniques.

## References

[1] Alonso-Fernandez, F., Fierrez-Aguilar, J., del-Valle, F., and Ortega-Garcia, J.: On-Line Signature Verification Using Tablet PC, 4th International Symposium on Image and Signal Processing and Analysis, pp. 245-250, 2005.

[2] Garcia-Salicetti, S., Beumier, C., Chollet, G., Dorizzi, B., Leroux les Jardins, J., Lunter, J., et al.: Biomet: A Multimodal Person Authentication Database Including Face, Voice, Fingerprint, Hand and Signature Modalities, Lecture Notes in Computer Science, vol. 2688, pp. 845–853, 2003.

[3] Gonzalez, R. C. and Woods, R. E.: Digital Image Processing, 3rd Edition, Prentice-Hall, Inc., 2007.

[4] Impedovo, D. and Pirlo, G.: Automatic Signature Verification: The State of the Art, IEEE Transactions on Systems, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 5, pp. 609-635, 2008.

[5] Lorette, G. and Plamondon, R: Dynamic Approaches to Handwritten Signature Verification, Computer Processing of Handwriting, ed: World Scientific, pp. 21-47, 1990.

[6] Martinez-Diaz, M., Fierrez, J., and Ortega-Garcia, J.: Automatic Signature Verification on Handheld Devices, Multimodality in Mobile Computing and Mobile Devices: Methods for Adaptable Usability, IGI Global, pp. 321-338, 2010.

[7] Mendaza-Ormaza, A., Miguel-Hurtado, O., Blanco-Gonzalo, R., and Diez-Jimeno, F.-J.: Analysis of handwritten signature performances using mobile devices, 45th IEEE International Carnahan Conference on Security Technology, pp. 1-6, 2011.

[8] Munich, M. E. and Perona, P.: Visual Identification by Signature Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 2, pp. 200-217, 2003.

[9] Sakoe, H. and Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 26, no. 1, pp. 43-49, 1978.

[10] Yeung, D.-Y., Chang, H., Yimin, X., George, S., Kashi, R., Matsumoto, T., et al.: SVC2004: First International Signature Verification Competition, International Conference on Biometric Authentication, Hong Kong, pp. 16-22, 2004.

# METODY WSTĘPNEGO PRZETWARZANIA DLA WERYFIKACJI I IDENTYFIKACJI PODPISU DYNAMICZNEGO

**Streszczenie** Podpis odręczny jest behawioralną cechą biometryczna która umożliwia automatyczną weryfikację i identyfikację autora podpisu. Podpis dynamiczny, oprócz informacji o kształcie, zawiera również dane dotyczące dynamiki składania podpisu takie jak trajektoria kreślenia, prędkość, zmiana nacisku i kątów nachylenia pióra. W literaturze można znaleźć wiele podejść do automatycznej weryfikacji podpisu, brakuje jednak prac z szerszą analizą metod wstępnego przetwarzania i oceną ich wpływu na poprawność pracy całego systemu. W niniejszej pracy zbadano wybrane techniki wstępnego przetwarzania takie jak: normalizacja, filtracja, próbkowanie oraz oceniono ich użyteczność w procesie weryfikacji i identyfikacji podpisu. W badaniach wykorzystano system bazujący na mierze odległości Dynamic Time Warping. Eksperymenty przeprowadzono na podpisach dynamicznych z bazy SVC2004.

**Słowa kluczowe:** podpis dynamiczny, wstępne przetwarzanie

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

# EMPIRICAL COMPARISON OF METHODS OF DATA DISCRETIZATION IN LEARNING PROBABILISTIC MODELS

Michał Wójciak, Anna Łupińska–Dubicka

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

**Abstract:** Very often statistical method or machine learning algorithms can handle discrete attributes only. And that is why discretization of numerical data is an important part of the pre–processing. This paper presents the results of the problem of data discretization in learning quantitative part of probabilistic models. Four data sets taken from UCI Machine Learning Repository were used to learn the quantitative part of the Bayesian networks. The continuous variables were discretized using two supervised and two unsupervised discretization methods. The main goal of this paper was to study whether method of data discretization in given data set has an influence on model's reliability. The accuracy was defined as the percentage of correctly classified records.

**Keywords:** discretization, continuous feature, probabilistic models, Bayesian networks, classification

## 1. Introduction

Often data are given in the form of continuous values. If their number is huge, building a proper model for such data can be difficult. Moreover, many data mining algorithms operate only in discrete variable space. For instance, probabilistic models such Bayesian networks, require discrete values for their nodes. In addition, discretization also can work as a variable (feature) selection method that can significantly impact the performance of classification algorithms used in the analysis of high–dimensional data.

This paper presents the results of data discretization in problem of learning quantitative part of probabilistic models, in particular one of their prominent members – Bayesian networks. One of the most important features of Bayesian networks is the

Advances in Computer Science Research, vol. 14, pp. 177-192, 2018. DOI: 10.24427/acsr-2018-vol14-0011

fact that they provide an elegant mathematical structure for modeling complicated relationships among random variables while keeping a relatively simple visualization of these relationships.

The experiments involved learning the conditional probability distribution of models created on the basis of four data set taken from UCI Machine Learning Repository [23]: *Banknote authentication*, *Heart disease*, *Image segmentation* and *Abalone*. The main purpose of this article was to study whether method of data discretization in given data set has an influence on model's reliability. The accuracy was defined as the percentage of correctly classified records.

The remainder of this paper is structured as follows. Section 2. explains the problem of data discretization and shortly outlines the methods of it. Section 3. explains the basic concepts of Bayesian networks. Section 4. introduces selected data sets and presents created Bayesian network models. Section 5. presents the results of experiments conducted on data sets with implemented methods of data discretization. Section 6. concludes the paper and indicates possible directions for further research.

## 2. Data discretization

Discretization of numerical data is an important part of the pre–processing, necessary in typical processes of knowledge discovery and data mining. Transforming continuous attribute values into their discrete counterparts enables further analysis using data mining algorithms, such as learning parameters of probabilistic models (existing algorithms mainly assume discrete variables for nodes). Even in the absence of such a requirement, discretization allows accelerating the process of data mining and increasing the accuracy (accuracy) of predictions (classification) [3].

According to the surveys [3,6,8,11] and finally the advanced review [17] many different discretization algorithms have been proposed in the last two decades. Their authors used a different approach, derived from statistics, machine learning, information theory and logic. In order to be able to better understand these issues, the comparative criteria used should be taken into account [8,17]:
*Local* or *global* discretization – in the case of global discretization, the entire problem space is considered at the same time. Local discretization at the moment solves only a selected subproblem. The division is made on the basis of a limited amount of information.
*Supervised* or *unsupervised* – during supervised discretization the decision (class) of each of the objects is taken into account. The main premise of supervision is to separate instances having different decisions from each other. If the method does not use information given by classes, it is known as unsupervised. The advantage of unsu-

pervised methods is the ability to use them to discretize databases that do not have a decision attribute.

*Static* or *dynamic* discretization – the static method of attribute dependence is not taken into account. During one discretization cycle, the maximum number of intervals for a given attribute is obtained, regardless of the others. Dynamic methods simultaneously consider cutting for many features, which allows the use of high–level dependencies.

Due to the multitude of existing discretization methods, there is a need to introduce quality assessment criteria [8]:

*Number of intervals* – the fewer intervals, the simpler the result table. It can be seen that the problem of minimizing the number of intervals is synonymous with minimizing the number of cuts.

*Number of inconsistencies* – it would be best if discretization did not introduce additional inconsistencies over those contained in the input database. Otherwise significant information can be lost.

*Accuracy of predictions* – defines how discretization helps to improve predictions. It should be emphasized that this criterion depends on the classification method and the procedure of conducting the experiment. It should also be emphasized that only the first two criteria are directly measurable. The accuracy of predictions is a function of both discretization and the classification algorithm. These criteria do not indicate unambiguously which of the tested methods is the best. Depending on the chosen base and the expected results, the weight of each criterion may fluctuate. What's more, there is no discretization method that would have an advantage over all criteria at the same time.

As mentioned before, there are several method to discretize continuous variables. Below short description of four of them, used in this paper, is presented.

**OneR algorithm** [3,6] is a supervised method of discretization, using information about the class. Values that have been previously sorted are divided into intervals whose limits are set based on both continuous values and class labels. There is an assumption that each of the intervals must contain a minimum number of examples equals to $k$, where $k$ is usually set to six. This assumption does not apply to the last range, which contains other, ungrouped examples. The exception occurs when the next attribute has the same class as most examples in a given range.

**Chi merge algorithm** [6,12,13,17] is a simple, supervised algorithm that uses the $\chi^2$ statistic to discretize numeric attributes. It checks each pair of adjacent rows in order to determine if the class frequencies of the two intervals are significantly different. It tests the hypothesis that the two adjacent intervals are independent. If the hypothesis is confirmed the intervals are merged into a single interval, if not, they remain separated.

**Equal–Width Discretization (EWD) algorithm** [6,7,9,14,17,18] belongs to class unsupervised methods. The main assumption of this algorithm is to divide the data set into $k$ intervals determined by the user of the algorithm. It first finds the minimum and maximum values of every variable, $X_i$, and then divides this range into a number, $k$, of user–specified, equal–width intervals. The discussed algorithm has one fundamental disadvantage – in most cases all elements of the data set will be unevenly distributed in groups. In extreme cases, even empty sets may be created or one set having more elements than all the others combined. Therefore, it is very important to properly adjust the $k$ parameter to minimize this span.

**Equal–Frequency Discretization (EFD) algorithm** [1,6,7,9,14,17], like **EWD**, is a representative of the unsupervised discretization methods. It determines the minimum and maximum values of the variable $X_i$, sorts all values in ascending order, and divides the range into a user–defined number of intervals $k$, in such a way that every interval contains the equal number of sorted values. Each of these intervals contains $N/k$ elements, where $N$ means the total number of $X_i$ variable values. This method eliminates the possibility of disproportionate intervals because the entire interval $< X_{min}; X_{max} >$ containing specific values is divided into compartments in terms of a specific number of elements, not on the basis of ranges of values.

## 3. Bayesian Networks

Bayesian networks (also knows as belief networks or causal networks, BNs) [16] are a special case of probabilistic models. They have found many practical application over the years, among them the best known and probably the most successful are decision support systems. Bayesian networks offer natural mechanism for reasoning under uncertainty, when we do not have access to the full knowledge of the analyzed phenomenon. They allow for easy and readable representations of the actual relationships, which makes it easier to apply the real relationships. Furthermore, Bayesian networks enable a combination of *a priori* knowledge and collected data.

Formally, a Bayesian network $\mathcal{B}$ is a pair $<\mathcal{G},\Theta>$, where $\mathcal{G}$ is an acyclic directed graph in which nodes represent random variables $X_1,\ldots,X_n$ and edges represent direct dependencies between pairs of variables [16]. $\Theta$ represents the set of parameters that describes the probability distribution for each node $X_i$ in $\mathcal{G}$, conditional on its parents in $\mathcal{G}$, i.e., $P(X_i|Pa(X_i))$. Often, the structure of the graph is given as a causal interpretation, convenient from the point of view of knowledge engineering and user interfaces. BNs allow for computing probability distributions over subsets of their variables conditional on other subsets of observed variables. The joint probability distribution is represented as follows:

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

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

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

Note that in the Equation 1, probability of a random variable $X_i$ depends only on the states of its parents. This simplification resulting from the assumption of conditional independence of variable, allows to represent the joint probability distribution more compactly. This is particularly significant in the case of large–scale networks with a large number of variables. If a network consists of $n$ binary nodes, then the full joint probability distribution would require storing $2^n$ values. Using the factored form would require $n2^k$, where $k$ is the maximum number of parents of a node.

## 4. Data sets and Models

For the purpose of this work, the UCI Machine Learning Repository [23] has been searched and four data set containing continuous attributes were chosen: *Banknote authentication*, *Hearth disease*, *Statlog (image segmentation)*, and *Abalone*. Then, for particular data set the probabilistic model were constructed. The graphical structure of a Bayesian network represents a set of domain variables and relationships among them.

### 4.1 Banknote authentication

The *Banknote authentication* [24] data set is a collection of data extracted from images of original and fake banknotes. The images were created using an industrial camera used to control the print quality. The resulting images are 400 x 400 pixels and 660 dpi. To extract interesting data from these images, a wavelet transformation was used. The data set contains four continuous variable and one decision class. The data set contains 1372 objects, however, some of them were removed due to the fact that they contained missing elements, which could significantly lead to incorrect results of classification quality. Figure 1 presents the Bayesian network created on the basis of *Banknote authentication* data set.

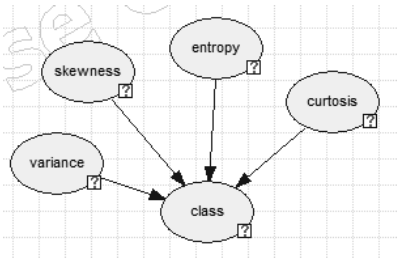


**Fig. 1.** A Bayesian network model of *Banknote authentication* data set.

### 4.2 Hearth disease

The *Hearth disease* [25] is a data set presenting knowledge about the diagnosis of a patient's heart disease. The measurements were carried out in four locations around the world: in Cleveland (United States, Ohio), in Budapest (Hungary), in Zurich (Switzerland) and Long Beach (United States, California). For the purposes of the work, only data collected by the clinic in Cleveland was used, as it was the only one that was processed. The data does not contain real information about the personal data of each of the patients examined. Data for the analysis of knowledge offered by the *Hearth disease* collection has thirteen attributes (including four continuous ones) and fourteenth, which is a decision class. The collection contains only 303 objects. Figure 2 presents the Bayesian network created on the basis of *Hearth disease* data set.

### 4.3 Statlog (image segmentation)

The *Image segmentation* [26] data set presents data extracted from seven pictures presenting brick, sky, vegetation, cement, window, path and grass. These images were adapted to analyze each pixel. All observations included in this set are presented for nine–pixel blocks (3x3). The data was presented using 19 attributes, where 18 of them were continuous attributes, one is a constant attribute and the twentieth attribute was a decision class. The data set contains 2 310 observations. There are no missing values in it, therefore all objects have been included in the research. Figure 3 presents the Bayesian network created on the basis of *Image segmentation* data set.

### 4.4 Abalone

*Abalone* [27] is a data set presenting a few basic physical data of abalone – an edible mollusc of warm seas, with a shallow ear-shaped shell lined with mother–of–pearl

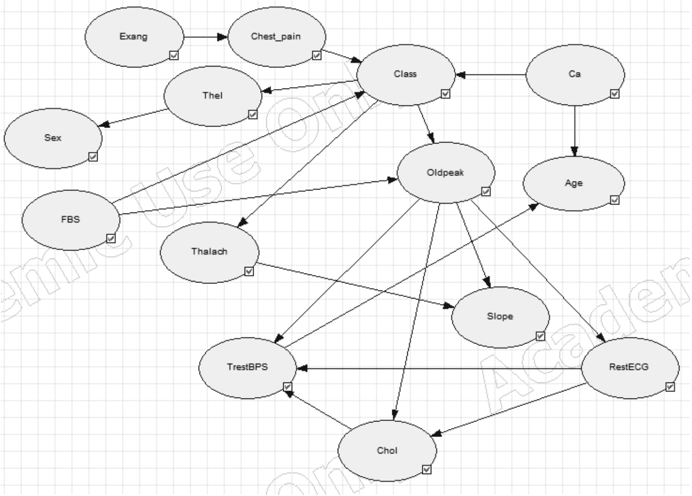


**Fig. 2.** A Bayesian network model of *Hearth disease* data set.

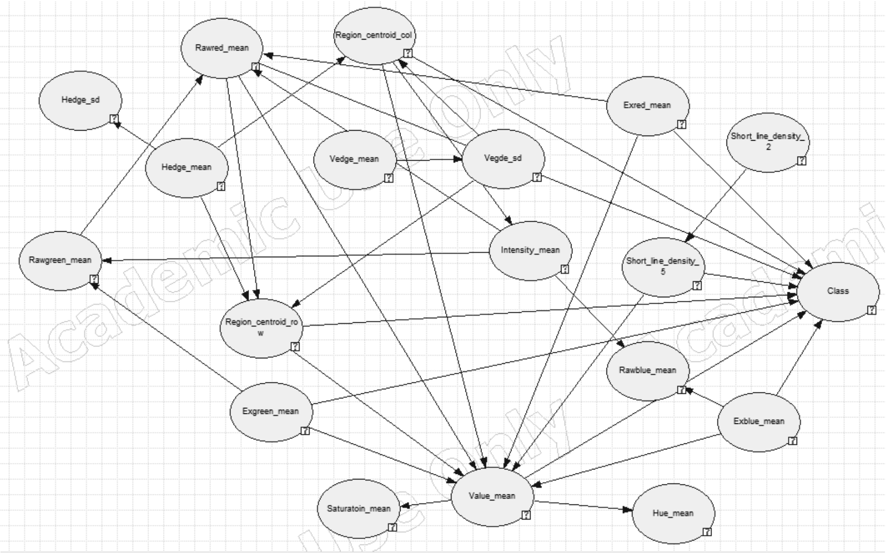


**Fig. 3.** A Bayesian network model of *Statlog (image segmentation)* data set.

and pierced with a line of respiratory holes. Based on these parameters, the age of the abalone is determined. The age of the snail is determined by counting the number of rings on the body using a microscope, but it is a very arduous and time–consuming process. The purpose of the collection is to determine the age of this creature without using a microscope. The set is presented by one nominal attribute, seven continuous attributes and ninth, which is a decision class. The above set contains 4177 observations and contains no missing data. Figure 4 presents the Bayesian network created on the basis of *Abalone* data set.

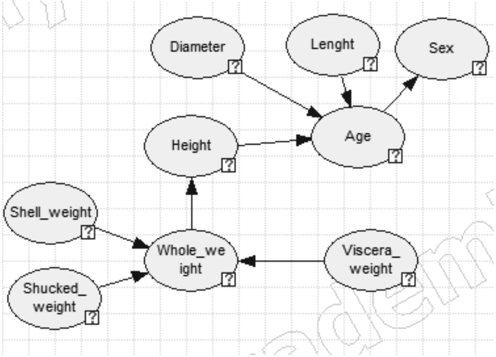


**Fig. 4.** A Bayesian network model of *Abalone* data set.

## 5. Experiments and Results

The main goal of the conducted experiments was to examine how particular methods of data discretization affect the quality of classification of created Bayesian network models, which were learned using discretized data. The quality assessment was determined by means of 10–fold cross–validation. Parameters of discretization methods for research purposes were selected as follows:

- The OneR method as a supervised method does not require specification of the interval length parameter because it is set to the value 6 in advance. However, it has also been decided to test additional values: 7 and 8.
- The Chi Merge method requires the value of parameter $\chi^2$. During the research confidence coefficients of 0.1, 0.2, 0.3 and 0.4 were used. The degree of freedom was determined based on the number of classes in the classification attribute. For the *Banknote authentication* set it was the value of 1, for *Hearth disease* the value of 4, for *Image segmentation* the value of 6 and for the set of *Abalone* – 27.
- In case of EWD and EFD methods, the number of intervals $k$ was set from 2 to 12.

The empirical part of the paper was performed using SMILE, an inference engine, and GeNIe Modeler, a development environment for reasoning in graphical probabilistic models, both developed at BayesFusion LLC, and available at [22].

### 5.1 OneR method

Table 1 shows the results obtained for the OneR method. OneR as a supervised method should not take parameters and its task is to classify based on the default value of the minimum interval length equal to $k = 6$. However, as part of the research, it has been decided to evaluate the quality of the network classification based on the discretized sets of variables not only for the minimum number of elements 6 but also for the minimum number of elements equal to 7 and 8. In the case of the *Hearth disease* and *Image segmentation* data sets, this modification brought a positive result, as the quality assessment increased relative to the result for the default parameter. In the case of the hearth disease data set, an increase of over 12 percentage points was achieved (for $k = 8$) and the best quality result for this set was obtained from among all the methods studied. *Image segmentation* achieved a slight improvement of about 2.8% for intervals with $k = 7$.

**Table 1.** The classification accuracy for the OneR discretization method for different intervals length.

| | k=6 | k=7 | k=8 |
|---|---|---|---|
| Banknote | **91.62%** | 91.18% | 89.80% |
| H. disease | 63.37% | 74.59% | **75.58%** |
| Image seg. | 74.51% | **77.23%** | 76.17% |
| Abalone | **45.70%** | 41.90% | 42.40% |

### 5.2 Chi Merge method

Table 6.4 presents the classification results of the network for sets discretized using the Chi Merge method. The best results were obtained for the smallest value of a confidence test of 0.1. For the *Abalone* and *Image segmentation* data sets the best results for values of 0.2 and 0.4 were obtained respectively. In addition, it can be observed that with the increase of the confidence value, the overall quality of the classification decreased. The exception is the set of *Abalone* for which the percentage of accurately classified objects grew with the increase of this coefficient value reaching 56.33%,

**Table 2.** The classification accuracy for the Chi Merge discretization method for different values of $\chi^2$.

| | $\chi^2 = 0.1$ | $\chi^2 = 0.2$ | $\chi^2 = 0.3$ | $\chi^2 = 0.4$ |
|---|---|---|---|---|
| Banknote | **96.23%** | 96.04% | 94.54% | 92.28% |
| H. disease | **61.22%** | 60.87% | 59.79% | 58.12% |
| Image seg. | 69.43% | **69.53%** | 68.71% | 66.82% |
| Abalone | 53.68% | 52.91% | 55.17% | **65.33%** |

which is the highest quality value measured for this set among all the analyzed algorithms. However, these fluctuations are not big for any of the data sets – the difference between the maximum and minimum closing in around 3-4 percentage points.

## 5.3 EWD method

Table 3 presents the results obtained for the EWD method, taking into account the value of parameter *k* (length of the interval). It can be noticed that the overall accuracy of the classification for the *Abalone* set is very low. The most probable reason is that the decision class attribute contains as many as 28 decision classes that include very diverse number of objects assigned to them. For the *Image segmentation* data set, empty values mean that calculations were impossible due to hardware limitations and the complexity of the Bayesian network. The EWD method worked well in the case of the *Banknote authentication* data set, where the results are very high, mostly exceeding 90%. The biggest differences between the maximum and minimum values occur for the *Image segmentation* data set – the difference between the maximum and minimum values is about 15%. The lowest range occurs for the *Abalone* data set, which is around 3.6%.

**Table 3.** The classification accuracy for the EWD discretization method for different intervals length.

| | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 | k=11 | k=12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Banknote | 85.35% | 94.1% | 95.63% | 97.67% | 97.08% | **98.03%** | 97.45% | 95.77% | 92.71% | 94.9% | 94.75% |
| H. disease | 57.76% | 57.43% | 56.77% | 56.11% | 58.09% | 58.75% | 56.44% | 57.10% | **59.41%** | 55.45% | 57.10% |
| Image seg. | 68.87% | 77.66% | 79.57% | 79.26% | 80.74% | 83.27% | **83.94%** | – | – | – | – |
| Abalone | 22.60% | 22.07% | 24.28% | 24.37% | 25.19% | 23.63% | **26.22%** | 24.66% | 24.83% | 24.92% | 23.94% |

## 5.4 EFD method

Table 4 shows the results obtained for EFD method taking into account the value of parameter *k*, i.e. the length of the interval. The first conclusion is that the obtained results for each data set are weaker than in case of EWD method. In the case of the *Image segmentation* collection, this result is definitely weaker and the difference was around 15 percentage points. In the case of other collections, they are about 1–2 percentage points. Again, a very poor classification result was achieved for the *Abalone* data set – about 23% on an average level. In turn, the best results were obtained by the *Banknote authentication* data set, with the difference that for the higher values of parameter *k*, the classification quality for this set began to decrease. When comparing the maximum quality results of the network classification, it can be observed that for EFD method they are higher for the *Banknote authentication* and *Hearth disease* data sets, and lower for the *Image segmentation* and *Abalone*. Very clear difference in the quality of EWD and EFD methods can be seen in the case of the *Image segmentation* data set.

**Table 4.** The classification accuracy for the EFD discretization method for different intervals length.

| | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 | k=11 | k=12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Banknote | 88.12% | 92.06% | 97.89% | **98.40%** | 98.25% | 97.38% | 94.9% | 93.44% | 92.27% | 88.34% | 86.88% |
| H. disease | 58.42% | 54.79% | 58.42% | **60.07%** | 55.45% | 57.10% | 55.78% | 54.13% | 57.43% | 53.14% | 55.12% |
| Image seg. | 63.85% | **78.87%** | 70.87% | 60.3% | 50.74% | 54.23% | 57.82 | – | – | – | – |
| Abalone | 21.43% | 21.33% | 23.13% | **24.83%** | 24.44% | 24.37% | 23.65% | 21.45% | 21.52% | 22.62% | 22.53% |

## 5.5 Methods comparison

Figure 5 presents the comparison of classification accuracy of different discretization methods for all data sets. For each analyzed data sets the best result for each particular algorithm was chosen and presented in the chart. As can be noticed each of the data sets received the best result for a different method. For the *Hearth disease* data set (75.58%) the OneR method (for the interval length $k = 8$) turned out to be the best. The Chi Merge method achieved the highest classification result (56.33%) for the *Abalone* data set (with the confidence coefficient $\chi^2 = 0.1$). The EWD method with number of intervals $k = 5$ proved to be the best for the *Banknote authentication* data set (98.40%). On the other hand, the EFD method achieved the best result for the *Image segmentation* data set equal to 83.94% for the length of the interval $k = 8$. At this point it should be mentioned that the supervised methods present generally higher quality than the unsupervised ones. However, this trend is not clearly visible in obtained results. In the case of the analyzed data sets, the proportions were divided in half.

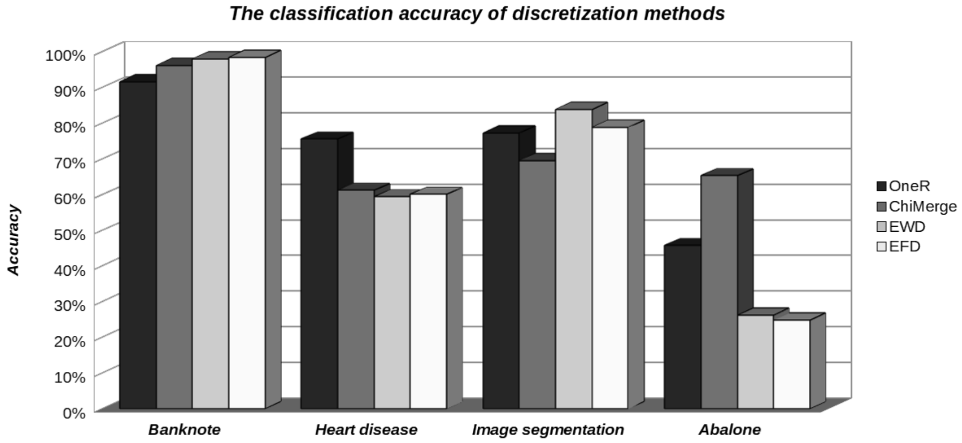


**Fig. 5.** The comparison of classification accuracy of different discretization methods for all data sets.

Regardless to the discretization method, the best results were achieved for *Banknote authentication* data set – each method's classification accuracy was above 90%. The probabilistic model created for this set was the only one in the form of a naïve Bayesian network. Taking into consideration the fact, that this data set contained only about 1 000 objects, such result can confirm the hypothesis stated in [3] that the accuracy of classification can depend on the complexity of created model and almost any discretization method results in significant performance gains for naïve Bayes networks.

The overall accuracy of the classification for the *Abalone* data set was very low – below 50% in most cases. The most probable reason is that the decision class attribute contains as many as 28 decision classes that include very uneven number of objects assigned to them. In such case, the Chi Merge method proved to be the best with the highest result about 65%.

## 6. Conclusion

The conducted research confirmed the belief that there is no universal discretization method, which gives the best result in every data set. Therefore, it is very important to carefully analyze the data on which the tests will be carried out. In order to choose the most effective method, it is worth conducting an experiment using few discretization methods. Basing on such experiment, the appropriate method should be chosen for the given data set.

The *Banknote authentication* data set, regardless of the method used, offers the results of measured quality above 90%. Such result can be the basis to hypothesise that the quality of classification does not only depend on the number and quantity of the observations examined, but also on the designed network model and its complexity. However, further experiments should be carried out using only naíve Bayes network models to check if they produce similar results or not.

Research has also shown that models created for data sets such as *Abalone*, which after the discretization process have many decision classes, achieve very poor classification results regardless of the chosen method. For such type of data sets, the Chi Merge method seemed to be a more universal method that produces good results, regardless of the type or size of data input, relative to other methods of discretization of sets. This does not mean, however, that it always achieved the best results. In some literature [15,20], Chi Merge method is reported to achieve lower classification error than those trained on data pre–processed by the other discretization methods. However, further experiments would be advisable to confirm its effectiveness in the case of data sets with a large number of attributes in the decision class.

At this point, it is also worth adding that in some literature [3] the supervised methods were reported to achieve better results than the unsupervised ones while the contradicting results were obtained by some others [2]. Also the results obtained in this article (as well as in work [21]) do not confirm the superiority of supervised method over unsupervised and vice versa. Therefore, further experimental comparison of of the unsupervised methods versus some of the common supervised methods should be carried out. However, the unsupervised methods will still remain as the only discretization option when we do not have prior known class labels required by the supervised methods.

## References

[1] R. Abraham, J. B. Simha, S. S. Iyengar, A comparative analysis of discretization methods for Medical Datamining with NaĂŻve Bayesian classifier, Information Technology, 2006.

[2] E. Cantú–Paz, Supervised and unsupervised discretization methods for evolutionary algorithms, In proc. Of the genetic and evolutionary computation conference, pp. 213–216, 2001.

[3] J. Dougherty, R. Kohavi, M. Sahami, Supervised and Unsupervised Discretization of Continuous Features, Machine Learning: Proceedings of the Twelfth International Conference, 1995.

[4] A. Ekbal, Improvement of Prediction Accuracy Using Discretization and Voting Classifier, The 18th International Conference on Pattern Recognition, IEEE, 2006.

[5] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine Learning 29 (1997), 131–163.

[6] S. García, J. Luengo, J. A. Sáez, V. López, F. Herrera, Survey of discretization techniques, Taxonomy and empirical analysis in supervised learning, IEEE Transactions on Knowledge and Data Engineering, vol. 25(4), pp. 734–750, 2013.

[7] M. Hacibeyoğlu, M. H. Ibrahim, Comparison of the effect of unsupervised and supervised discretization methods on classification process, International Journal of Intelligent Systems and Applications in Engineering, vol. 4(1), pp. 105–108, 2016.

[8] F. Hussain H. Liu C. L. Tan M.Dash, Discretization: An enabling technique, Data Mining and Knowledge Discovery (2002) 6: 393.

[9] , F. Kaya, Discretizing Continuous Features for Naive Bayes and C4. 5 Classifiers, University of Maryland publications: College Park, MD, USA, 2008.

[10] R. Kerber., Chi Merge: Discretization of numeric attributes, In Proc. Tenth National Conference on Artificial Intelligence, pp. 123–128. MIT Press 1992.

[11] S. Kotsiantis, D. Kanellopoulos, Discretization Techniques: A recent survey, GESTS International Transactions on Computer Science and Engineering, vol.32 (1), 2006.

[12] K. Lavangnananda, S. Chattanachot, Study of discretization methods in classification, 9th International Conference on Knowledge and Smart Technology, pp. 50–55, IEEE, 2017.

[13] P. Lehtinen, M.i Saarel, T. Elomaa, Online Chi Merge Algorithm, Springer, 2012.

[14] D. M. Maslove, T. Podchiyska, H. J. Lowe Discretization of continuous features in clinical datasets J Am Med Inform Assoc., vol. 20(3), pp. 544–553, 2013.

[15] I. Mitov, I. Krassimira, M. Krassimir, V. Velychko, P. Stanchev, K. Vanhoof Comparison of discretization methods for preprocessing data for pyramidal growing network classification method, Information Science & Computing, International Book Series, Number 14, pp. 31–39, 2009.

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

[17] S. Ramírez–Gallego, S. García, H. Mouriño–Talín, D. Martínez–Rego, V. Bolón–Canedo, A. Alonso–Betanzos, J. M. Benítez, F. Herrera Data discretization: taxonomy and big data challenge, WIREs Data Mining Knowledge Discovery, 2015.

[18] A. Rayner, Discretization Numerical Data for Relational Data with One-to-Many Relations

[19] P. Spirtes, C. Glymour, R. Scheines, Causation Prediction and Search, Springer-Verlag, New York, 1993.

[20] C. Zeynel, Y. Figen, Comparison of Chi-square based algorithms for discretization of continuous chicken egg quality traits, Journal of Agricultural Informatics, vol. 8, pp. 13–22, 2017.

[21] C. Zeynel, Y. Figen, Unsupervised Discretization of Continuous Variables in a Chicken Egg Quality Traits Dataset, Turkish Journal of Agriculture – Food Science and Technology, vil. 5, pp. 315–320, 2017.

[22] BayesFusion, LLC, [`https://www.bayesfusion.com/`], Accessed 19-08-2017.

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

[24] Volker Lohweg University of Applied Sciences, Ostwestfalen-Lippe [`https://archive.ics.uci.edu/ml/datasets/Banknote+authentication`], Accessed 03-07-2017.

[25] Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D., University Hospital, Zurich, Switzerland: William Steinbrunn, M.D., University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D., V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D, [`https://archive.ics.uci.edu/ml/datasets/heart+disease`], Accessed 10-07-2017.

[26] Vision Group, University of Massachusetts, [`https://archive.ics.uci.edu/ml/datasets/Statlog+(Image+segmentation)`], Accessed 01-06-2017.

[27] W. J. Nash, T. .L Sellers, S. R. Talbot, Andrew J. Cawthorn, W. B. Ford, The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait, Sea Fisheries Division, Technical Report No. 48 (ISSN 1034-3288), 1994. [`https://archive.ics.uci.edu/ml/datasets/abalone`], Accessed 12-07-2017.

# PORÓWNANIE METOD DYSKRETYZACJI DANYCH W UCZENIU MODELI PROBABILISTYCZNYCH

**Streszczenie** Bardzo często algorytmy uczenia maszynowego są nie są przystosowane do korzystania ze zmiennych ciągłych. Z tego powodu dyskretyzacja danych jest istotną częścią wstępnego przetwarzania. W artykule przedstawiono wyniki prac nad problemem dyskretyzacji danych w uczeniu modeli probabilistycznych. Cztery zestawy danych pobrane z repozytorium uczenia maszynowego UCI zostały wykorzystane do nauczenia parametrów ilościowej części sieci bayesowskich. Występujące w wybranych zbiorach zmienne ciągłe były dyskretyzowane przy użyciu dwóch metod nadzorowanych i dwóch nienadzorowanych. Głównym celem tego artykułu było zbadanie, czy metoda dyskretyzacji danych w danym zbiorze ma wpływ na niezawodność modelu. Dokładność metod była definiowana jako odsetek poprawnie sklasyfikowanych rekordów.

**Słowa kluczowe:** dyskretyzacja, zmienne typu ciągłego, modele probabilistyczne, sieci Bayesa, klasyfikacja

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

# THE LIST OF REVIEWERS (2018)

1. Cezary Bołdak
2. Marta Chodyka
3. Mariusz Frąckiewicz
4. Jakub Gałka
5. Mariusz Gibiec
6. Krzysztof Jurczuk
7. Piotr Kisielewski
8. Tomasz Krzeszowski
9. Andrzej Kużelewski
10. Robert Milewski
11. Dorota Mozyrska
12. Teresa Mroczek
13. Tatiana Odzijewicz
14. Janusz Rafałko
15. Katarzyna Maria Rutczyńska-Wdowiak
16. Mariusz Rybnik
17. Andrzej Sawicki
18. Marcin Skoczylas
19. Adam Słowik
20. Bartłomiej Stasiak
21. Agnieszka Wosiak
22. Adam Zagórecki