JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 25 No. 2 (2017), pp. 29-42

Outlier Detection Using the Multiobjective Genetic Algorithm

Agnieszka Duraj[1], Łukasz Chomątek[1]

[1]*Lodz University of Technology*
Institute of Information Technology
ul. Wólczańska 215, 90-924 Łódź
agnieszka.duraj@p.lodz.pl
lukasz.chomatek@p.lodz.pl

Abstract. *Since almost all datasets may be affected by the presence of anomalies which may skew the interpretation of data, outlier detection has become a crucial element of many datamining applications. Despite the fact that several methods of outlier detection have been proposed in the literature, there is still a need to look for new, more effective ones. This paper presents a new approach to outlier identification based on genetic algorithms. The study evaluates the performance and examines the features of several multiobjective genetic algorithms.*
Keywords: *outliers detection, genetic algorithm.*

1. Introduction

The necessity of dealing with voluminous amounts of data (big data)on a daily basis has led to the development of expert systems and has given rise to the development of intelligent data processing algorithms. Intelligent data analysis facilitates the exploration process at every level. It endeavours to mimic perception by finding and using patterns in the collected data. From this point of view, an essential element to this process is the ability to detect outliers, because every anomaly

may lead to classification or grouping errors and, thus, alter dramatically the analysis result. The users of information systems (decision support systems, expert systems, analytical systems) often deal with the problem of outliers, which may affect data processing results. Although several methods for outlier detection have been proposed in the literature, there is still scope for further development.

In the literature, the concept of an outlier has been defined differently,depending on the application area. For example, a different definition has been proposed for medical image analysis [1], for the detection of computer network congestion or hacking. Other definitions of the outlier have been used for machine diagnostics, for the identification of new molecular structures in pharmaceutical research and for medical diagnostics. In the above cases, the outlier may be treated as the cause of a machine fault, a measurement error, or a certain distinctness, such as unknown features or other attribute values. Among the most commonly used definitions are those proposed by Hawkings [2] Barnet [3] and Aggrawal [4], namely:

- Hawkins [2] "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism".

- Barnett and Lewis [3]:"an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data".

- Aggarwal [4] "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism".

In previous work, the authors of this paper [5, 6] proposed an evolutionary outlier detection method. The novelty of this approach consisted in using a whole set of criteria to decide whether an object was an outlier or not. The present study examines the performance of an outlier detection method using several genetic algorithms. The method proposed is also evaluated in terms of outlier misidentification rates (the number of non-outlier observations being falsely identified as outliers).

The paper is organized as follows. Section 2 gives a brief overview of existing outlier detection methods. The new approach using genetic algorithms is described in Section 3. Section 4 presents the results of the experiments. The paper concludes with the summary and suggestions for further research.

2. Related work

Outlier detection is a rapidly growing branch of data mining. A rich overview of outlier analysis may be found in [2, 4, 7]. Outlier detection using statistical methods is presented in [2, 3]. Researchers often treat linear or logarithmic regression as a preliminary step of data analysis, detecting all outlying points [2, 8, 9, 10]. Koscielniak [11] developed robust regression procedures based on the single and repeated median for calibration models which are non-linear in the parameters. These methods are computed without any explicit optimization function and initially provide no outlier detection procedures.

Undoubtedly, the most popular outlier detection algorithms are: the distance-based-outlier (DB-outlier for short) [12, 13], the Local outlier factor (LOF) [14], Connectivity-based Outlier Factor (COF) [15], and Density Based Spatial Clustering of Applications with Noise (DBSCAN) [16], local outlier [9, 17] See also [18, 19, 20] .

In [21] and [22], outlier detection using the above methods is described. Also worth mentioning are the innovative works by Duraj et al. [22, 23, 24] in which the authors formulate the definition of an outlier in linguistic summaries. The method detects outliers for both numerical and textual data.

Nowak-Brzezińska [25] offers a method for enhancing the performance of knowledge base mining by modification of both the structure of knowledge bases and inference algorithms, thereby improving the efficiency of the inference process. In [26], Nowak-Brzezińska examines the influence of using different similarity measures and clustering methods on the number of outliers detected during the mining process. Innovative solutions were also presented by Emec and Rogowski [27] and Smolinski [28]. In [27], it is shown that observed data can contain values that differ from expected ones and can be interpreted as outliers but in fact are caused by specific physical phenomena.

Outlier detection using genetic algorithms was proposed in [29]. The authors proposed the algorithm, where genes represented the indices of observations that should be treated as outliers. They used three fitness functions. The first one was based on LeastSquare, the second one was based on Cook's Distance and the third one – on the Andrews-Pregibon rate. The method applied a uniform order-based crossover where duplicated entries were removed in the offspring. Tolvi [30] proposed another approach, where individuals were binary vectors. In this encoding, 1 on the i-th place in the chromosome meant that the observation was an outlier. In this method, a linear regression model was proposed to compute the fitness func-

tion. The genetic algorithm was applied to the selection of variables, which was based on the Bayesian Information Criterion [31].The same encoding has been used by many other authors [32, 33]. See also the work by Taloba et al. [34] which proposed Improved Genetic k-Means algorithm to perform clustering with the automatic outlier detection.

3. Problem formulation and the proposed method

Genetic algorithms are a proven and often used iterative heuristic optimization tool. The potential development process may be outlined as follows:

- Each individual consists of a number of genes.
- The values stored in genes are used to compute the value of the fitness function, which is a measure of the individual's effectiveness.
- Each iteration of the genetic algorithm consists of three steps: selection, crossover and mutation.

In practical solutions, there are many methods that determine the a typicality of data, i.e. outliers in a sample. Their effectiveness depends on the type of data set being processed. In this paper, the multi-objective optimization problem (MOP) is described. The present study focuses on finding non-dominated solutions with respect to the domination relation. The domination relation is defined as follows:

The solution $f(x_1) \in \mathbb{R}^n$ dominates $f(x_2) \in \mathbb{R}^n$ iff

$$\forall_{i\in 1,\ldots,n} f_i(x_i) \leq f_i(x_2) \tag{1}$$

and

$$\exists_{i\in 1,\ldots,n} f_i(x_1) < f_i(x_2) \tag{2}$$

where $f_i(x)$ is the i-th value of the fitness function f, and n is the number of objectives.

Konak et al. in [35] provide a comprehensive study of genetic algorithms dedicated for solving MOP. The most popular approaches in this domain are:

- SPEA2 algorithm was proposed by Zitzler et al. [36] ranks the solutions by the number of other individuals they dominate and utilize the concept of archive to protect the best solutions.

- NSGA-II algorithm was proposed by Deb et al. [37], which also favours non-dominated solutions and ensures that individuals are genetically diversified.
- PESA-II algorithm [38] not only uses the archive but also divides the search space into a number of hypercubes. In each region of the search space, an independent optimization process is performed.

The task of properly tuning deterministic or genetic algorithms is very time consuming, since the deterministic and genetic algorithms are parameterized. Finding a comprehensive set of parameters for these algorithms for all problems is laborious and tedious. Several variants of algorithms that identify outlier values can be distinguished. All should be treated equally. The authors propose to choose the optimal Pareto solution related to (dedicated to) the set. This results in a number of solutions, which are then checked for the presence of outliers.

3.1. Encoding and genetic operators

Each chromosome consists of l binary genes, where l is the number of samples in the dataset. The value of each gene can be 1 or 0, depending on whether the corresponding sample is an outlier or not.

To obtain the phenotype, all the samples need to be taken from the dataset, where the corresponding gene is set to 1.

During the experiments, a single point crossover was used with the constant probability p_c. It means that for the chromosomes selected for the reproduction we chose a locus, after that we chose a locus, after that the parts of the chromosomes were exchanged between the parents.

In the mutation, the values of genes in the offspring chromosomes are flipped. Each gene could be changed with the constant probability p_m.

3.2. Proposed fitness function

The performance of the genetic algorithm is determined by the properly composed objective function. In our experiments, three types of measures were used as the components of the fitness function:

- average distance from the nearest neighbours of outlier observations,
- distance from the centroid of observations marked as outliers,

- overall number of outliers in the solution.

Let us denote as ch a chromosome, which represents an individual in the population. For each individual, we can denote as x_{ch} the set of samples identified as outliers and as x'_{ch} a set of samples that are not outliers.

One of the objectives in the fitness function is the average distance of the samples in x_{ch} from the k nearest samples in x'_{ch}. To obtain such a value we need to:

- for each sample $s \in x_{ch}$ calculate the distances from the samples in x'_{ch}.
- sort the samples and take the k-th value.
- add the values obtained in the previous step and divide them between the number of elements in x_{ch}.

This objective is denoted as $d_k(ch)$.

An average distance of the samples in x_{ch} to the centroid of the x'_{ch} dataset is the second type of objective. The centroid is computed as the average position of the vectors in x'_{ch}:

$$c(x'_{ch}) = \frac{\sum_{s' \in x'_{ch}} s'}{|x'_{ch}|} \tag{3}$$

The actual value of this objective is obtained as follows:

$$dc(ch) = \frac{\sum_{s \in x_{ch}} s - c(x'_{ch})}{|x_{ch}|} \tag{4}$$

where $|x_{ch}|$ is the number of elements in the set x_{ch}.

The last type of objective is the number of the identified outliers (denoted as $no(ch)$) as it is assumed that the number of outliers should be much smaller than the number of samples in the whole dataset.

The fitness function applied was composed of five objectives: d_1, d_2, d_3, dc and no.

$$fitness(ch) = [-d_1(ch), -d_2(ch), -d_3(ch), -dc(ch), no(ch)] \tag{5}$$

In eq. (5), first four objectives are taken with "−", because we expect all of the criteria to be minimized (eq. (1), (2)).

Obviously, there can be more than one individual in the Pareto-set found by genetic algorithms. In this case we decided to introduce the accuracy factor for

Table 1: Performance of the algorithms on the limited feature set

Algorithm	Evaluations	CO [%]	COAcc	FO [%]	FOAcc
NSGA-II	2000	83.33	0.733	18.29	0.644
	3000	94.44	0.581	32.92	0.278
	4000	88.89	0.590	20.73	0.228
	5000	77.78	0.523	7.31	0.189
SPEA2	2000	77.78	0.557	9.76	0.426
	3000	44.44	0.662	1.22	0.300
	4000	50.00	0.511	1.22	0.100
	5000	38.88	0.500	0.00	0.000
PESA2	2000	72.22	0.731	13.414	0.333
	3000	61.11	0.654	1.22	0.999
	4000	44.44	0.663	1.22	0.2
	5000	33.33	0.556	1.22	0.667

each sample s (denoted as $acc(s)$). Let X denote the set of individuals that are on the Pareto front. Value of $acc(s)$ is calculated as follows:

$$acc(s) = \frac{\sum_{ch \in X} p(s, ch)}{|X|} \tag{6}$$

where:

$$p(s, ch) = \begin{cases} 1 & \text{for } s \in x_{ch} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

4. Results of the experiments

The experiments were performed on data from Wisconsin Breast Cancer set which is in the UCI Machine Learning [39] repository. In the above set, each of the observations has 10 features, the last of which signifies the type (benign or malignant) of the disease. From this collection of data, the first 100 observations were selected, 82 of which were identified as benign and the rest as malignant cases.

Three different genetic algorithms were used in the calculations:

- NSGA-II
- PESA2
- SPEA-2

The algorithms were run with the following set of parameters:

- evaluations of the fitness function: 2000, 3000, 4000, 5000
- population size: 100
- mutation probability: 2%
- crossover probability: 90%
- number of bisections: 5 (only for PESA algorithm)
- archive size: 10
- tournament selection, 2 competitors

The values for population size and evaluation of the fitness function were chosen arbitrarily. The other parameters were given default values suggested in the literature.

Two series of experiments were conducted. In the first series, the first four attributes for each observation were used, while in the second one all attributes were applied. The type of disease identified in the data set was never taken into account in determining the value of the objective function.

In each run of the two experiments, the following were calculated:

- the percent of correctly identified outliers (denoted CO)
- the average accuracy for correctly found outliers (denoted COAcc)
- the percent of normal observations identified as outliers (denoted FO)
- the average accuracy for incorrectly identified outliers (denoted FOAcc)

The results of the first experiment are presented in Table 1. The NSGA-II algorithm correctly finds most of the outliers.However, in comparison to other algorithms, it is more likely to wrongly identify non-outlier observations as outliers. With more than 3,000 objective function calculations, the values of FOAcc were significantly lower than those of COAcc – even if a correct observation was recognized as an outlier, it had a low value of Acc. The results of the experiment conducted on a full set of features are shown in Table 2. All algorithms behaved like in the case of the limited set of features:

- The NSGA-II algorithm identified the highest number of exceptions, but at the same time often identified correct observations as outliers,
- The other two algorithms were less effective in finding outliers, but they were also less likely to make a mistake about a correct observation.

For all algorithms, the results obtained were worse than for a limited set of features.This is due to a larger search space.

Table 2: Performance of the algorithms on the full feature set

Algorithm	Evaluations	CO [%]	AOAcc	FO [%]	AFOAcc
NSGA-II	2000	83.33	0.578	48.78	0.319
	3000	88.88	0.805	19.51	0.563
	4000	61.11	0.636	18.29	0.295
	5000	38.89	0.714	0.00	0.000
SPEA2	2000	61.11	0.782	6.10	0.460
	3000	44.44	0.612	1.22	0.200
	4000	44.44	0.537	2.44	0.100
	5000	22.22	0.325	0.00	0.000
PESA2	2000	55.55	0.875	3.659	0.667
	3000	27.77	0.700	0.00	0.000
	4000	27.77	0.536	1.22	0.101
	5000	22.22	0.375	0.00	0.000

5. Conclusions

- The genetic algorithms presented here,with the proposed fitness function,have proved suitable for the task of outlier detection.
- It can be seen that for the full set of features the number of "true" outliers is smaller. This is logical since the search space has more dimensions.
- Fewer outliers are identified falsely for the full set of features.
- The NSGA-II algorithm demonstrates superior performance over the other algorithms with regard to outlier detection rate and accuracy values. However, it also produces the highest number of falsely identified outliers.
- The obtained results depend strongly on the number of iterations – the value of FO factor is the lowest when the number of evaluations is the highest.
- The decreased number of correctly identified outliers in the case of an increase in the number of iterations is probably due to the fact that some of the outliers have a greater effect on the value of the objective function. Thus, the remaining observations are eliminated.
- Further work will seek to eliminate the above phenomenon, for example, by changing the fitness function.

Acknowledgment

This work was supported by a grant of the Dean of the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. We would like to thank Piotr Szczepaniak for motivating, very interesting and valuable discussions.

References

[1] Tomczyk, A., *Detection of line segments*, Journal of Applied Computer Science, Vol. 22, No. 2, 2014, pp. 81–90.

[2] Hawkins, D. M., *Identification of outliers*, Vol. 11, Springer, 1980.

[3] Barnett, V. and Lewis, T., *Outliers in statistical data*, Chichester: John Wiley, 1995. 584p, 1964.

[4] Aggarwal, C. C., *Outlier detection in categorical, text and mixed attribute data*, In: Outlier Analysis, Springer, 2013, pp. 199–223.

[5] Chomatek, L. and Duraj, A., *Multiobjective genetic algorithm for outliers detection*, In: INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on, IEEE, 2017, pp. 379–384.

[6] Duraj, A. and Chomatek, L., *Supporting Breast Cancer Diagnosis with Multiobjective Genetic Algorithm for Outlier Detection*, In: International Conference on Diagnostics of Processes and Systems, Springer, 2017, pp. 304–315.

[7] Aggarwal, C. C. and Yu, P. S., *Outlier detection for high dimensional data*, Vol. 30, ACM Sigmod Record.

[8] Goel, A., Xu, H., and Shatz, S. M., *A Multi-State Bayesian Network for Shill Verification in Online Auctions.* In: SEKE, 2010, pp. 279–285.

[9] He, Z., Xu, X., and Deng, S., *Discovering cluster-based local outliers*, Pattern Recognition Letters, Vol. 24, No. 9, 2003, pp. 1641–1650.

[10] Hekimoglu, S., Erenoglu, R. C., and Kalina, J., *Outlier detection by means of robust regression estimators for use in engineering science*, Journal of Zhejiang University Science A, Vol. 10, No. 6, 2009, pp. 909–921.

[11] Kościelniak, P., *Non-linear robust regression procedure for calibration in flame atomic absorption spectrometry*, Analytica chimica acta, Vol. 278, No. 1, 1993, pp. 177–187.

[12] Knorr, E. M., Ng, R. T., and Tucakov, V., *Distance-based outliers: algorithms and applications*, The VLDB Journal—The International Journal on Very Large Data Bases, Vol. 8, No. 3-4, 2000, pp. 237–253.

[13] Knox, E. M. and Ng, R. T., *Algorithms for mining distancebased outliers in large datasets*, In: Proceedings of the International Conference on Very Large Data Bases, Citeseer, 1998, pp. 392–403.

[14] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J., *LOF: identifying density-based local outliers*, In: ACM sigmod record, Vol. 29, ACM, 2000, pp. 93–104.

[15] Jin, W., Tung, A. K., and Han, J., *Mining top-n local outliers in large databases*, In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 293–298.

[16] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al., *A density-based algorithm for discovering clusters in large spatial databases with noise.* In: Kdd, Vol. 96, 1996, pp. 226–231.

[17] Kriegel, H.-P., Kröger, P., Schubert, E., and Zimek, A., *LoOP: local outlier probabilities*, In: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, 2009, pp. 1649–1652.

[18] Orair, G. H., Teixeira, C. H., Meira Jr, W., Wang, Y., and Parthasarathy, S., *Distance-based outlier detection: consolidation and renewed bearing*, Proceedings of the VLDB Endowment, Vol. 3, No. 1-2, 2010, pp. 1469–1480.

[19] Kreinovich, V., Longpré, L., Patangay, P., Ferson, S., and Ginzburg, L., *Outlier detection under interval uncertainty: algorithmic solvability and computational complexity*, Reliable Computing, Vol. 11, No. 1, 2005, pp. 59–76.

[20] Schubert, E., Zimek, A., and Kriegel, H.-P., *Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection*, Data Mining and Knowledge Discovery, Vol. 28, No. 1, 2014, pp. 190–237.

[21] Duraj, A. and Krawczyk, A., *Finding outliers for large medical datasets*, Przeglad Elektrotechniczny, Vol. 86, 2010, pp. 188–191.

[22] Duraj, A. and Szczepaniak, P. S., *Information Outliers and Their Detection*, In: Information Studies and the Quest for Transdisciplinarity, World Scientific Publishing Company, 2017, pp. 413–437.

[23] Duraj, A., Szczepaniak, P. S., and Ochelska-Mierzejewska, J., *Detection of Outlier Information Using Linguistic Summarization*, 2016, pp. 101–113.

[24] Duraj, A., *Outlier detection in medical data using linguistic summaries*, In: INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on, IEEE, 2017, pp. 385–390.

[25] Nowak-Brzezińska, A., *Mining rule-based knowledge bases inspired by rough set theory*, Fundamenta Informaticae, Vol. 148, No. 1-2, 2016, pp. 35–50.

[26] Nowak-Brzezińska, A., *Outlier mining in rule-based knowledge bases*, In: INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on, IEEE, 2017, pp. 391–396.

[27] Emets, V. and Rogowski, J., *Scattering of acoustical waves by a hard strip and outlier phenomenon*, In: INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on, IEEE, 2017, pp. 376–378.

[28] Smolinski, M., *Resolving classical concurrency problems using adaptive conflictless scheduling*, In: INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on, IEEE, 2017, pp. 397–402.

[29] Crawford, K. D. and Wainwright, R. L., *Applying Genetic Algorithms to Outlier Detection.* In: ICGA, 1995, pp. 546–550.

[30] Tolvi, J., *Genetic algorithms for outlier detection and variable selection in linear regression models*, Soft Computing, Vol. 8, No. 8, 2004, pp. 527–533.

[31] Schwarz, G. et al., *Estimating the dimension of a model*, The annals of statistics, Vol. 6, No. 2, 1978, pp. 461–464.

[32] Alma, Ö. G., Serdar, K., and Aybars, U., *Genetic algorithm based outlier detection using Bayesian information criterion in multiple regression models having multicollinearity problems*, Gazi University Journal of Science, Vol. 22, No. 3, 2009, pp. 141–148.

[33] Cucina, D., di Salvatore, A., and Protopapas, M. K., *Outliers detection in multivariate time series using genetic algorithms*, Chemometrics and Intelligent Laboratory Systems, Vol. 132, 2014, pp. 103–110.

[34] Taloba, A. I., Marghny, M., and El-Aziz, R. M. A., *Outlier Detection using Improved Genetic K-means*, International Journal of Computer Applications, 2014.

[35] Konak, A., Coit, D. W., and Smith, A. E., *Multi-objective optimization using genetic algorithms: A tutorial*, Reliability Engineering & System Safety, Vol. 91, No. 9, 2006, pp. 992–1007.

[36] Zitzler, E., Laumanns, M., Thiele, L., et al., *SPEA2: Improving the strength Pareto evolutionary algorithm*, 2001.

[37] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE transactions on evolutionary computation, Vol. 6, No. 2, 2002, pp. 182–197.

[38] Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J., *PESA-II: Region-based selection in evolutionary multiobjective optimization*, In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., 2001, pp. 283–290.

[39] Lichman, M., *UCI Machine Learning Repository*, 2013.