STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXIX**, Number 1, 2024
DOI: 10.24193/subbi.2024.1.03

A COMPREHENSIVE EVALUATION OF ROUGH SETS CLUSTERING IN UNCERTAINTY DRIVEN CONTEXTS

ARNOLD SZEDERJESI-DRAGOMIR

Abstract. This paper presents a comprehensive evaluation of the Agent BAsed Rough sets Clustering (ABARC) algorithm, an approach using rough sets theory for clustering in environments characterized by uncertainty. Several experiments utilizing standard datasets are performed in order to compare ABARC against a range of supervised and unsupervised learning algorithms. This comparison considers various internal and external performance measures to evaluate the quality of clustering. The results highlight the ABARC algorithm's capability to effectively manage vague data and outliers, showcasing its advantage in handling uncertainty in data. Furthermore, they also emphasize the importance of choosing appropriate performance metrics, especially when evaluating clustering algorithms in scenarios with unclear or inconsistent data.

1. Introduction

Clustering algorithms play an important role in uncovering patterns and structures from unlabelled data across several scientific and engineering domains [5, 12, 16, 14, 7]. The added value of these algorithms lies in their ability to group data points based on underlying similarities, thereby facilitating a deeper understanding of dataset characteristics without prior knowledge of the group identities. In real-world contexts where uncertainty and ambiguity often pervade, the ability to discern coherent groups within a dataset becomes indispensable. However, traditional clustering techniques are often inadequate in environments driven by uncertainty, including the presence of hybrid data (vague data or outliers). This limitation shows the necessity for innovative approaches that can robustly handle the complexities induced by the uncertainty and ambiguity of such landscapes.

Received by the editors: 5 March 2024.

2010 *Mathematics Subject Classification.* 68T37, 68T99.

1998 *CR Categories and Descriptors.* I.5.3 [**Pattern Recognition**]: Clustering – *Algorithms*; I.5.3 [**Pattern Recognition**]: Clustering – *Similarity measures* .

Key words and phrases. rough sets, clustering, metrics.

The introduction of rough sets theory by Pawlak [28] has facilitated the development of clustering algorithms capable of handling uncertainty more effectively. Rough sets have mostly been applied to feature extraction [2, 17, 22, 27, 35, 19, 37, 33, 4], their use in direct cluster modeling being significantly less common. The approaches from [21, 25, 24, 20] investigate rough sets clustering, but they are all partitioning methods. ABARC [11], on the other hand, is a hierarchical clustering algorithm that distinguishes itself by its adeptness at detecting hybrid data by using rough sets, as well as isolating outliers, thereby promising enhanced clustering performance in scenarios driven by uncertainty in data.

The evaluation of clustering algorithms in the context of uncertainty-driven environments needs to take into account the particularities detected in data. Internal and external performance metrics serve as critical tools in this process, providing insights into an algorithm's ability to generate cohesive and well-separated clusters while aligning with external validity measures when ground truth is available.

This paper aims to perform a comprehensive comparison of the ABARC algorithm against several supervised and unsupervised learning algorithms, employing a suite of performance metrics to assess each algorithm's efficacy across standard datasets. Through this comparative analysis, our aim is to show the strengths and limitations of the ABARC algorithm and its counterparts, thereby contributing to the ongoing research on optimal clustering approaches in the context of data uncertainty.

The paper is structured as follows: Section 2 presents an overview of the clustering algorithm based on rough sets, Section 3 illustrates the comprehensive experiments we made, including evaluation based on external, internal and rough metrics and Section 4 draws the conclusions of this paper and presents potential future work.

2. Rough sets clustering

Rough sets [28] represent an effective methodology for addressing data uncertainty and vagueness, without the need of membership functions (which could be hard to build) like in fuzzy set theory. Employing an equivalence relation R within a dataset U, rough set theory proposes a mechanism to approximate uncertain subsets $X \subseteq U$ via two distinct and precise sets: the lower and upper approximations. The lower approximation is comprised of elements that are surely in X and it is defined as $R^{\downarrow}(X) = \{x \in U : [x]_R \subseteq X\}$, where $[x]_R$ represents the equivalence class of x under R. Conversely, the upper approximation includes elements that possibly belong to X, defined as $R^{\uparrow}(X) = \{x \in U : [x]_R \cap X \neq \emptyset\}$. The boundary region, delineated as

$Bnd_R(X) = R^{\uparrow}(X) - R^{\downarrow}(X)$, consists of objects that cannot be definitively classified as belonging or not belonging to the subset X. Accordingly, the rough set of X relative to R is denoted as $RS(X) = \{R^{\downarrow}(X), R^{\uparrow}(X)\}$.

Rough sets clustering [11] uses rough sets theory to effectively group a dataset into clusters while acknowledging the existing uncertainties and ambiguities in data.

Definition 1. *Given a dataset U (universe of discourse) and an equivalence relation R on U, the goal of* ***rough sets clustering*** *is to partition U into a set of clusters $\{C_1, C_2, \ldots, C_k\}$ such that:*

- $U = \bigcup_{i=1}^{k} C_i$ *and* $C_i \cap C_j = \emptyset$ *for* $i \neq j$.
- *Each cluster C_i is represented by its lower and upper approximations ($R^{\downarrow}(C_i)$, $R^{\uparrow}(C_i)$) with respect to R.*
- *The boundary region for each cluster C_i is given by $Bnd_R(C_i) = R^{\uparrow}(C_i) - R^{\downarrow}(C_i)$.*

In the context of clustering: (1) objects in the lower approximation of a cluster *definitively* belong to that cluster; (2) objects in the upper approximation *might* belong to the cluster (3) objects in the boundary region of a cluster may belong to the boundary regions of other clusters.

In Algorithm 1 we show an overview of the ABARC rough sets clustering algorithm from [11].

Algorithm 1 Rough Sets Clustering

```
Require: $X$ (dataset), $imax$ (number of trials), $\lambda$ (similarity limit)
Initialize $AG$ (set of agents) with one agent for each instance in $X$.
For each agent in $AG$, assign it to a unique cluster.
for $i = 1$ to $imax$ do
    for each $agent_k$ in $AG$ do
        Find a similar agent ($sa_k$) using a similarity threshold $\lambda$.
        if $sa_k$ is found then
            Move $agent_k$ to the cluster of $sa_k$.
        end if
    end for
end for
for each cluster representative $R_k$ do
    Find similar clusters based on a rough similarity limit.
    Update and unify clusters based on similarity.
end for
Handle outliers by assigning them to the closest cluster.
```

The algorithm receives a dataset X, along with a maximum trial limit $imax$ and a similarity threshold λ, as inputs. In this appproach, each element of X is represented by an agent and the set of all agents is denoted with AG. Initially, every agent is allocated to a distinct cluster, leading to a total of n clusters corresponding to n agents.

Iteratively, up to $imax$ rounds, the algorithm refines the clustering structure by allowing each agent $agent_k$ to seek peers within the similarity boundary set by λ. Upon identifying a similar agent sa_k, $agent_k$ relocates to sa_k's cluster, thus similar agents are grouped together.

Based on the representative of each cluster, the algorithm checks the similarity among clusters through these representatives. If any clusters are similar, they are merged into a unified cluster. This step ensures that clusters that are close to each other or have significant overlap can be combined to form a more cohesive and meaningful cluster. It is possible for a representative to be similar to more than one other cluster representative in which case the corresponding data (rough instances) will be treated as it would belong to several clusters.

An optional phase addresses outliers - agents that do not seem to fit into any other cluster - by assigning them to the nearest cluster, thus ensuring that all data points are included in a cluster.

3. Experimental evaluation

In this section we calculate various external (Section 3.1), internal (Section 3.2), as well as rough (Section 3.3) metrics as we compare the ABARC approach with several other algorithms in the literature. The ABARC algorithm is compared with other approaches in three scenarios: including all instances, eliminating only outliers and eliminating both outliers and rough instances (i.e. eliminating all hybrid data).

The experiments are performed on the following datasets: Iris [8], Seeds [18], and Wine [9]. These datasets have been chosen primarily for benchmarking purposes and, secondly, because they present challenges such as the presence of outliers and instances that are not linearly separable, which makes them suitable for applying the ABARC algorithm.

3.1. External evaluation metrics.

3.1.1. *Metrics.*

- Accuracy - in a clustering context, it represents the percentage of instances that were correctly predicted out of all instances

- Precision - focuses on how many predicted instances were classified correctly for a given class:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- Recall - focuses on how many actual instances were predicted correctly for a given class:

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

- F1-Score - incorporates both precision and recall using harmonic mean (thus punishing extreme values) with even weights, this metric is also for a given class:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- Macro F1-Score - combines the F1 Score from each class using arithmetic mean (this can be applied to precision and recall metrics too):

$$Macro\ F_1 = \frac{1}{|C|} * \sum_{c \in C} F_{1_c}$$

where C is the set of classes, and c is a class

- Weighted Average F1-Score - same as Macro F1-Score but with class sizes used as weights (again this can be used for precision and recall too):

$$Weighted\ Average\ F_1 = \frac{1}{|C|} * \sum_{c \in C} |c| * F_{1_c}$$

- Micro F1-Score (Accuracy) - to calculate this we take all the samples together and compute precision and recall, and then the F1 Score. In cases where the number of predicted instances is equal to the actual instances, we will have the following equation hold:

$$Micro\ Precision = Micro\ Recall = Micro\ F_1 = Accuracy$$

- Kappa Score - when talking about Kappa Score we need to introduce new terms: *Agree* which is the proportion of correctly predicted instance over all instances (similar to Accuracy) and *Chance Agree* which is computed from the probabilities of predicting a class or being in a class, formally:

$$Agree = \frac{1}{N} * \sum_{c \in C} |predicted_c \cap actual_c|$$

Table 1. Overall supervised metrics for the Iris dataset.

Case Study	Class	Instances	Prec	Recall	F1	Kappa
Clusters with hybrids	Macro	163	91.19	91.19	91.15	-
	Weighted	163	90.8	90.88	90.8	-
	Micro	163	90.798	90.798	90.798	-
	Kappa	163	-	-	-	86.2
Clusters without outliers	Macro	152	90.64	90.74	90.64	-
	Weighted	152	90.13	90.21	90.12	-
	Micro	152	90.132	90.132	90.132	-
	Kappa	152	-	-	-	85.1
Clusters without hybrids	Macro	126	98.35	98.35	98.35	-
	Weighted	126	98.41	98.41	98.41	-
	Micro	126	98.413	98.413	98.413	-
	Kappa	126	-	-	-	97.6

where N is the total number of instances, $predicted_c$ is the set of all instances predicted in class c, and $actual_c$ is the set of all instances actually being in class c.

$$Chance\ Agree = \sum_{c \in C} \frac{|predicted_c|}{N} * \frac{|actual_c|}{N}$$

$$Kappa\ Score = \frac{Agree - Chance\ Agree}{1 - Chance\ Agree}$$

3.1.2. *Results and discussion.*

Iris dataset. Analyzing the results from Table 1 we can see that outliers make no real difference, but when we eliminate rough instances we get much better results on all metrics.

We have also compared our metrics to some related work. We have used as a comparison the following results from [36]: KMEA, WKME, EWKM, ESSC, AFKM, SC, SSC-MP, ERKM; and from [29]: Bayes Network Classifier, J48, Random Forest, OneR. In Table 2 we can see that the F1-Score for the two ABARC cases is better than all of the others, but the Kappa Score is better only after removing hybrids.

We have also compared to algorithms from Scikit learn [38] like Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbours (KNN), Decision Tree (DT), Gaussian Naive Bayes (GNB), Support Vector Machines (SVM). From Table 3 we can again conclude that removing hybrids is essential to have the best values for all the metrics.

Table 2. Comparison with related work on Iris dataset.

Algorithm	Precision	Recall	F1-Score	Kappa Score
KMEA [23]			81.2	
WKME [13]			79.8	
EWKM [15]			82.6	
ESSC [6]			84.8	
AFKM [1]			81.6	
SC [31]			47.2	
SSC-MP [32]			76.7	
ERKM [36]			90.2	
Bayes Network Classifier				89
J48				94
Random Forest				93
OneR				91
ABARC /w hybrids	91.2	91.2	91.2	86.2
ABARC /wo hybrids	**98.4**	**98.4**	**98.4**	**97.6**

Table 3. Comparison with Scikit learn on Iris dataset.

Algorithm	Precision	Recall	F1-Score	Kappa Score
LR	95.4	95.2	95.2	92.9
LDA	98.2	97.9	97.9	96.9
KNN	97.8	97.8	97.7	97.0
DT	95.3	95.1	95.1	92.9
GNB	95.7	95.5	95.5	92.8
SVM	97.8	97.7	97.7	96.9
ABARC /w hybrids	91.2	91.2	91.2	86.2
ABARC /wo hybrids	**98.4**	**98.4**	**98.4**	**97.6**

Seeds dataset. When taking a look in Table 4, the overall metrics show small difference when outliers are removed, and eliminating rough instances does not seem to make any difference. A potential reason why hybrid data might have such a small impact is the reduced number of outliers and rough instances.

We have used the same algorithm from Scikit learn. When taking a look on Table 5 we can observe that our approach is nowhere near being the best. This can once more happen because the algorithm's performance is not affected by rough instances and outliers.

Wine dataset. Wine is one of the datasets where hybrids make difference. The idea can be observed when we take a look at the overall metrics in Table 6: outliers improve metrics, but rough instances are the ones that make the real difference bumping all metrics to above 99%.

Table 4. Overall supervised metrics for the Seeds dataset.

Case Study	Class	Instances	Prec	Recall	F1	Kappa
Clusters with hybrids	Macro	213	91.52	91.61	91.5	-
	Weighted	213	91.55	91.84	91.63	-
	Micro	213	91.549	91.549	91.549	-
	Kappa	213	-	-	-	87.3
Clusters without outliers	Macro	192	91.13	91.59	91.28	-
	Weighted	192	91.15	91.42	91.21	-
	Micro	192	91.146	91.146	91.146	-
	Kappa	192	-	-	-	86.7
Clusters without hybrids	Macro	178	91.46	92.23	91.77	-
	Weighted	178	91.57	91.79	91.62	-
	Micro	178	91.573	91.573	91.573	-
	Kappa	178	-	-	-	87.2

Table 5. Comparison with Scikit learn on Seeds dataset.

Algorithm	Precision	Recall	F1-Score	Kappa Score
LR	90.0	90.2	89.5	84.9
LDA	**95.9**	**96.2**	**96.0**	**94.2**
KNN	91.7	92.0	91.4	87.7
DT	88.3	88.5	88.1	82.7
GNB	89.7	90.1	89.5	84.9
SVM	92.6	92.7	92.5	89.2
ABARC /w hybrids	91.5	91.6	91.5	87.3
ABARC /wo hybrids	91.5	92.2	91.8	87.2

Table 6. Overall supervised metrics for the Wine dataset.

Case Study	Class	Instances	Prec	Recall	F1	Kappa
Clusters with hybrids	Macro	186	93.2	94.79	93.59	-
	Weighted	186	93.55	94.3	93.51	-
	Micro	186	93.548	93.548	93.548	-
	Kappa	186	-	-	-	90.2
Clusters without outliers	Macro	165	94.23	95.31	94.32	-
	Weighted	165	94.55	95.31	94.49	-
	Micro	165	94.545	94.545	94.545	-
	Kappa	165	-	-	-	91.8
Clusters without hybrids	Macro	148	99.24	99.29	99.26	-
	Weighted	148	99.32	99.34	99.32	-
	Micro	148	99.324	99.324	99.324	-
	Kappa	148	-	-	-	99.0

We have done the one of the comparisons from Iris on the Wine dataset too, illustrated in Table 7. As there is only one metric interpreting the results

Table 7. Comparison with related work on Wine dataset.

Algorithm	F1-Score
KMEA [23]	94.6
WKME [13]	94.8
EWKM [15]	90.4
ESSC [6]	95.0
AFKM [1]	94.3
SC [31]	86.9
SSC-MP [32]	58.4
ERKM [36]	89.9
ABARC /w hybrids	93.6
ABARC /wo hybrids	**99.3**

Table 8. Comparison with Scikit learn on Wine dataset.

Algorithm	Precision	Recall	F1-Score	Kappa Score
LR	95.0	96.1	95.3	93.1
LDA	97.6	97.8	97.6	96.5
KNN	66.4	68.2	65.9	51.5
DT	89.6	89.2	89.2	84.3
GNB	97.5	97.8	97.6	96.5
SVM	60.7	39.5	30.4	9.5
ABARC /w hybrids	93.2	94.8	93.6	90.2
ABARC /wo hybrids	**99.2**	**99.3**	**99.3**	**99.0**

is trivial and we can see the same tendency: ABARC with hybrids has good performance but not good enough to be better than all of the other related work, but when we remove hybrids the value becomes almost perfect, thus being the best of all.

We have the comparison with Scikit learn algorithms on the Wine dataset too. Table 8 shows that the performance of ABARC is comparable to that of the related work, albeit slightly lower. Nevertheless, the advantage of our approach is that it can also detect and isolate hybrid data.

3.2. **Internal evaluation metrics.**

3.2.1. *Metrics.*

- Purity - a measure of the extent to which clusters contain a single class:

$$Purity = \frac{1}{N} \sum_{k \in K} max_{c \in C} \; a_{ck}$$

where N is the number of instances, K is the set of clusters, and C is the set of classes, and $a_{ck} = |c \cap k|$

- Entropy - a measure of uncertainty

$$Entropy = \sum_{k\in K} \frac{|k|}{N} * (-\sum_{c\in C} P(a_{ck}) * \log_2 P(a_{ck}))$$

  where

$$P(a_{ck}) = \frac{a_{ck}}{|c|} = \frac{|c \cap k|}{|c|}$$

- V-Measure - is again based on entropy but considers homogeneity and completeness with different importance. Homogeneity means that a clustering must assign only those datapoints that are members of a single class to a single cluster, completeness is symmetrical to homogeneity: a clustering must assign all of those datapoints that are members of a single class to a single cluster. Formally we calculate homogeneity, completeness and V-Measure as:

$$H(C|K) = -\sum_{k\in K}\sum_{c\in C} \frac{a_{ck}}{N} * \log \frac{a_{ck}}{\sum_{c\in C} a_{ck}}$$

$$H(C,K) = -\sum_{c\in C} \frac{\sum_{k\in K} a_{ck}}{|C|} \log \frac{\sum_{k\in K} a_{ck}}{|C|}$$

$$h = \begin{cases} 1 & H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C,K)} & \text{otherwise} \end{cases}$$

$$H(K|C) = -\sum_{c\in C}\sum_{k\in K} \frac{a_{ck}}{N} * \log \frac{a_{ck}}{\sum_{k\in K} a_{ck}}$$

$$H(K,C) = -\sum_{k\in K} \frac{\sum_{c\in C} a_{ck}}{|C|} \log \frac{\sum_{c\in C} a_{ck}}{|C|}$$

$$c = \begin{cases} 1 & H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K,C)} & \text{otherwise} \end{cases}$$

$$V_\beta = \frac{(1+\beta) * h * c}{\beta * h + c}$$

3.2.2. *Results and discussion.* Considering the results from Table 9 we can observe that on Iris the accuracy and purity drops a bit as we eliminate outliers and rough instances, but the entropy and the V-Measure, after dropping both of them, are significantly better, which makes us assume that outliers and rough instances do not really affect homogeneity but they affect completeness. On the Seeds dataset we cannot see any real difference when eliminating them, thus they do not affect our performance. Finally, on Wine we can see all metrics improve, entropy and V-Measure improve significantly, so on this

TABLE 9. Unsupervised performance measurements for the Iris, Seeds and Wine datasets.

	Case Study	Inst	Acc	Entropy	Purity	V
Iris	Clusters with hybrids	150	98.66%	0.0803	0.987	0.733
	Clusters without outliers	139	98.56%	0.0847	0.986	0.717
	Clusters without outliers and rough	126	98.41%	0.0204	0.984	0.932
Seeds	Clusters with hybrids	210	92.857%	0.0839	0.929	0.721
	Clusters without outliers	190	92.105%	0.0863	0.921	0.711
	Clusters without outliers and rough	178	91.573%	0.0829	0.916	0.719
Wine	Clusters with hybrids	178	97.753%	0.0569	0.978	0.8
	Clusters without outliers	157	99.363%	0.0418	0.994	0.854
	Clusters without outliers and rough	148	99.324%	0.0088	0.993	0.97

dataset eliminating them makes our results almost perfect regardless of the metric used.

To compare with some related work we used the following results:

- KMEA, WKME, EWKM, ESSC, AFKM, SC, SSC-MP, ERKM [36]
- Bayes Network Classifier, J48, Random Forest, OneR [29]
- KM, EWKM, AFKM, FCM, SCAD, Entropy-based Variable Feature Weighted Fuzzy k-Means (EVFWFKM) [30]
- UFT-k-means, k-prototypes, Improved k-prototypes, KL-FCM-GM [34]

These are used in Table 10, and there can be multiple entries for a single algorithm (ex. EWKM) as they are taken from different results probably run using different configuration. The first comparison from Table 10 again suggests that ABARC has the performance, this time on Seeds too. Although both the compared metrics are the best in our approach, on the dataset Iris and Seeds the accuracy is actually better with hybrids than without them (this can happen when hybrid instances are accidentally put in the cluster specified by the official documentation), but the entropy values are always better without hybrids. Other algorithms does not seem to be even close to the values reported by ABARC in any of the cases.

When we compare to the Scikit learn algorithms we have the same results as for the supervised metrics. They are much better on Seeds dataset, but our approach especially without hybrids has much better performance on Iris and Wine dataset from both metrics' point of view.

3.3. Rough evaluation metrics.

3.3.1. *Metrics.* We have also evaluated the ABARC algorithm against the following rough indices from [26]:

Table 10. Unsupervised comparison with related work for the Iris, Seeds and Wine datasets.

Dataset	Algorithm	Accuracy	Entropy
Iris	KMEA [23]	80.54%	
	WKME [13]	78.47%	
	EWKM [15]	82.09%	
	ESSC [6]	84.66%	
	AFKM [1]	81.27%	
	SC [31]	80.66%	
	SSC-MP [32]	71.20%	
	ERKM [36]	90.36%	
	Bayes Network Classifier	92.66%	
	J48	96%	
	Random Forest	95.33%	
	OneR	94%	
	KM [23]	88.67%	
	EWKM [15]	89.78%	
	AFKM [1]	90.67%	0.299
	FCM [3]	82.67%	0.395
	SCAD [10]	88.67%	0.395
	EVFWFKM [30]	92.67%	0.294
	ABARC /w hybrids	**98.66%**	0.08
	ABARC /wo hybrids	98.41%	**0.02**
Seeds	UFT-k-means	89.05%	
	k-prototypes	86.67%	
	Improved k-prototypes	84.76%	
	KL-FCM-GM	57.62%	
	ABARC /w hybrids	**92.857%**	0.084
	ABARC /wo hybrids	91.573%	**0.083**
Wine	KMEA [23]	94.43%	
	WKME [13]	94.71%	
	EWKM [15]	90.24%	
	ESSC [6]	95.06%	
	AFKM [1]	93.99%	
	SC [31]	87.07%	
	SSC-MP [32]	58.65%	
	ERKM [36]	90.16%	
	ABARC /w hybrids	97.753%	0.057
	ABARC /wo hybrids	**99.324%**	**0.009**

(1) Average Accuracy, α index - it is the average of the ratio of the number of objects in lower approximation to that in upper approximation of each cluster, it captures the average degree of completeness of knowledge about all clusters: $\alpha = \frac{1}{|K|} \sum_{k \in K} \frac{\omega * A_k}{\omega * A_k + (1-\omega) * B_k}$ where A_k is the size of the lower approximation of cluster k, B_k is the size

Table 11. Unsupervised comparison with Scikit learn algorithms for the Iris, Seeds and Wine datasets.

Dataset	Algorithm	Accuracy	V-Measure
Iris	LR	94.67	86.5
	LDA	97.33	91.7
	KNN	96.67	89.9
	DT	95.33	86.7
	GNB	95.33	87.2
	SVM	96.00	88.6
	ABARC /w hybrids	**98.66**	73.3
	ABARC /wo hybrids	98.41	**93.2**
Seeds	LR	90.00	73.6
	LDA	**96.19**	**88.1**
	KNN	91.90	77.3
	DT	88.10	67.7
	GNB	90.00	73.4
	SVM	92.86	78.5
	ABARC /w hybrids	92.857	72.1
	ABARC /wo hybrids	91.573	71.9
Wine	LR	95.51	85.1
	LDA	98.32	94.2
	KNN	69.70	39.8
	DT	90.46	75.9
	GNB	96.59	90.4
	SVM	38.19	9.7
	ABARC /w hybrids	97.753	80.0
	ABARC /wo hybrids	**99.324**	**97.0**

of the boundary region of cluster k and ω is the weight of lower approximation (we used 0.6)

(2) Average Roughness, ρ index - represents the average degree of incompleteness of knowledge about all clusters: $\rho = 1 - \alpha$

(3) Accuracy of Approximation, α^* index - it captures the exactness of approximate clustering: $\alpha^* = \frac{\sum_{k \in K} \omega * A_k}{\sum_{k \in K} \omega * A_k + (1-\omega) * B_k}$

(4) Quality of Approximation, γ index - it is the ratio of the total number of objects in lower approximations of all clusters to the cardinality of the universe of discourse: $\gamma = \frac{1}{N} \sum_{k \in K} A_k$

3.3.2. *Results and discussion.* We have compared our rough indices results with the ones reported in the book in Table 12 (only on Iris and Wine datasets). From the comparison we can say that our approach matches the algorithms discussed in the related work on the Iris dataset. The first three indices are slightly lower but the last one is significantly better, probably meaning that

Table 12. Rough indices for the Iris, Seeds and Wine dataset.

Dataset	Iris				Wine			
Algorithm	α Index	ρ Index	α^* Index	γ Index	α Index	ρ Index	α^* Index	γ Index
$RFCM^{MBP}$	0.999971	0.000029	0.999963	0.625000	0.8387	0.1613	0.9251	0.5000
RFCM	0.999986	0.000014	0.999988	0.800000	0.8918	0.1082	0.9259	0.8275
RPCM	0.999983	0.000017	0.999985	0.553333	0.8433	0.1567	0.9306	0.6255
RFPCM	**0.999987**	**0.000013**	**0.999989**	0.766667	0.9012	0.0988	0.9258	0.7234
ABARC	0.999980	0.000020	0.999981	**0.913333**	**0.9989**	**0.0011**	**0.9989**	**0.9438**

overall our rough score is better but it is slightly worse on one of the clusters. On the Wine dataset all indices are the best in the ABARC algorithm's case, meaning that we have better rough clustering from all points of view.

4. Conclusions and future work

In this paper, we have carried out a comprehensive evaluation of the Agent BAsed Rough sets Clustering (ABARC) algorithm, which is a new approach for clustering in uncertain environments. Experiments were done using standard datasets and against multiple supervised and unsupervised methods too. Besides this evaluation, we also analyze the impact of several internal and external metrics, especially in the context of unpredictability.

The results suggest that removing hybrids increases the performance of the ABARC algorithm. Compared to other approaches on Iris and Wine datasets the algorithm outperforms all the related approaches with respect to almost any of the considered metrics. This outcome emphasises the importance of hybrid data detection and hence the need of applying algorithms that are tailored to uncertainty driven environments.

As a future work we plan to analyze rough instances and outliers even more in order to potentially gain extra relevant information about the given datasets as well as work on applying ABARC in other domains, like software engineering, biology, chemistry or even medicine.

References

[1] Bachem, O., Lucic, M., Hassani, H., and Krause, A. Fast and provably good seedings for k-means. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 55–63.

[2] Bera, S., Giri, P. K., Jana, D. K., Basu, K., and Maiti, M. Multi-item 4d-tps under budget constraint using rough interval. *Applied Soft Computing 71* (2018), 364 – 385.

[3] Bezdek, J. C., Ehrlich, R., and Full, W. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences 10*, 2-3 (Jan. 1984), 191–203.

[4] Bharadwaj, A., and Ramanna, S. Categorizing relational facts from the web with fuzzy rough sets. *Knowledge and Information Systems 61*, 3 (Dec 2019), 1695–1713.

[5] Coy, S., Czumaj, A., and Mishra, G. On parallel k-center clustering. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures* (New York, NY, USA, 2023), SPAA '23, Association for Computing Machinery, p. 65–75.

[6] Deng, Z., Choi, K.-S., Chung, F.-L., and Wang, S. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. *Pattern Recognition 43*, 3 (Mar. 2010), 767–781.

[7] Fielding-Singh, P., and Fan, J. X. Dietary patterns among us children: A cluster analysis. *Journal of the Academy of Nutrition and Dietetics* (2023).

[8] Fisher, R. A. *UCI Machine Learning Repository: Iris Data Set.* http://archive.ics.uci.edu/ml/datasets/Iris, 1936.

[9] Forina, M. *UCI Machine Learning Repository: Wine Data Set.* https://archive.ics.uci.edu/ml/datasets/wine, 1991.

[10] Frigui, H., and Nasraoui, O. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition 37*, 3 (Mar. 2004), 567–581.

[11] Găceanu, R. D., Szederjesi-Dragomir, A., Pop, H. F., and Sârbu, C. Abarc: An agent-based rough sets clustering algorithm. *Intelligent Systems with Applications 16* (2022), 200117.

[12] Hong, J., and Kim, S.-W. C-affinity: A novel similarity measure for effective data clustering. In *Companion Proceedings of the ACM Web Conference 2023* (New York, NY, USA, 2023), WWW '23 Companion, Association for Computing Machinery, p. 41–44.

[13] Huang, J., Ng, M., Rong, H., and Li, Z. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 5 (May 2005), 657–668.

[14] Janowski, A. M., Ravellette, K. S., Insel, M., Garcia, J. G., Rischard, F. P., and Vanderpool, R. R. Advanced hemodynamic and cluster analysis for identifying novel rv function subphenotypes in patients with pulmonary hypertension. *The Journal of Heart and Lung Transplantation* (2023).

[15] Jing, L., Ng, M. K., and Huang, J. Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on Knowledge and Data Engineering 19*, 8 (Aug. 2007), 1026–1041.

[16] Karim, S. M., Habbal, A., Hamouda, H., and Alaidaros, H. A secure multifactor-based clustering scheme for internet of vehicles. *Journal of King Saud University - Computer and Information Sciences 35*, 10 (2023), 101867.

[17] Kato, Y., Saeki, T., and Mizuno, S. Considerations on the principle of rule induction by strim and its relationship to the conventional rough sets methods. *Applied Soft Computing 73* (2018), 933 – 942.

[18] Kulczycki, P. *UCI Machine Learning Repository: Seeds Data Set.* https://archive.ics.uci.edu/ml/datasets/seeds, 2012.

[19] Lei, L. Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. *Applied Soft Computing 62* (2018), 923 – 932.

[20] Li, Y., Fan, J.-c., Pan, J.-S., Mao, G.-h., and Wu, G.-k. A novel rough fuzzy clustering algorithm with a new similarity measurement. *Journal of Internet Technology 20*, 4 (2019), 1145–1156.

[21] Lingras, P., and West, C. Interval set clustering of web users with rough k-means. *J. Intell. Inf. Syst. 23*, 1 (2004), 5–16.

[22] Liu, Y., Qin, K., and Martinez, L. Improving decision making approaches based on fuzzy soft sets and rough soft sets. *Applied Soft Computing 65* (2018), 320 – 332.

[23] MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* (Berkeley, Calif., 1967), University of California Press, pp. 281–297.

[24] Maji, P., and Pal, S. *Rough-fuzzy pattern recognition: applications in bioinformatics and medical imaging*, vol. 3. John Wiley & Sons, 2012.

[25] Maji, P., and Pal, S. K. Rough set based generalized fuzzy -means algorithm and quantitative indices. *Trans. Sys. Man Cyber. Part B 37*, 6 (2007), 1529–1540.

[26] MAJI, P., and PAL, S. K. *ROUGH-FUZZY PATTERN RECOGNITION*. Wiley, 2012.

[27] Pamucar, D., Stevic, Z., and Zavadskas, E. K. Integration of interval rough ahp and interval rough mabac methods for evaluating university web pages. *Applied Soft Computing 67* (2018), 141 – 163.

[28] Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[29] Rajhasthan, Sharma, K., and College, S. Classification of iris dataset using wekas, Dec 2019.

[30] Singh, V., and Verma, N. K. An entropy-based variable feature weighted fuzzy k-means algorithm for high dimensional data. *arXiv preprint arXiv:1912.11209* (2019).

[31] Tarn, C., Zhang, Y., and Feng, Y. Sampling clustering. *CoRR abs/1806.08245* (2018).

[32] Tschannen, M., and Bolcskei, H. Noisy subspace clustering via matching pursuits. *IEEE Transactions on Information Theory 64*, 6 (June 2018), 4081–4104.

[33] Wang, P.-C., Su, C.-T., Chen, K.-H., and Chen, N.-H. The application of rough set and mahalanobis distance to enhance the quality of osa diagnosis. *Expert Systems with Applications 38*, 6 (2011), 7828 – 7836.

[34] Wei, M., Chow, T. W., and Chan, R. H. Clustering heterogeneous data with k-means by mutual information-based unsupervised feature transformation. *entropy 17*, 3 (2015), 1535–1548.

[35] Xie, X., Qin, X., Yu, C., and Xu, X. Test-cost-sensitive rough set based approach for minimum weight vertex cover problem. *Applied Soft Computing 64* (2018), 423 – 435.

[36] Xiong, L., Wang, C., Huang, X., and Zeng, H. An entropy regularization k-means algorithm with a new measure of between-cluster distance in subspace clustering. *Entropy 21*, 7 (July 2019), 683.

[37] Yang, H.-H., and Wu, C.-L. Rough sets to help medical diagnosis - evidence from a taiwan's clinic. *Expert Systems with Applications 36*, 5 (2009), 9293 – 9298.

[38] https://scikit-learn.org.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Computer Science Department, Cluj-Napoca, România

Email address: `arnold.szederjesi@ubbcluj.ro`