Clustering-based rule generation methods for fuzzy classifier using Autonomous Data Partitioning algorithm

M O Svetlakov\textsuperscript{1} and I A Hodashinsky\textsuperscript{2}
\textsuperscript{1} Junior researcher, Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia
\textsuperscript{2} Professor of Computer Science, Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia
E-mail: svetlakov.m4@gmail.com

Abstract. In this paper, clustering-based rule generation methods for fuzzy classifier using non-parametric Autonomous Data Partitioning algorithm have been proposed. ADP-algorithm is used to determine the number of clusters for use in various k-means-like clustering algorithms. Proposed method contributes to solving the problem of determining optimal number of clusters/rules. The efficiency of fuzzy classifiers with rules constructed by the specified algorithms has been tested on data sets from the KEEL repository. Experimental results show that proposed method outperforms baseline algorithm (the extremums rulebase generation algorithm) both in terms of classification accuracy and geometric mean metrics.

1. Introduction
Fuzzy systems rival in accuracy other machine learning methods and found applications in various fields, i.e. control systems for technological processes [1]. Fuzzy classifiers stand out among other machine learning classifying methods because "IF-THEN" type fuzzy rules can be interpreted by human experts and can also be created using human experts knowledge. When creating a fuzzy system, the structure of the rule base is generated and the parameters are identified. Parameter are then usually “fine”-tuned, which is generally performed through derivatives, swarm intelligence algorithms or evolutionary computations, so it is important to have good initial estimation of rulebase parameters and structure in order to speed-up learning convergence process. One of the main ways to build an initial base of rules for a fuzzy classifier is to build a set of rules using the extremums algorithm. The main drawback is the number of rules created is always equal to the number of classes in the dataset from which the rulebase is created.

It is possible to use a clustering method and use clusters to form a set of fuzzy rules, but usually clustering methods require pre-specifying the number of clusters or spend time trying out different values of hyperparameters. Proposed method contributes to solving the problem of determining optimal number of clusters/rules. Autonomous Data Partitioning clustering algorithm is the non-parametric clustering algorithm that can help to extract close to optimal number of rules, low enough to keep fuzzy classifier compact. The extracted number of rules is then used to set the number of clusters for various clustering algorithms.
2. Fuzzy classifier

As discussed in [2], a fuzzy model can be constructed either on the basis of expert knowledge or using dataset of table-like values \( D = \{(x^p; y_p), p = 1, ..., z\} \), where \( x^p = (x^p_1, ..., x^p_n) \) is the vector of input features values for the \( p \)-th classification object, \( y_p \) is the value of the output variable (object class). So, input data \( D \) look the following way:

\[
D = \left[ \begin{array}{cccccc}
  x^1_1 & x^1_2 & \ldots & x^1_j & \ldots & x^1_n & y_1 \\
  x^2_1 & x^2_2 & \ldots & x^2_j & \ldots & x^2_n & y_2 \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
  x^m_1 & x^m_2 & \ldots & x^m_j & \ldots & x^m_n & y_m \\
\end{array} \right] \tag{1}
\]

A fuzzy system is formed from a set of ”IF-THEN” type fuzzy. Each rule has two parts; IF-part (rule antecedents) have a statement about input features values, THEN-part (rule consequent) have a statement about output class value. In a fuzzy classifier, the \( i \)-th rule look the following way:

\[
R_{ij} : IF \; x_1 = A_{1i} \; AND \; x_2 = A_{2i} \; AND \ldots \; AND \; x_n = A_{ni} \; THEN \; class = c_j, \tag{2}
\]

where \( A_{ki} \) is a fuzzy term for the \( k \)-th feature in the \( i \)-th rule \((i \in [1,R])\), \( R \) is the number of rules; \( c_j \) - identifier of the \( j \)-th class, \( j \in [1,m] \); \( x_i \in A \).

A fuzzy classifier can be represented by a function:

\[
c = f(x^p, \Theta, r), \tag{3}
\]

where \( \Theta \) is a vector describing the antecedent part of the fuzzy classifier rule base \((\Theta = \theta^k, k = 1, \ldots, s)\), \( r = (r_1, \ldots, r_s) \) is a vector of consequent values. Using training data \( D = \{(x^p; y_p), p = 1, ..., z\} \), let’s define the classification quality function that is essentially widely used classification accuracy \( E_{acc}(\Theta) \):

\[
delta(x, y, r, \Theta) = \begin{cases} 
1, & \text{if } y_p = f(x^p, \Theta), \; p = 1, 2, \ldots, z, \\
0, & \text{otherwise}
\end{cases} \tag{4}
\]

\[
E_{acc}(x, y, r, \Theta) = \frac{1}{z} \sum_{p=1}^{z} \delta(x, y, r, \Theta). \tag{5}
\]

Alternatively, we can use geometric mean as metric:

\[
E_{geom}(x, y, r, \Theta) = \left( \prod_{i=1}^{m} E(x_{class_i}, y_{class_i}, r, \Theta) \right)^\frac{1}{m}. \tag{6}
\]

3. Generating rules

3.1. Baseline method

The algorithm for generating a rule base using extreme values of input features [3] is used as baseline algorithm. This method can result in super compact rulebases and have good enough initial accuracy (the generated rulebase is usually further optimized using various metaheuristics). The disadvantage of this algorithm is that the number of rules created is always equal to the number of classes in the dataset from which the rulebase is created. The extremum algorithm iterates over all classes data, finding minimum and maximum value for each feature for each class. From minimum and maximum values tuples fuzzy terms \( A_{kj} \) are formed.
3.2. Proposed method
This paper uses a hybrid algorithm for building the rule base. The first phase uses a non-parametric clustering algorithm Autonomous Data Partitioning (ADP) [4]. The algorithm is similar to the algorithm for constructing rules for the Angelov-Yager system [5], uses similar concepts - local density, global density, clouds-clusters. In terms of computational costs, this algorithm mostly needs only distance matrix for all data points. Using distance matrix global density for each data point is calculated, data points are sorted with regards to the global mode, local modes are detected. Data clouds (clusters) are formed around those local modes.

Autonomous Data Partitioning algorithm does not require hyperparameter tuning and automatically produces a certain number of clusters. The resulting number of rules is used as a parameter $k$ (the number of generated cluster centroids) for the k-means clustering algorithm and few others. Based on the obtained $k$ clusters, the rules of the fuzzy system are formed, the resulting rule base is estimated using the criterion $E_{acc}(x,y,r,\Theta)$.

4. Experiment
Experiment datasets were taken from the KEEL repository [6]. The cross-validation scheme is used (5-fold, 80:20 ratio training/test data splits), the membership function type used for fuzzy terms was Gaussian-like function. The datasets are described in table [1].

| Dataset     | Classes | Features | Samples | $ADP_k$ | $OPTICS_k$ | $BIRCH_k$ |
|-------------|---------|----------|---------|---------|------------|-----------|
| appendicitis| 7       | 7        | 106     | 13.4    | 2.4        | 50.0      |
| balance     | 3       | 4        | 645     | 57.0    | 1.0        | 500.0     |
| banana      | 2       | 2        | 5300    | 12.6    | 262.4      | 13.2      |
| bupa        | 2       | 6        | 345     | 21.8    | 2.4        | 118.4     |
| glass       | 7       | 9        | 214     | 12.8    | 7.0        | 67.6      |
| haberman    | 2       | 3        | 306     | 19.8    | 15.0       | 47.0      |
| hayes-roth  | 3       | 4        | 160     | 19.8    | 3.2        | 73.2      |
| heart       | 2       | 13       | 270     | 30.6    | 12.8       | 209.0     |
| ionosphere  | 2       | 33       | 351     | 25.2    | 6.2        | 250.8     |
| iris        | 3       | 4        | 150     | 12.6    | 6.4        | 30.8      |
| mammographic| 2       | 5        | 830     | 28.6    | 45.0       | 110.4     |
| monk-2      | 2       | 6        | 432     | 48.8    | 4.0        | 345.6     |
| newthyroid  | 3       | 5        | 215     | 8.8     | 3.6        | 30.6      |
| phoneme     | 2       | 5        | 5404    | 31.4    | 203.6      | 221.6     |
| somar       | 2       | 60       | 208     | 35.6    | 5.4        | 166.4     |
| tae         | 3       | 5        | 151     | 13.8    | 10.4       | 46.6      |
| titanic     | 2       | 3        | 2201    | 3.8     | 12.4       | 13.8      |
| wdbc        | 2       | 30       | 569     | 25.6    | 2.0        | 444.4     |

For this work, Python package scikit-learn [7] implementation clustering algorithms implementations were used. Autonomous Data Partitioning’s original author implementation was taken from supplementary source of [4]. From all scikit-learn clustering methods only the following were selected:

- Minibatch K-means (Minibatch) with default parameters;
- K-means (Kmeans) with default parameters;
Table 2. $1 - E_{acc}(x, y, r, \Theta)$, Test data.

| Data          | Extrem. | ADP   | Minibatch | Kmeans | Aggl. | Spectral |
|---------------|---------|-------|-----------|--------|-------|----------|
| appendicitis  | 0.27403 | 0.12208 | 0.16926   | 0.13247 | 0.15065 | 0.15065 |
| balance       | 0.53920 | 0.47040 | 0.19520   | 0.23360 | 0.21120 | 0.23680 |
| banana        | 0.56321 | 0.44962 | 0.19075   | 0.16943 | 0.19642 | 0.21566 |
| bupa          | 0.51594 | 0.51594 | 0.41739   | 0.41449 | 0.45797 | 0.40000 |
| glass         | 0.50487 | 0.68217 | 0.50000   | 0.49502 | 0.51838 | 0.51860 |
| haberman      | 0.44098 | 0.31036 | 0.28408   | 0.27784 | 0.27795 | 0.26822 |
| hayes-roth    | 0.59375 | 0.61875 | 0.58750   | 0.59375 | 0.50625 | 0.58125 |
| heart         | 0.33704 | 0.40741 | 0.28519   | 0.27037 | 0.27407 | 0.26667 |
| ionsphere     | 0.11095 | 0.34161 | 0.24773   | 0.23074 | 0.15082 | 0.37895 |
| iris          | 0.05333 | 0.30667 | 0.06000   | 0.04667 | 0.08667 | 0.08667 |
| mammographic  | 0.32972 | 0.54842 | 0.20179   | 0.22220 | 0.21062 | 0.18414 |
| monk-2        | 0.44448 | 0.37273 | 0.35646   | 0.38412 | 0.37049 | 0.35643 |
| newthyroid    | 0.04186 | 0.24651 | 0.07907   | 0.08837 | 0.08372 | 0.14884 |
| phoneme       | 0.25703 | 0.29349 | 0.23094   | 0.22372 | 0.22169 | 0.22465 |
| sonar         | 0.44228 | 0.43229 | 0.43206   | 0.42811 | 0.38420 | 0.49930 |
| tae           | 0.64860 | 0.64817 | 0.45742   | 0.49720 | 0.46387 | 0.44387 |
| titanic       | 0.32303 | 0.32531 | 0.22534   | 0.22625 | 0.22807 | 0.22625 |
| wdbc          | 0.08260 | 0.55186 | 0.07387   | 0.05974 | 0.07555 | 0.13881 |

Mean rank 5.55556 6.38889 3.22222 2.94444 3.30556 3.63889

% gain baseline -6% 8% 8% 9% 7%

- Agglomerative Clustering (Aggl.) with default parameters;
- Spectral Clustering (Spectral) with default parameters;
- BIRCH with default parameters;
- OPTICS with default parameters;

The number of clusters was extracted not only using the ADP algorithm, but also using the BIRCH and OPTICS algorithms. The number of generated clusters using different methods is shown in table 1. The other two algorithms (BIRCH and OPTICS) generate too many or too few rules and therefore are not suitable for determining the number of clusters. All selected (except BIRCH and OPTICS) clustering algorithms need pre-specifying number of clusters (in this work this parameter is set to number of clusters extracted by ADP algorithm, $ADP_k$ in table 1).

The ADP algorithm has also been tested to generate an initial rule base without use of the k-means algorithm. Table 2 shows the accuracy metric (5) results for evaluating the initial basic rules built by different algorithms on the test data samples, and table 3 shows geometric mean (6) metric results for the test data samples. In order to determine better algorithm it’s better to use accuracy metric because glass dataset have severe class imbalance, giving inadequate geometric mean metric results. The best results on the training and test sets were shown by the algorithms Minibatch K-means, K-means, Agglomerative Clustering, Spectral Clustering (an average increase in accuracy of 13% on training and 8% on the test for accuracy metric, 15% gain on training and 14% gain on the test for geometric mean metric). The results of these 4 algorithms do not differ (the Friedman test showed $p-value = 0.88233 > 0.05$).
Table 3. $1 - E_{geom}(x, y, r, \Theta)$, Test data.

| Data          | Extrem. | ADP   | Minibatch | Kmeans | Aggl.  | Spectral |
|---------------|---------|-------|-----------|--------|--------|----------|
| appendicitis  | 0.38199 | 0.26045 | 0.36167 | 0.36167 | 0.39784 | 0.36167  |
| balance       | 1.00000 | 0.93692 | 0.83655 | 0.83655 | 0.83655 | 0.75290  |
| banana        | 0.56413 | 0.88624 | 0.20182 | 0.18347 | 0.16512 | 0.18347  |
| bupa          | 0.60357 | 0.63987 | 0.55477 | 0.50434 | 0.50434 | 0.55477  |
| glass         | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.00000  |
| haberman      | 0.59798 | 0.80372 | 0.59562 | 0.59562 | 0.65518 | 0.65518  |
| hayes-roth    | 1.00000 | 0.84961 | 0.67284 | 0.67284 | 0.67284 | 0.67284  |
| heart         | 0.39860 | 0.48292 | 0.24531 | 0.24531 | 0.24531 | 0.24531  |
| ionosphere    | 0.12235 | 0.54034 | 0.23928 | 0.23928 | 0.26321 | 0.26321  |
| iris          | 0.05697 | 0.31513 | 0.07633 | 0.06939 | 0.07633 | 0.06245  |
| mammographic  | 0.33631 | 0.57229 | 0.19381 | 0.19381 | 0.19381 | 0.17443  |
| monk-2        | 0.49143 | 0.40426 | 0.35900 | 0.35900 | 0.35900 | 0.35900  |
| newthyroid    | 0.08878 | 1.00000 | 0.14242 | 0.14242 | 0.14242 | 0.14242  |
| phoneme       | 0.30335 | 1.00000 | 0.25890 | 0.25890 | 0.28479 | 0.25890  |
| sonar         | 0.54371 | 0.49941 | 0.42465 | 0.42465 | 0.46712 | 0.42465  |
| tae           | 0.82628 | 0.67425 | 0.49167 | 0.49167 | 0.49167 | 0.44250  |
| tianic        | 1.00000 | 0.85188 | 0.32292 | 0.32292 | 0.32292 | 0.29063  |
| wdbc          | 0.11746 | 0.89778 | 0.05960 | 0.05960 | 0.05960 | 0.06556  |

% gain baseline -18% 13% 14% 14% 14%

5. Conclusions
Using a combination of the ADP algorithm and any of the 4 clustering algorithms supporting specifying the number of clusters (Minibatch K-means, K-means, Agglomerative Clustering, Spectral Clustering), it is possible to generate an initial rulebase for fuzzy classifier with a higher accuracy and number of rules than in case of using extremum generation algorithm.

Acknowledgments
This research was funded by the Ministry of Education and Science of the Russian Federation within the framework of scientific projects carried out by teams of research laboratories of educational institutions of higher education subordinate to the Ministry of Science and Higher Education of the Russian Federation, project number FEWM-2020-0042.

References
[1] Gadaras I and Mikhailov L 2009 Artif. Intell. Med. 47 25–41 ISSN 0933-3657 URL [https://doi.org/10.1016/j.artmed.2009.05.003](https://doi.org/10.1016/j.artmed.2009.05.003)
[2] Nauck D and Kruse R 1999 Artificial Intelligence in Medicine 16 149–169 ISSN 0933-3657 URL [https://www.sciencedirect.com/science/article/pii/S0933365798000700](https://www.sciencedirect.com/science/article/pii/S0933365798000700)
[3] Hodashinsky I and Samsonov S 2017 Business Informatics 61–67 URL [https://bijournal.hse.ru/data/2017/05/22/1172269370/7.pdf](https://bijournal.hse.ru/data/2017/05/22/1172269370/7.pdf)
[4] Gu X, Angelov P and Principe J 2018 Information Sciences 460
[5] Angelov P and Gu X 2017 2017 Evolving and Adaptive Intelligent Systems (EAIS) 1–7
[6] Alcala-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sanchez L and Herrera F 2010 Journal of Multiple-Valued Logic and Soft Computing 17 255–287
[7] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R,
Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E 2011 J. Mach. Learn. Res. 12 2825–2830 ISSN 1532-4435