Title: Enhancing the Efficiency of a Decision Support System through the Clustering of Complex Rule-Based Knowledge Bases and Modification of the Inference Algorithm

Author: Agnieszka Nowak-Brzezińska

Citation style: Nowak-Brzezińska Agnieszka. (2018). Enhancing the Efficiency of a Decision Support System through the Clustering of Complex Rule-Based Knowledge Bases and Modification of the Inference Algorithm. "Complexity" (Vol. 13 (2018), art. ID 2065491), doi 10.1155/2018/2065491

Hindawi
Complexity
Volume 2018, Article ID 2065491, 14 pages
https://doi.org/10.1155/2018/2065491

Research Article

Enhancing the Efficiency of a Decision Support System through the Clustering of Complex Rule-Based Knowledge Bases and Modification of the Inference Algorithm

Agnieszka Nowak-Brzezińska

Institute of Computer Science, Faculty of Computer Science and Material Science, Silesian University, ul.Będzińska 39, 41-200 Sosnowiec, Poland

Correspondence should be addressed to Agnieszka Nowak-Brzezińska; agnieszka.nowak@us.edu.pl

Received 19 April 2018; Revised 9 September 2018; Accepted 30 September 2018; Published 6 December 2018

Guest Editor: Ireneusz Czarnowski

Decision support systems founded on rule-based knowledge representation should be equipped with rule management mechanisms. Effective exploration of new knowledge in every domain of human life requires new algorithms of knowledge organization and a thorough search of the created data structures. In this work, the author introduces an optimization of both the knowledge base structure and the inference algorithm. Hence, a new, hierarchically organized knowledge base structure is proposed as it draws on the cluster analysis method and a new forward-chaining inference algorithm which searches only the so-called representatives of rule clusters. Making use of the similarity approach, the algorithm tries to discover new facts (new knowledge) from rules and facts already known. The author defines and analyses four various representative generation methods for rule clusters. Experimental results contain the analysis of the impact of the proposed methods on the efficiency of a decision support system with such knowledge representation. In order to do this, four representative generation methods and various types of clustering parameters (similarity measure, clustering methods, etc.) were examined. As can be seen, the proposed modification of both the structure of knowledge base and the inference algorithm has yielded satisfactory results.

1. Introduction

Big Data is no longer just about processing a huge number of bytes, but doing things with data that you could not do previously. It is not just tabular data you can easily stick into a spreadsheet or a database [1]. Where computer scientists were once limited to mere gigabytes or terabytes of information, they are now studying petabytes and even exabytes of information. At the same time, the tools to sift all that data are getting better as computer scientists refine and improve the algorithms they use to extract meaning from the deluge of data [2]. There is no doubt that big data are now rapidly expanding in all science and engineering domains. While the potential of these massive data is undoubtedly significant, fully making sense of them requires new ways of thinking and novel learning techniques to address the various challenges. Most traditional machine learning techniques are not inherently efficient or scalable enough to handle the data with the characteristics of large volume, different types, high speed, uncertainty and incompleteness, and low value density. In response, machine learning needs to reinvent itself for big data processing [3]. Current hot topics in the quest to improve effectiveness of the machine learning techniques include search for compact knowledge representation methods and better tools for knowledge discovery and integration.

The main subject of the author's scientific work lies at the boundary of artificial intelligence, methods of representation and exploration of domain knowledge, statistical methods of data analysis, and machine learning methods. Recent work focuses on managing complex knowledge bases with rule representation and the development of new inference algorithms in such data sets.

In order to extract useful domain knowledge from the studied area, a lot of data should be collected beforehand. Much also depends on how the rules are induced. For example, effective rule induction algorithms can generate a compressed set of several dozen or several hundred rules for a data set consisting of several thousand objects. That is why when talking about domain knowledge bases, files with several thousand rules are often considered to be too large [4]. The author's experience of working on such amount of data is presented in [5]. In this research, the author has focused on discovering the optimal methods for big data storage, managing management, and exploration. In order to do this, the preliminary experiments, using medium-sized knowledge bases with various types and sizes of data, were carried out. The goal is to specify the most important parameters that facilitate a quick and effective discovery of new knowledge in knowledge bases.

In inference processes based on the rule-based knowledge bases, we explore new domain knowledge by activating the rules (components of a rule-based system with form: IF premises THEN conclusion) with true premises—the ones which may have been covered by the facts given a priori. The process of activating a given rule results in dealing with its conclusion as a new fact. The more rules and initial facts in a given knowledge base, the more rules that can be activated. Of course, the recent solutions in the area of decision support systems require that they additionally perform the task in the shortest time and with the least human involvement. Let us take an example of the medical system, in which we aim to make a decision as fast as possible, based on the knowledge (facts) about a particular patient. The system searches a knowledge base with rules in order to find all the rules relevant to the given set of facts. In case of a big data set, with many rules, such a process can be too time-consuming. The classic approach is then inefficient, as it has to search every rule in a given knowledge base, which in case of big dataset takes too much time. Thus, new solutions need to be discovered and developed. Such solutions should result in the effectiveness not worse than it is in the case of the classic approach, doing it as quickly and as efficiently as possible. It requires a deep analysis of the knowledge stored in the knowledge bases and exploration of the information about a given domain, for example, in the form of so-called meta-knowledge (knowledge about knowledge). In the literature, there is a lot of research devoted to the subject of meta-knowledge and meta-rules [6–8].

It is widely known that the best way to learn a new field is to use generalization skills. Generalization is the process of discovering general features, important features, and the features common for a given class of objects. Following this path, the generalization of the information saved in the rules allows us to gain knowledge about those rules. By attributing similar rules to one group and through the generalization of such groups, we obtain knowledge about many rules without having to review each rule separately.

The notion proposed in this paper is built around the idea of the similarity analysis between the rules and then their subsequent clustering. Among numerous clustering algorithms, the agglomerative hierarchical clustering (AHC) algorithm was chosen (the author previously analysed many other algorithms as well [9, 10]). Its most important feature (and advantage) is the fact that it clusters (agglomerates) the most similar rules and forms a group from them. Regarding the rules in the knowledge base, we must take into account that from a certain moment of clustering, the rules cease to be similar in any respect and there is no reason to cluster them any longer. Thus, the classic clustering AHC algorithm requires a modification. Furthermore, to effectively (efficiently and quickly) find the right group of rules to activate, it is necessary to describe them optimally. The author has recently devoted much attention to the proposal and analysis of methods for representing groups of rules, using the generalization approach [11]. This paper is aimed at verifying the effectiveness of inference, i.e., the ability to activate rules by reviewing only a selected part of the entire knowledge base, most relevant to the given facts. An inference process can be considered successfully finished where only a small part of the entire knowledge base is searched and we are able to successfully find and activate a given rule (or rules).

It turns out that some clustering parameters have a significant impact on the structure of groups of rules (a tendency to create small or large clusters, to identify atypical rules and separate them from groups). Moreover, certain methods of representation of rule clusters (representative generation methods) are characterized by a tendency to create overly general representatives (or sometimes empty) or overly detailed representatives that have ceased to reflect the content of the whole group. Having knowledge about which clustering parameters and which representative generation methods ensure the best efficiency, we will be able to strive to achieve optimal results.

The structure of the paper is as follows. Section 2 introduces the rule-based knowledge bases and inference processes in decision support systems. Managing of rules in knowledge bases is the main subject of Section 3. The proposed approach with a description of the clustering algorithm and inference algorithm for a hierarchical structure of a knowledge base with rule clusters is presented in Section 4. The results of experiments with their interpretation are included in Section 5. The summary is presented in Section 6.

2. Knowledge-Based Systems

The knowledge-based system (KBS) is a system that uses artificial intelligence to solve problems. It focuses on using knowledge-based techniques to support human decision making, learning, and action. Such systems are capable of cooperating with human users and are fit for purpose. We may even say that they are better than humans are, as they are enriched with the virtues of efficiency and effectiveness. They are able to diagnose diseases, repair electrical networks, control industrial workplaces, create geological maps, etc. Representation of knowledge is difficult because an expert knowledge can be imprecise and/or uncertain. In general, the knowledge is represented as a large set of simple rules. Conclusions are generally obtained through the inference process. The expert systems have been pioneers in the field of knowledge-based systems. They replace one or more experts for problem solving. In many situations, they may be more useful than traditional computer-based information systems. There are many circumstances when they become particularly useful: when an expert is not available, when expertise is to be stored for future use or when expertise is to be cloned or multiplied, when intelligent assistance and/or training are required for decision-making or problem-solving, or when more than one expert's knowledge has to be stored on one platform. All these situations make them very useful nowadays, and thus, it is very important to improve their performance and usability. The improvement may concern both the structure of the knowledge base and the inference algorithms.

2.1. Rule-Based Knowledge Bases. Among various methods of knowledge representation, rules are the most popular form.

Rule-based knowledge representation uses the Horn clause form: "if premise then conclusion." This is one of the most natural ways for domain experts to explain and present their knowledge. Activation of the rules during the inference process results in adding their conclusions as new facts (new knowledge). Let us assume that the knowledge base KB is a set of N rules: $\mathrm{KB} = \{r_1, r_2, \dots, r_N\}$. Every rule $r \in \mathrm{KB}$ has a form $r = \mathrm{cond}_1(r) \wedge \mathrm{cond}_2(r) \wedge \cdots \wedge \mathrm{cond}_m(r) \longrightarrow \mathrm{concl}(r)$, where $\mathrm{cond}_1(r) \wedge \cdots \wedge \mathrm{cond}_m(r)$ is the conjunction of the rule's conditions (premises) and $\mathrm{concl}(r)$ is the conclusion of the rule r.

Rules may be generated automatically using one of many possible algorithms based on the machine learning techniques. The knowledge base can be composed of different types of rules: classification rules, association rules, regression rules, or the so-called survival ones [12]. In addition, the rule set can be obtained by transforming the decision tree [13]. They also can be given by experts, but such process is a very difficult task. Usually, the value of experts' knowledge is rated so highly that experts are reluctant to share it. Therefore, to carry out the right number of experiments, it was decided to use the knowledge base with rules generated automatically from data shared within the UCI machine learning repository [14]. An efficient algorithm for generating rules automatically from data is the LEM algorithm [15]. It is based on the rough set theory [16–18] and induces a set of certain rules from the lower approximation (lower approximation is a description of the domain objects that are known with certainty to belong to the subset of interest), and, respectively, a set of possible rules from the upper approximation (upper approximation is a description of the objects that possibly belong to the subset of interest). This algorithm follows a classical greedy scheme which produces a local covering of each decision concept. It covers all examples from the given approximation using a minimal set of rules.

The procedure for preparing knowledge bases for this work was as follows. Each selected set of data from the repository was rewritten as a decision table, which was then subject to the process of rule induction (LEM2 algorithm) using the RSES tool [19].

As an example, let us take a heart disease dataset [20], which originally contains 303 instances, described by 14 nominal and numerical attributes (age: in years, sex: (1 = male; 0 = female), cp: chest pain type with values (1): typical angina, (2): atypical angina, (3): nonanginal pain, and (4): asymptomatic and others). The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

The piece of the original dataset is as follows:

```
   63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0,
2.3,3.0,0.0,6.0,0
   67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0,
1.5,2.0,3.0,3.0,2
   67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0,
2.6,2.0,2.0,7.0,1
   37.0,1.0,3.0,130.0,250.0,0.0,0.0,187.0,0.0,
3.5,3.0,0.0,3.0,0
   41.0,0.0,2.0,130.0,204.0,0.0,2.0,172.0,0.0,
1.4,1.0,0.0,3.0,0
   56.0,1.0,2.0,120.0,236.0,0.0,0.0,178.0,0.0,
0.8,1.0,0.0,3.0,0
```

A knowledge base with 99 rules has been achieved. The source file is as shown in Sourcecode 1.

The rule `(blood_sugar=0)&(angina=0.0)&(thal=3.0)&(sex=0)&(pain_type=3.0)=>(disease=1[23]) 23` should be read as: *if* (blood sugar = 0) and (thal = 3.0) and (sex = 0) and (pain_type = 3.0) *then* (disease = 1) which is covered by 23 of the 303 instances in the original dataset (8% of 303 instances cover this rule).

When the size of the input data (which rules are to be generated from) increases, the number of generated rules does too. Let us look at the *diabetes* data set [14]. It contains the data for 768 objects described with 8 continuous attributes. Processing the data with LEM2 and RSES with an implementation of the LEM2 algorithm, 490 rules have been created. For the *nursery* dataset, which originally contains 12,960 instances described with 9 conditional attributes, 867 rules have been generated. Such numbers make it difficult or even impossible to be analysed by a person. It is also important to note that the generated rules might have a varying number of premises. It can be said that the fewer premises a rule has, the easier it is to determine if it is true (it requires less number of conditions to cover). On the other hand, making a decision dependent on the highest possible number of conditions may suggest that if all the conditions have been met, the decision must be correct.

When looking globally at a knowledge base with rules, it turns out that it may contain a large number of short rules (with one premise or a few) but also some rules described with a large number of premises with only a few premises that differentiate them. This, in turn, brings about various problems at the rule analysis stage in the inference process. When there is a set of many long rules (described with several premises) which differ from one another by a single premise, it can extend the inference process which then attempts to check all the rules which are deemed fit to be activated. Another possible outcome might be that in a given knowledge base there is an uneven distribution of rules connected with given premises. This may result in a large group of rules dedicated to one area only and one or very few rules describing other areas of the domain (the particular part of the

```
RULE_SET heart_disease
ATTRIBUTES 14
agenumeric 1
sex numeric 1
.....
diseasesymbolic
DECISION_VALUES 2
2
1
RULES 99
(blood_sugar=0)&(angina=0.0)&(thal=3.0)&(sex=0)&(pain_type=3.0)=>(disease=1[23]) 23
(blood_sugar=0)&(angina=0.0)&(thal=3.0)&(no._of_vessels=0)&(sex=0)&(electrocardiograph=0.0)
=>(disease=1[22]) 22
....
...
(blood_sugar=0)&(sex=1)&(electrocardiograph=2.0)&(angina=0.0)&(pain_type=1.0)&(age=42)
=>(disease=1[1]) 1
(blood_sugar=0)&(sex=1)&(electrocardiograph=2.0)&(no._of_vessels=0)&(thal=7.0)&(angina=1.0)&(age=53)
=>(disease=1[1]) 1
```

Sourcecode 1

domain has not been sufficiently explored). Finding rare rules might become a nontrivial task. When taking into consideration the matter of big sets of often dispersed rules, it turns out that for the effectiveness of the inference processes, decision support systems founded on rule-based knowledge representation should be equipped with rule management mechanisms. In other words, they are methods and tools which help to review the rules effectively and quickly find those to be activated. One of the available solutions is rule clustering. In the subject literature, this issue has been extensively described and most of the time it focuses on cluster analysis [21]. Assuming every rule cluster as a group of similar rules, it is possible to create its representative as a set of all the features that describe the group in the best possible way. Let us imagine there is a knowledge base with a large number of rules which are subject to clustering. As a result, there will be a structure of groups of rules which are similar to one another. The extent of cohesiveness of a knowledge base will translate into the number and size of the resulting clusters of rules. There are several possible scenarios: a small number of clusters which contain a large number of rules in each of them or a large number of clusters which contain a few rules in each of the clusters. Of course, the scenarios described above are at the extreme ends of the scale. However, the generated structure of clusters may be well-balanced where each cluster contains a comparable number of rules and the number of rules is close to the size of each cluster (e.g., if there are 100 rules which are divided into 10 clusters with 10 rules in each).

Subsequently, the effectiveness of the knowledge extraction from rule clusters depends on the rule cluster quality and the efficiency of inference algorithms. For rule clusters, we create representatives and they are then searched in the process of inference. Due to the fact that the quality of representatives and the optimization of inference processes are so important, better solutions are still being sought.

To make the rule activation process possible, apart from the gathered knowledge, an inference mechanism is necessary. The following subsection presents the definition of inference and a short description of the existing inference algorithms and discusses the parameters and the inference control strategies.

2.2. Inference Algorithm. An inference engine is a software program that refers to the existing knowledge, manipulates the knowledge in line with needs, and makes decisions about actions to be taken. It generally utilizes pattern matching and search techniques for conclusions. Through these procedures, the inference engine examines existing facts and rules and adds new facts when possible. There are two common methods of deriving new facts from rules and known facts. These are data-driven (forward chaining) and goal-driven (backward chaining) inference algorithms. The most popular one, with respect to the usability in real-life applications, is the data-driven algorithm based on the *modus ponens* rule—a common inference strategy. It is simple and easy to understand [22]. The framework can be given as follows: the rule states that when A is known to be true and a rule states "if A, then B," it is valid to conclude that B is true.

The data-driven algorithm starts with some facts and applies rules to find all possible conclusions. It is applicable when the goal of inference is undefined. The inference with a given goal is provided until this goal is considered as a new fact. The case in which there are more than one possible rule to activate, in a given iteration of the inference algorithm, is called in the literature *a conflict set*, and the method which deals with the issue is called the conflict set resolution strategy [23]. It should be emphasized, especially in case of a big dataset, that such situation occurs very often. There are many possible strategies proposed in the literature, but the most popular ones are to use the *FIFO* (First In First Out) or *LIFO* (Last In First

Out) techniques familiar in programming languages. When there are many rules and facts involved in an expert system, classic inference algorithms become ineffective. Inference times become unacceptable, and the number of newly generated facts exceeds the limit of the new knowledge that can be properly absorbed.

In such cases, it is necessary to find new inference algorithms which ensure effective management of the analysis process for rules to be activated. One may also consider changing the structure of the knowledge base with the rules to organize them in a specific and well-described structure so that later its search would be effective.

In this paper, the author continues her research on modification of a knowledge base structure with rules into a hierarchical one where the quality of representatives of the created rule clusters is as important as the quality of these clusters.

Therefore, the author proposes the following method of optimization. At the first stage, the knowledge base structure is modified. In the classic approach where the knowledge base is a set of rules written without any specific order, it is necessary to search the entire set of rules. The author proposes to cluster the rules with similar premises into the rule clusters. Among various methods, the agglomerative hierarchical clustering algorithm is used in this research (the author has also studied the use of other algorithms [10]). Its classic approach assumes merging, in every iteration, the two most similar rules or groups of rules into one group. The proposed modification of this approach is based on finding the optimal moment to cut the created hierarchical structure of rules. It should be finished when there is not enough similarity between the rules or groups of rules which remained to be clustered. Details of the proposed approach are presented in the following section.

3. Rule Clustering

Too many rules in the knowledge base can negatively affect the effectiveness of management of rules. One of the ways of managing the rules is to cluster them into groups and to describe the groups by their representatives. Each cluster is described using a so-called group representative (Profile). The notion of cluster analysis indicates that objects in the analysed dimension are split into clusters which collect the objects most similar to one another and the resulting clusters are as different as possible [21]. The optimal structure of rule clusters assumes a maximum internal similarity and a minimal external similarity between groups of rules. It guarantees an optimum internal cohesion and external separateness of clusters. In the next subsection, the author briefly introduces other clustering methods.

3.1. A Short Characteristic of Clustering Algorithms. Within the scope of cluster analysis algorithms, it is possible to select either partitional (sometimes called k-optimizing algorithms, as exemplified by k-means) or hierarchical algorithms (which provide additional knowledge about the order of clustering the most similar objects together, e.g., the agglomerative hierarchical clustering algorithm (AHC)). Both partitional and hierarchical algorithms utilize the distance or similarity measurement in the process of finding similar objects. Moreover, there are algorithms based on the intracluster density (DBSCAN [24] and OPTICS [25]) and, most recently, spectral analysis algorithms (SMS (spectral mean shift) [26]).

Assuming that clustering is an automated process performed on a random set of rules with an unknown structure, the best solution which helps to avoid other possible problems is to use a hierarchical algorithm. The above-mentioned problems are, among others, an inability to determine an optimum number of clusters (necessary for partitional algorithms), the need to separate rare objects (rules) from the created clusters, and a motivation to gain additional knowledge on the sequence of rule clustering so that for each rule, another most similar rule or cluster can be found. In the density-based algorithms, similarly to partitional algorithms, additional clustering parameters like a minimum proximity threshold or the number of objects in a cluster need to be defined. The agglomerative hierarchical clustering algorithm (AHC) is free of such limitations [9, 10]. This algorithm has many modifications which vary from the original with respect to a changing stop condition of the clustering process.

3.2. Agglomerative Hierarchical Clustering Algorithm. The author proposes the clustering of rules with similar premises which produces a hierarchical structure (dendrogram). In the classic form of the agglomerative hierarchical clustering algorithm (AHC), the clustering process of individual rules should be continued until a single cluster of rules is obtained with a reservation that at each step a cluster is created by joining pairs of the most similar rules or clusters of rules. Accordingly, for the N number of rules in a knowledge base, the number of the algorithm's iterations is equal to $N - 1$. It is easy to notice that for numerous knowledge bases the inference's duration time might be a problem. This is an unacceptable feature for big knowledge bases, and modifications which reduce the number of iterations are welcome.

3.3. Clustering Parameters. There are various clustering parameters that help to achieve optimal clustering results. In this research, the author has analysed such parameters as similarity measures, the number of clusters to create, and clustering methods.

3.3.1. Similarity Measures. Clustering of similar objects requires that similarities (or distances) between the object be defined. In the literature, there is a lot of research devoted to the analysis of available measures of similarity and dissimilarity of objects [27, 28]. These measures (in this paper) have been used to determine the similarities of rules between one another as well as the similarities of rules and clusters of rules in relation to the cluster representatives. The same measures can be subsequently used to measure the similarity of representatives for clusters of rules and facts in the inference process. To provide the universality of the solution, both the single rules and clusters use the conjunction of pairs which consist of an attribute and its value. The values of attributes may be symbolic and continuous.

Generally, a similarity value for a pair of rules r_i and r_j which belong to a set of rules R is calculated in the following way:

$$\mathrm{sim}\left(r_i, r_j\right) = \sum_{f=1}^{m} w_f * \mathrm{sim}_f\left(r_{if}, r_{jf}\right), \tag{1}$$

where sim_f is a similarity value between two rules r_i and r_j in relation to the f − th attribute and the value w_f is the weight of the attribute a_f (usually determined as $w_f = 1/d$ for $f = 1, \dots, d$, where d is the number of attributes). Alternatively, weights 0 and 1 can be used for attributes (where 0 for the f − th attribute's weight means that the attribute does not appear in the rule while 1 means that a given attribute constitutes the rule's premise part). The similarity value can be obtained by using one of a various possible similarity measures. The author dealt with the influence of measures of similarity on the clustering quality in [29, 30]. In [29], nine various measures were described and analysed: SMC (simple matching coefficient) and its modification wSMC (weighted simple matching coefficient), Gower's measure (widely known in the literature), two measures used for information search in large text files (OF and IOF) and four measures based on the probability of occurrence for a given feature in the description of a rule or a group of rules (Goodall's measures) [27, 28]. In this research, the author uses the same set of similarity measures (in the experimental stage, each of these methods was used). The measures have been widely described by the author in [29, 30]; therefore, the issue is not discussed again in this work.

For example, the similarity value sim_f based on the wSMC equals 1 if rules r_i and r_j contain the same value for the f attribute. Otherwise it equals 0. Hence, only if rules r_i and r_j contain the same values for the every attribute in their premises and weight w_f is determined as $w_f = 1/d$ for $f = 1, \dots, d$ and d is the number of attributes, then the similarity value $\mathrm{sim}(r_i, r_j)$ equals 1. If the rules differ at least for one attribute, the value is less than 1. Value 0 for $\mathrm{sim}(r_i, r_j)$ (in case of wSMC similarity measure) means that there was not even one attribute for which rules r_i and r_j would have the same value.

Some of the analysed measures determine the similarity of rules using the frequency $f(r_{if})$ of occurrence of a certain pair of attributes and its values in the entire set of rules ($f(r_{if})$ denotes the number of times a premise r_{jf} appears in rules), while others are based on probabilities $p_f(r_{if})$ ($p_f(r_{if})$ denotes the sample probability of the case when a premise r_{if} appears in rules: $p_f(r_{if}) = f(p_f(r_{if}))/N$).

3.3.2. Number of Clusters. To determine an optimum similarity threshold might be impossible if the algorithm needs to be made independent from the type of data. It must be remembered that when similar rules are to be clustered, the threshold has to be set up at a reasonably high level or the clustering within a knowledge base can be initiated for rules which are practically dissimilar to one another and it might be impossible to reach a high level of similarity. In [9, 10], the author has presented an approach based on the termination of clustering when the intercluster similarity is no greater than the intracluster similarity. Unfortunately, the computations required for this approach are too burdening as far as the clustering algorithm is concerned. Another solution is the termination of clustering at a certain level as an attempt to force upon the number of clusters. Then, the AHC algorithm joins the rules and their clusters as long as the assumed number of clusters is reached. The above-described solution is presented in this paper.

In the literature, there are multiple papers which deal with the issue of an optimum selection of the number of clusters in the clustering algorithms [31, 32]. The most prevalent approach to be found in these papers underlines the necessity to perform numerous iterations for a gradually changing number of clusters and then choosing an optimum solution. Theoretically, it means that the number of possible partitions for a knowledge base with N rules equals N because, having 5 rules to cluster, we may place every rule in 1 or 2, 3, 4 and even into 5 clusters. Of course, the first and last solutions do not make sense (we would achieve one big cluster with an entire set of rules or 5 singular rule clusters). For this reason, the starting parameter value pertaining to the number of groups is 2 and increases by 1 in every partition until the number of clusters is smaller than the number of rules. If numerous knowledge bases are concerned, such an approach would not be time-effective.

The author has attempted to propose heuristics which help to determine an optimum number of clusters. The number of clusters K to be created is calculated with respect to the equations $K_1 = \lceil \sqrt{N} + i * \%N \rceil$ and $K_2 = \lceil \sqrt{N} - i * \%N \rceil$. K_1 and K_2 are the numbers of clusters to create, and N denotes the number of rules. It is easy to see that the modification consists in the clustering for a gradually changing (one step at a time, iteratively relative to the variable i, for $i = 1, 2, \dots$) parameter K. Such a solution makes it possible to find the optimal number of clusters to create and does not require checking all possible scenarios but only some of them. For example, in case of a heart disease dataset with 99 rules, all the possible rule partitions, based on the proposed heuristics, are as follows: $K = 1, \dots, 20$. Hence, instead of generating 99 different rule partitions, only 20 are created and analysed.

3.3.3. Clustering Methods. In this paper, the author has used four most popular methods as found in the literature. The first of them, the single-link method (SL), measures the distance between clusters R_p and R_q as a minimum distance between a random pair of rules r_i and r_j where $r_i \in R_p$ and $r_j \in R_q$. The second one is called the complete-link method (CL) and defines the distance between the cluster R_p and R_q as the longest distance between any two objects in two clusters.

```
Algorithm 1: hierarchical clustering for rules
Data: $KB = \{r_1 \dots r_N\}$ – rules from knowledge base; $K$ - number of clusters to create;
Result: $PR = \{R_1, R_2, \dots, R_K\}$ - the structure of a $K$ number of clusters of rules; $Profiles(PR) = \{Profile(R_1), Profile(R_2), \dots, Profile(R_K)\}$ - a set of representatives for these clusters;
begin
    /* create a clusters structure $PR := \{R_1, R_2, \dots, R_N\}$ in which each cluster $R_i = \{r_i\}$ is a single cluster, $i = 1, 2, \dots, N$ ; */
    $M := N$;
    while $M > K$ do
        /* create similarity matrix $S_{M \times M}$ for all clusters of rules $R_j, R_l \in PR$ in which every cell $s[j,l]$ contains similarity value for a pair of clusters $R_j$ and $R_l$: */
        $s[j,l] := sim(Profile(R_j), Profile(R_l))$ ;
        /* find a cell with the maximum value of similarity */
        $(j,l) = \arg\max_{1 \leqslant j,l \leqslant M}\{s[j,l]\}$;
        /* create a new cluster $R_q$ (and its representatative $Profile(R_q)$) which contains clusters $R_j$ and $R_l$: $R_q := R_j \cup R_l$. Remove clusters $R_j$ and $R_l$ from the $PR$ and add $R_q$ to $PR$: */
        $PR := PR \cup R_q \setminus \{R_j, R_l\}$;
        $M := M - 1$;
    end
    /* return $PR$ and $Profiles(PR)$; */
end
```

```
Algorithm 2: data-driven inference algorithm
Data: $PR = \{R_1, R_2, \dots, R_K\}$ - $K$ clusters of rules; $F = \{f_1, f_2, \dots, f_f\}$- set of facts; $goal$ - goal of the inference;
Result: $F_{new} = \{f_1, f_2, \dots, f_p\}$ - set of new facts explored from $PR$
begin
    boolean $stop = false$, $result = false$, $F_{new} = \{\emptyset\}$, $iterationCounter = 0$;
    if $goal! = null$ then
        foreach $fact\ f_h \in F$ do
            if $f_h == goal$ then
                $stop = true$ ;$result = true$;
            end
        end
    end
    while $stop == false$ and $iterationCounter < 2$ do
        /* Find clusters of rules -the most relevant to the set $F$ */
        foreach $cluster\ R_i \in PR$ do
            calculate $sim(Profile(R_i), F)$;
        end
        $Facts_{counter} = 0$; $R_{relevant} = R_i : sim(R_i, F)$ is maximal;
        if $R_{relevant}! = null$ then
            $iterationCounter$ ++;
            activate($R_{relevant}$);
            $F_{new} := F_{new} \cup concl(R_{relevant})$;
            foreach $fact\ f_g \in F_{new}$ do
                if $goal! = null$ then
                    if $f_g == goal$ then
                        $stop = true$;$result = true$;
                    end
                end
            end
        end
        else
            $stop = true$;$iterationCounter = 0$;
        end
    end
    return $result$;
end
```

Figure 1: The pseudocodes of the hierarchical clustering algorithm for rules and the data-driven algorithm for rule clusters.

There are two more methods known in the literature—the average link method and the centroid link method. The former, marked as AL in this paper, measures the distance between the luster R_p and R_q as an average distance of all pairs of objects located within the examined clusters. The latter, marked in this paper as CoL, always calculates the distance between the clusters R_p and R_q as a distance between their centroids. A centroid is a pseudo-object whose attribute values are mean values of all objects in the cluster.

4. Proposed Approach

Having obtained groups which consist of similar rules, in fact only a small part of the knowledge base is searched. The previous object-by-object analysis, where the searched objects need to match the knowledge in the most possible way, can be reduced to matching the input data to each cluster's representative and selecting the best matching representative.

4.1. Hierarchical Structure of a Knowledge Base. As the resulting structure is one or more binary trees with M number of nodes, it is easier to reduce the computing complexity of the inference algorithm from the linear to the $\log_2 M$ complexity as the former emerges from the necessity of review of all rules in the knowledge base in order to find a set of activable rules. The knowledge base's structure with rule clusters shall be defined as a sorted pair $(\mathrm{PR}, \mathrm{Profiles}(\mathrm{PR}))$ where $\mathrm{PR} = \{R_1, R_2, \dots, R_K\}$ represents the structure of a K number of clusters and $\mathrm{Profiles}(\mathrm{PR}) = \{\mathrm{Profile}(R_1), \mathrm{Profile}(R_2), \dots, \mathrm{Profile}(R_K)\}$ constitute a set of representatives for these clusters (for $K \ll N$). The following two conditions must be met: $\bigcup_{j=1,2,\dots,K} R_j = \mathrm{KB}$ and $R_l \cap R_j = \varnothing$ for $j \neq l$ and $j, l = 1, 2, \dots, K$. A *hierarchical knowledge base* contains a structure of clusters of rules together with their representatives. As a result of the application of the AHC algorithm with a set criterion of stopping the agglomeration, we get a number of clusters (equal to K) containing other rule clusters or single rules. This structure is then searched in the inference process.

4.2. Agglomerative Hierarchical Clustering: A Proposed Approach. The pseudocodes of the hierarchical clustering algorithm for rules and data-driven inference algorithm for rule clusters are presented in Figure 1. Iteratively, until a given number of clusters (K) is not achieved, at every step of the clustering process, we create a similarity matrix for all rule clusters. Each cell contains a similarity value for a pair of rule clusters R_l and R_j. Then, we have to choose a matrix cell with the biggest similarity. At the end of each iteration, we create a new cluster R_q which contains the merged clusters R_l and R_j and we remove the clusters from the structure PR and add the new cluster R_q to it. The cluster analysis in effect produces fairly homogeneous groups of rules together with their representatives.

4.3. Knowledge Extraction in Rule Clusters. The decision-making process consists of extraction of new knowledge based on both the rules in a knowledge base and the facts. Since the rules have been merged into groups, the inference process must apply to the rule clusters. The idea proposed by the author is based on the method widely known in the literature within the domain of retrieval information systems and searching within hierarchical structures. Rule clustering with the AHC algorithm creates a hierarchical structure in the form of a dendrogram. A similar structure was obtained in the SMART system [33] where textual documents were subject to clustering. The clusters therein were defined as such sets of documents where each item is similar to all the remaining parts of the set. The obtained hierarchy of documents was then searched through analysis of the similarity between the groups' representatives and the given query. At each level of the hierarchy, the most similar group was chosen. The process ended when the most relevant group (document) was found [34]. The objective of the procedure is to maximise the search efficiency by matching a request with only a small subset of the stored documents, at the same time minimizing the loss of the relevant documents retrieved in the search. It is necessary to remember that cluster representatives are analysed; thus, the efficiency of searching within documents depends on the quality of the representatives. There are many possible ways to build a cluster representative. For example, document clusters can be represented by the set of the features most common for all the documents in a given cluster. The representative can be general or specific, which is very important in the context of inference efficiency. General representatives as a short type description may be easy to analyse but take more time to find a given document. Specific representatives contain usually many features in their descriptions and thus it takes much more time to analyse one representative, but usually we can easily find a given document.

In this project, the author works with rules in a knowledge base which are a very specific data type and thus require a specific way to manage them properly. They may have different lengths and may contain not only different attribute values but, above all, completely different attributes, which significantly affect the ability to compare them and to look for similarities.

4.4. Rule Clusters' Representatives. When a set of clusters has been generated, it is possible to construct a representative classification vector for each cluster, called a *centroid vector*, such that the property assignment of the centroid reflects the typical, or average, values of the corresponding property values for all elements within each given cluster. Various methods can be used to generate the centroid vectors. Considering the fact that rules in a knowledge base are a specific type of data and most of the time those rules are recorded with various types of data, the author proposes an approach which considers both nominal and numeric features in a representative's description. To find out which form of a representative (general or detailed) provides a greater effectiveness of the resulting structure and inference processes, the author proposes several different approaches. It should be noticed that in her previous research [11], the author analysed also other methods of generating cluster representatives. Each rule cluster $R_q \in \mathrm{PR}$ is assigned a representative called a *profile* ($\mathrm{Profile}(R_q)$). In the basic approach (further referred to as the threshold approach), a representative consists of all such attributes which have appeared in k% of rules in a given group (default $k = 30\%$):

$$\begin{aligned}\mathrm{Threshold}(R_q) = \cup\{p_s : \mathrm{frequency}(\mathrm{getAttr}(p_s)) \\ \geq k \text{ for } p_s \in \mathrm{cond}(r_i), r_i \in R_q\},\end{aligned} \tag{2}$$

where $\mathrm{frequency}(\mathrm{getAttr}(p_s))$ returns the number of times when the attribute of a given premise p_s appears in the conditional part of all rules in the group R_q. If a given attribute reaches a set threshold then, depending on its type, its value (for symbolic features) or a mean (for numeric features) is added to the representative.

As this method analyses only the attribute part in pairs (attribute, value), the accuracy of the searching process may not be as precise as it is for other methods. Finding similar representatives with this technique means only that a rule cluster containing a given attribute has been found.

The conditional and decision parts of every rule are created from a given set of pairs (attribute, value). For the following set of attribute $A = \{a, b, c, d, e, dec\}$ and their values $V_a = \{a_1, a_2, a_3\}$, $V_b = \{b_1, b_2\}$, $V_c = \{c_1, c_2\}$, $V_d = \{d_1, d_2\}$, $V_e = \{e_1, e_2\}$, and $V_{dec} = \{A, B, C\}$, we may consider a few different scenarios (for simplicity's sake, in the example let us assume that all the attributes are at a nominal scale). For the knowledge base $\mathrm{KB} = \{r_1, r_2, r_3, r_4\}$, the following rules are

$$\begin{aligned} r_1 &: (a, a_1) \wedge (c, c_2) \longrightarrow (dec, A), \\ r_2 &: (a, a_2) \wedge (c, c_2) \longrightarrow (dec, B), \\ r_3 &: (b, b_1) \longrightarrow (dec, C), \\ r_4 &: (a, a_3) \longrightarrow (dec, A). \end{aligned} \tag{3}$$

We may say that rule r_3 is unlike the others (it is described by other attributes) while rules r_1 and r_2 are quite similar because besides the same premise (c, c_2), they also contain a similar premise with an attribute a. Rule r_4 is (like rule r_3) unlike others, but looking only at the attribute part, we may say that it is more similar to rules r_1 and r_2 than rule r_3, containing an attribute a.

Assuming that the selected clustering algorithm will first join the rules r_1 and r_2 and then include rule r_4 in the same cluster, the representative created with the use of the threshold method (with a k parameter set to value 50%) is $\mathrm{Profile}(r_1, r_2, r_4) = \{(a, a_1), (a, a_2), (a, a_3), (c, c_2)\}$. Undeniable advantages of approximation of sets based on the rough set theory can be found in numerous papers such as [16–18]. The rough set is the approximation of a vague concept (set) by a pair of precise concepts, called lower and upper approximations. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset. Using the notions of lower and upper set approximation, a representative is created with the use of the lower/upper approximation method. The lower approximation method defines a cluster's representative as all pairs (attribute, value) which appear in the conditional part of each rule in the analysed cluster. Conversely, a cluster's representative designated with the upper approximation method shall contain all such pairs (attribute, value) which have appeared in the conditional part of at least one rule in the cluster. The definition of a lower approximation for a group's profile R_q is as follows:

$$\mathrm{LowerApp}\left(R_q\right) = \cup \left\{ p_s : \bigwedge_{r_i \in R_q} p_s \in \mathrm{cond}\left(r_i\right) \right\}, \tag{4}$$

and an analogical definition for the upper approximation method is

$$\mathrm{UpperApp}\left(R_q\right) = \cup \left\{ p_s : \bigvee_{r_i \in R_q} p_s \in \mathrm{cond}\left(r_i\right) \right\}, \tag{5}$$

where $\mathrm{cond}(r_i)$ means the conditional part of the $r_i -$ th rule, and p_s is a single premise in this rule r_i. The representative for rule cluster r_1, r_2, and r_4 using the lower approximation-based method regrettably contains an empty set, while using the upper approximation-based approach it contains the following features: $\mathrm{Profile}(r_1, r_2, r_4) = \{(a, a_1), (a, a_2), (a, a_3), (c, c_2)\}$. It is imprecise as it contains the features which cover less that 30% of the rules in a given group.

Hence, it seems justifiable to control the level of coverage of features selected for group representatives. It has led to an alternative way of creating cluster representatives, namely, the weighted representative method. In this method, giving weight (expressed as k%), a representative is created from all pairs (attribute, value) which have appeared at least in k% of rules in a given group.

$$\mathrm{Weighted}\left(R_q\right) = \cup \left\{ p_s : \bigvee_{r_i \in R_q} p_s \in \mathrm{cond}\left(r_i\right) \& \mathrm{frequency}\left(\left(p_s\right) \geq k\right. \right\}. \tag{6}$$

The representative of a group of rules r_1, r_2, and r_4 selected with the use of this approach (with a value of the k parameter set at 50%) is $\mathrm{Profile}(r_1, r_2, r_4) = \{(c, c_2)\}$ because only this particular premise appears in at least 50% of the rules in this group. This clearly shows the difference between the threshold and weighted approach. It must be emphasized that representatives of clusters are created promptly with clusters of rules, and as a result, there might be empty/blank representatives even though a cluster has been created. This happens when the representative designation method is too restrictive (capture conditions for some features in a representative are relatively high and difficult to fulfil) and simultaneously a stop condition has not been reached as the created structure still has more groups than the assumed threshold and the groups are continuously clustered. Such restrictive requirements are the traits of the lower approximation method. This method stipulates that a feature included in a representative's description is concurrently a common feature of all rules that constitute a cluster. This condition is usually too difficult to fulfil, especially when rules in a knowledge base are short and rarely have common premises. In consequence, at some stage (when groups are clustered into groups at a higher level of hierarchy), there are clusters without representatives. Such structures have to be avoided as they hinder a review of such group and making use of clustering as a tool in the exploration of knowledge bases. An excessive reduction of the conditions examined in the course of designation of representatives makes them too detailed and often inadequate for the described clusters. For instance, using the upper approximation method or setting up too low a threshold for the designation of representatives in the weighted or threshold representative methods (e.g., a 25% threshold) for a cluster of four rules, when a given feature is included as a premise in at least one rule, it is sufficient to be included in the cluster's representative.

4.5. Inference Process in a Hierarchical Knowledge Base. At the core of big data analytics is data science (deep knowledge discovery through data inference and exploration). A knowledge representation requires some process that, given a description of a situation, can use the knowledge to make conclusions. When the knowledge is properly represented, the inference reaches appropriate conclusions in a timely fashion. Thus, the knowledge must be adapted to the inference strategy to ensure that certain inferences are made from the knowledge. Inference in classic knowledge bases matches the entire set of rules to the known facts to deduce new facts. It is impossible to work on the entire set of rules and facts in case of big knowledge bases. Therefore, in this and previous research tasks [9], the author defines the model of the hierarchical knowledge base with rule clusters and rule clusters' representatives.

Inference in a hierarchical knowledge base involves using hierarchy properties to optimize the search of clusters of rules. The results of inference and the course of the inference process itself depend strongly on the goal of inference.

When considering the forward inference (data-driven), we need to take into account the inference with a given hypothesis to prove or without it. In the first case, we review the representatives of clusters of rules at each level and eventually select the rule or rule cluster most relevant to the given facts. If a selected rule can be activated, the result leads to the addition of a new fact to the knowledge base. When this new fact is simultaneously the goal of the inference, the process should end successfully. When the goal of the inference is not specified, we proceed as long as there are any rules that can be activated. Thus, as a result, the implemented inference algorithm leads to the exploration of a number of new facts, and one of the measures of inference efficiency is, among others, a percentage of new facts compared to the ones given at the beginning. The more new facts, the more effective the reasoning process is.

	r_1	r_2	r_3	r_4	r_5
r_1					
r_2	0.33				
r_3	0	0			
r_4	0	0	0.5		
r_5	0.25	0.5	0.5	0.33	

Step 1

→

	r_1	$r_2 \cup r_5$	r_3	r_4
r_1				
$r_2 \cup r_5$	0.25			
r_3	0	0		
r_4	0	0	0.5	

Step 2

→

	r_1	$r_2 \cup r_5$	$r_3 \cup r_4$
r_1			
$r_2 \cup r_5$	0.25		
$r_3 \cup r_4$	0	0	

Step 3

→

	$r_1 \cup r_2 \cup r_5$	$r_3 \cup r_4$
$r_1 \cup r_2 \cup r_5$		
$r_3 \cup r_4$	0	

Step 4

Figure 2: The course of the AHC clustering algorithm for a given knowledge base.

In the classic approach, premises of each rule are examined to see whether they match the set of facts. If they do, the rule is activated and its conclusion is added to the set of facts. If this new fact is a given hypothesis to be proved, the process ends successfully. If there is no given goal of inference, the process is repeated until there is at least one rule to be activated.

In the approach proposed in this research, only representatives of the created rule clusters are analysed, which significantly shortens the time of inference. Usually, the number of the created rule clusters is significantly smaller than the number of rules being clustered. However, the success of the inference process depends on the quality clustering and the approach to creating the representatives. For the structure of K clusters with their representatives, the inference process looks as follows. For the given set of input facts, we are looking at the representative clusters from the highest level in the created hierarchical structure, and at every level of the hierarchy, going from the root to the leaves, we choose the cluster most relevant to the facts. If the selected group is already a single rule, and all its premises match a given set of facts, then the rule is activated and its conclusion is added as a new fact to the knowledge base. If the new fact is simultaneously a given goal to be proved, the inference process is successful. Otherwise, the search process continues until the requested goal of the inference is confirmed or there are any rules to activate. It is easy to see that in the most optimistic case the process lasts only one iteration, during which one rule is activated and its conclusion matches the given goal of inference which ends the process successfully. Of course, the inference process succeeds also if the given hypothesis is proved in more than one iteration, or if any rule was activated (when no hypothesis was specified). For this reason, in the experimental stage, the author examined the following cases: was the goal specified, was it achievable, and was it eventually achieved? It was additionally examined whether any rule had been activated, how many rule clusters had been searched, and if an empty representative had occured during the searching process.

Verification of the correctness of the proposed solution consists of comparing the result of the inference for a hierarchical knowledge base with rule clusters with the result obtained for a classic knowledge base (without rule clusters) and classic inference (analyzing all the rules one by one). In the course of verification, it was checked how frequently the specified goal of inference had been confirmed or any new knowledge had been deduced from the rules and facts.

The pseudocode of the data-driven inference algorithm for rule clusters is presented as Algorithm 2 in Figure 1. The most important procedure is the one which makes it possible to find the most relevant (to the set F) rule cluster first and then the most relevant rule in the selected group. For each cluster R_i, its representative $\mathrm{Profile}(R_i)$ is compared to the set of facts F, and as a result, a group with the maximum similarity is selected ($i = 1, 2, \ldots, K$). The review time needed in the classic approach to search every rule is reduced to the time needed to search cluster representatives. Most of the time, K (number of clusters) is significantly smaller than N (number of rules). The selected rule is activated, and the inference process is finished successfully if the new fact is a requested goal of inference. If not, the process is continued.

4.6. Analysis of the Proposed Idea. For a structure containing about a thousand clusters of rules, about a dozen or so representatives will be compared to find the group which is most similar to the given information. Due to the logarithmic computational complexity of the algorithm, the more rules we group, the greater the time gain from browsing the cluster structure is. This is undoubtedly the biggest advantage of using this approach. Especially with big data sets, such solutions are particularly useful. The disadvantage may be the omission of other rules relevant to the given facts. This approach is more optimal in relation to the approach presented in the author's previous research [9, 10]. The optimization arises from the fact that if, at a given level of analysed structure of rule clusters, the group selected as more relevant contains other clusters (which means additional subsequent searches), we check if the other cluster (omitted at this level, less relevant) is not a single rule. If that is the case, and the premises of this rule match the facts, such rule is activated and makes it possible to finish the inference process earlier.

4.7. Example of Rule Clustering and the Inference Process for Rule Clusters. Let us assume that a given knowledge base contains five rules:

$$\begin{aligned} r_1 &: (a, a_1) \wedge (b, b_1) \wedge (c, c_1) \longrightarrow (dec, A), \\ r_2 &: (a, a_1) \longrightarrow (dec, B), \\ r_3 &: (d, d_1) \longrightarrow (dec, C), \\ r_4 &: (d, d_1) \wedge (e, e_1) \longrightarrow (dec, C), \\ r_5 &: (a, a_1) \wedge (d, d_1) \longrightarrow (dec, B). \end{aligned} \tag{7}$$

The course of the AHC clustering algorithm for this knowledge base, in case of using the wSMC similarity measure, is presented in Figure 2.

Table 1: The course of knowledge exploration for an example of knowledge base.

Step	LowerApp(R_i)	UpperApp(R_i)	Threshold(R_i)/Weighted(R_i)
Representative generation	$R_1 = \{\phi\}$ $R_2 = \{(d, d_1)\}$	$R_1 = \{(a, a_1), (a, a_2), (b, b_1), (c, c_1)\}$ $R_2 = \{(d, d_1), (e, e_1)\}$	$R_1 = \{(a, a_1), (b, b_1)\}$ $R_2 = \{(d, d_1), (e, e_1)\}$
Similarity between F and Profiles	$\mathrm{Sim}(F, R_1) = 0$ $\mathrm{Sim}(F, R_2) = 0$	$\mathrm{Sim}(F, R_1) = 0.25$ $\mathrm{Sim}(F, R_2) = 0$	$\mathrm{Sim}(F, R_1) = 0.5$ $\mathrm{Sim}(F, R_2) = 0$
Choosing the most relevant group	ϕ	R_1	R_1
Finding rule for activation	ϕ	$\mathrm{Sim}(F, r_1) = 0.33$ $\mathrm{Sim}(F, r_2) = 1$ $\mathrm{Sim}(F, r_3) = 0$	$\mathrm{Sim}(F, r_1) = 0.33$ $\mathrm{Sim}(F, r_2) = 1$ $\mathrm{Sim}(F, r_3) = 0$
Activated rule	ϕ	r_2	r_2
New facts	ϕ	(dec, B)	(dec, B)

Table 2: Inference efficiency vs. representative generation methods.

Representative generation method	New knowledge Less than 100%	 At least 100%	Goal not achieved[a]	Goal achieved
Threshold	23,145 (48.71%)	24,375 (51.29%)	40,657 (85.56%)	6863 (14.44%)
LowerApp	5692 (47.91%)	6188 (52.09%)	10,459 (88.04%)	1421 (11.96%)
UpperApp	6377 (53.68%)	5503 (46.32%)	9277 (78.09%)	2603 (21.91%)
Weighted	22,901 (48.19%)	24,619 (51.81%)	41,036 (86.36%)	6484 (13.64%)

[a] Empty representative found during inference.

As a result, two clusters of rules are generated: R_1 which contains rules r_3 and r_4 and R_2 which contains r_1, r_2, and r_5. The lower and upper approximation-based representatives for these groups are as follows:

$$\begin{aligned}
&\mathrm{LowerApp}(\mathrm{Profile}(R_1)) = \{(d, d_1)\},\\
&\mathrm{UpperApp}(\mathrm{Profile}(R_1)) = \{(d, d_1), (e, e_1)\},\\
&\mathrm{LowerApp}(\mathrm{Profile}(R_2)) = \{(a, a_1)\},\\
&\mathrm{UpperApp}(\mathrm{Profile}(R_2)) = \{(a, a_1), (b, b_1), (c, c_1), (d, d_1)\},
\end{aligned} \tag{8}$$

and there is also a given input set of facts $F = \{(a, a_1), (b, b_1)\}$. The course of the inference, taking into account the type of representatives, is presented in Table 1.

This basic example clearly illustrates how a representative generation method influences the efficiency of the inference process, producing different results. In case of the LowerApp method, no rule would be activated and no new knowledge would be extracted. When considering big data sets, one should bear in mind that the chosen cluster representation method can significantly affect the amount of new knowledge extracted from the knowledge base of hundreds or thousands of rules. The lower approximation method (producing general descriptions for rule clusters) unfortunately can make the process of discovering new knowledge from rules and facts impossible (because of empty representatives).

5. Experiments

The experiments were aimed at investigating whether the presented clustering methods (SL, CL, AL, and CoL) and representative generation methods (Threshold, LowerApp, UpperApp, and Weighted) influence the efficiency of inference and the quality of created rule clusters. The subjects of the experiments are four datasets: *heart*, *libra*, *weather*, and *krukenberg*, with various numbers of attributes and rules [14]. The smallest knowledge base contains 5 attributes and 5 rules and the greatest number of rules is two hundred, while the greatest number of attributes is 165 elements. In the experiments, many possible combinations were examined for each knowledge base: nine similarity measures, four clustering methods, and four representative generation methods with three various percentage thresholds and various numbers of clusters. The total number of experiment equals 178,200, and it results from the necessity of using all possible combinations of different similarity measures, clustering methods, cluster number, representative generation methods (with various values of threshold k), and the additional parameters related to the inference process such as a different number of facts and the cases with a given hypothesis to be proved or without any hypothesis. All tables summarize the results obtained for the whole 178,200 of the experiments performed.

Tables 2–4 present the results of the analysis of the influence of using various methods for representatives of rule clusters on the inference efficiency.

Table 3: The quality of rules clusters vs. representative generation methods.

Representative generation Method	BCS		O		ARL			BRL		
	Mean	SD	Mean	SD	Mean	SD	Min–Max	Mean	SD	Min–Max
Threshold	76.68	59.18	3.93	5.66	4.05	3.08	0.0–9.75	5.85	3.63	0.0–14.0
LowerApp	78.46	60.45	3.70	5.43	1.39	0.60	0.6–3.75	2.71	1.90	1.0–9.0
UpperApp	80.94	61.34	3.97	5.98	25.94	37.60	2.2–279.0	86.79	94.28	4.0–279.0
Weighted	77.72	59.21	3.85	5.57	4.13	3.59	0.0–14.5	6.83	6.83	0.0–19.0

Table 4: Description of inference efficiency vs. representative generation methods.

Representative generation method	Fired rules		Empty representative		New facts		Searched clusters	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Threshold	5.31	21.69	0.0	0.0	0.92	1.95	54.53	102.35
LowerApp	5.65	23.31	71.13	60.55	0.79	1.67	62.05	111.50
UpperApp	11.53	31.68	0.0	0.0	1.32	2.63	95.14	121.03
Weighted	4.64	19.93	30.45	48.67	0.82	1.69	52.52	101.17

Table 5: Inference efficiency vs. clustering methods.

Clustering method	New knowledge		Goal not achieved[a]	Goal achieved
	Less than 100%	At least 100%		
SL	14,721 (49.57%)	14,979 (50.43%)	24,941 (83.98%)	4759 (16.02%)
CL	14,182 (47.75%)	15,518 (52.25%)	25,122 (84.59%)	4578 (15.41%)
AL	14,517 (48.88%)	15,183 (51.12%)	25,795 (86.85%)	3905 (13.15%)
CoL	14,695 (49.48%)	15,005 (50.52%)	25,571 (86.10%)	4129 (13.90%)

[a]Empty representative found during inference.

Table 2 presents the frequency of finishing the inference successfully (the goal of the inference has been reached or/ and any new fact was induced from rules and facts already known) and the frequency of exploration of at least 100% of new knowledge (new facts) in accordance with the input knowledge. Table 3 presents a description of created clusters dependent on different representative generation methods in the form of the following factors: BCS (biggest cluster's size), O (the number of outliers), and ARL/BRL (average/biggest representative's length). Table 4 contains a description of inference efficiency presented as an average number of fired rules, an average number of empty representatives, and the average number of new facts as well as the number of the searched clusters. It is easy to observe that the representative generation method which allows confirming a given goal most often is the UpperApp method (in 21.91% of cases while the LowerApp method allows us to confirm the goal only in 11.96% of cases). If we aim to achieve a lot of new facts (new knowledge), then the representative generation method which allows to get the new knowledge exceeding 100% of input knowledge is the LowerApp method (in 52.09% of cases). The *New knowledge* column with the value *At least 100%* corresponds to the case where for a given set of input facts, at least the same number of new facts was generated during the inference process.

The UpperApp method generates the biggest cluster size, the greatest number of outliers, and a much wider range of representatives than it is in case of other representative generation methods. Only for the UpperApp and Threshold representative generation method are empty representatives not generated at all.

Tables 5–7 contain similar information as Tables 2–4 but for various clustering methods.

The SL clustering method makes it possible to confirm a given goal of inference most often. This method also generates the smallest size of the biggest cluster, the smallest number of outliers, and the shortest lengths of the generated representatives for the created rule clusters. The abovementioned method also yields the smallest number of fired rules, the earliest time of achieving empty representatives, and the smallest number of searched clusters.

6. Conclusions

The decision support systems founded on rule-based knowledge representation should be equipped with rule management mechanisms. Effective exploration of new knowledge in every domain of human life requires new algorithms of knowledge organization and searching of created data structures. Optimization proposed by the author in this paper is based on the cluster analysis method and modification of

Table 6: Quality of rule clusters vs. clustering methods.

Clustering method	BCS		O		ARL		BRL	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
SL	50.78	52.60	2.81	5.14	6.24	14.31	11.99	33.11
CL	83.01	64.37	3.14	5.49	6.26	14.06	14.22	40.03
AL	83.56	55.36	4.84	5.52	5.69	13.76	14.70	39.52
CoL	93.46	56.36	4.72	6.06	5.84	13.73	15.18	41.37

Table 7: Description of inference efficiency vs. clustering methods.

Clustering	Fired rules		Empty representative		New facts		Searched clusters	
method	Mean	SD	Mean	SD	Mean	SD	Mean	SD
SL	4.92	19.08	18.60	41.91	0.95	1.94	43.48	84.77
CL	5.60	22.26	20.06	43.39	0.91	1.88	63.75	110.60
AL	5.30	22.54	19.16	42.62	0.91	2.00	46.31	100.04
CoL	6.98	25.54	19.36	42.64	0.87	1.83	80.62	119.48

the inference algorithm, which searches within representatives of the created rule clusters instead of rules. This article presents both the description of the proposed approach and the results of the experiments carried out for the chosen knowledge bases.

Among various clustering algorithms, the agglomerative hierarchical clustering algorithm was selected with a modification proposed by the author in which rule clusters are built until a given number of clusters is reached. For every rule cluster, a representative is created. During the inference process, only representatives are analysed, and at every level of the created hierarchical structure, the most relevant representative is selected and further analysed. This means it is possible to search only a small part of the whole knowledge base with the same accuracy that would be achieved when the whole knowledge base is searched. During the previous experiments, it was shown that for big knowledge bases (with more than a thousand of rules), only 1.5% of the whole KB has to be analysed to finish the inference process successfully. For every combination of the clustering parameters such as similarity measures, number of clusters, and others—Tables 2–4 present the results of the described and examined methods of the cluster representative generation. Tables 5–7 present the results for four different clustering methods, respectively.

As expected, the UpperApp representative method corresponds with creating the biggest size and the largest representatives of the created clusters. As a result, this method leads to a successful conclusion more frequently. Therefore, it is recommended to consider further analysis of both the representative generation methods and the inference algorithm in order to propose new optimizations and achieve a higher efficiency.

Data Availability

The readers can access the data through the link: http://zsi.tech.us.edu.pl/~nowak/data.rar where original four datasets and four report files generated during the experimental stage are uploaded. The original knowledge bases and associated files with set of facts were used as input data for the CluVis software (implemented by the author) to build a hierarchical structure of every knowledge base and then to run the inference process. The results are report CSV-type files with inference efficiency measures such as factors calculated during the experiments.

Conflicts of Interest

The author declares that she has no conflicts of interest.

References

[1] http://news.mit.edu/2014/big-fast-weird-data.

[2] https://www.cnbc.com/2014/02/12/inside-the-wacky-world-of-weird-data-whats-getting-crunched.html.

[3] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, 2016.

[4] K. M. Wiig, *Expert Systems: a Manager's Guide*, International Labour Office, Geneva, Switzerland, 1990.

[5] R. Simiński and A. Nowak-Brzezińska, "Goal-driven inference for web knowledge based system," in *Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part IV*, vol. 432 of *Advances in Intelligent Systems and Computing*, pp. 99–109, Karpacz, Poland, 2015.

[6] T. Breidenstein, I. Bournaud, and F. Woliński, "Knowledge discovery in rule bases," in *Knowledge Acquisition, Modeling and Management*, vol. 1319, Lecture Notes in Computer Science, pp. 329–334, Springer, 1997.

[7] A. Hashizume, B. Yongguang, X. Du, and N. Ishii, "Generating representative from clusters of association rules on numeric attributes," in *Intelligent Data Engineering and Automated Learning*, vol. 2690, Lecture Notes in Computer Science, pp. 605–613, Springer, 2003.

[8] F. Ye, J. Wang, S. Wu, H. Chen, T. Huang, and L. Tao, "An integrated approach for mining meta-rules," in *Machine Learning and Data Mining in Pattern Recognition*, vol. 3587

of Lecture Notes in Computer Science, , pp. 549–557, Springer, 2005.

[9] A. Nowak, A. Wakulicz-Deja, and S. Bachliński, "Optimization of speech recognition by clustering of phones," *Fundamenta Informaticae*, vol. 72, no. 1–3, pp. 283–293, 2006.

[10] A. Nowak and A. Wakulicz-Deja, "The concept of the hierarchical clustering algorithms for rules based systems," in *Intelligent Information Processing and Web Mining*, pp. 565–570, Springer, 2005.

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

[12] Ł. Wróbel, M. Sikora, and M. Michalak, "Rule quality measures settings in classification, regression and survival rule induction — an empirical approach," *Fundamenta Informaticae*, vol. 149, no. 4, pp. 419–449, 2016.

[13] J. Stefanowski, "On rough set based approaches to induction of decision rules," in *Rough Sets in Data Mining and Knowledge Discovery*, L. Polkowski and A. Skowron, Eds., pp. 500–529, Physica, Verlag, Heidelberg, 1998.

[14] M. Lichman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Sciences, Irvine, CA, USA, 2013.

[15] J. W. Grzymala-Busse, "Rule induction," in *Data Mining and Knowledge Discovery Handbook*, pp. 249–265, Springer, Boston, MA, USA, 2nd edition, 2010.

[16] R. Slowinski, S. Greco, and B. Matarazzo, "Rough sets in decision making," in *Encyclopedia of Complexity and Systems Science*, pp. 7753–7787, Springer, 2009.

[17] A. Skowron, "Extracting laws from decision tables: a rough set approach," *Computational Intelligence*, vol. 11, no. 2, pp. 371–388, 1995.

[18] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough sets," *Communications of the ACM*, vol. 38, no. 11, pp. 88–95, 1995.

[19] J. G. Bazan, M. S. Szczuka, and J. Wróblewski, "A new version of rough set exploration system," in *Rough Sets and Current Trends in Computing*, pp. 397–404, Springer, 2002.

[20] R. Detrano, A. Janosi, W. Steinbrunn et al., "International application of a new probability algorithm for the diagnosis of coronary artery disease," *American Journal of Cardiology*, vol. 64, no. 5, pp. 304–310, 1989.

[21] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., 1988.

[22] B. G. Buchanan and E. H. Shortliffe, *Rule Based Expert Systems: the Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley Series in Artificial Intelligence)*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[23] C. L. Forgy, "Rete: a fast algorithm for the many pattern/many object pattern match problem," in *Expert Systems*, pp. 324–341, IEEE Computer Society Press, 1990.

[24] W. K. Loh and Y. H. Park, "A survey on density-based clustering algorithms," in *Ubiquitous Information Technologies and Applications*, Y. S. Jeong, Y. H. Park, C. H. Hsu, and J. J. Park, Eds., vol. 280 of Lecture Notes in Electrical Engineering, pp. 775–780, Springer, Berlin, Heidelberg, 2014.

[25] H. K. Kanagala and V. V. Jaya Rama Krishnaiah, "A comparative study of K-means, DBSCAN and OPTICS," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, Coimbatore, India, 2016.

[26] A. Dudek, "A comparison of the performance of clustering methods using spectral approach," in *Data Analysis Methods and Its Applications*, pp. 143–156, C.H. Beck, Warszawa, Poland, 2012.

[27] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: a comparative evaluation," in *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 243–254, Atlanta, GA, USA, 2008.

[28] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, p. 857, 1971.

[29] A. Nowak-Brzezińska and T. Rybotycki, "Comparison of similarity measures in context of rules clustering," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 235–240, Gdynia, Poland, 2017, IEEE Conference Publications.

[30] A. Nowak-Brzezińska and T. Rybotycki, "Impact of clustering parameters on the efficiency of the knowledge mining process in rule-based knowledge bases," *Schedae Informaticae*, vol. 25, pp. 85–101, 2017.

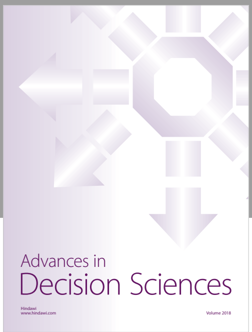
[31] Y. Jung, H. Park, D. Z. Du, and B. L. Drake, "A decision criterion for the optimal number of clusters in hierarchical clustering," *Journal of Global Optimization*, vol. 25, no. 1, pp. 91–111, 2003.

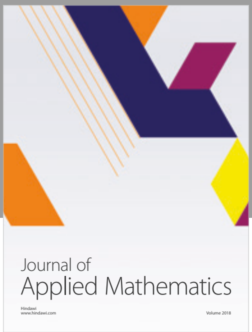
[32] S. Still and W. Bialek, "How many clusters? An information-theoretic perspective," *Neural Computation*, vol. 16, no. 12, pp. 2483–2506, 2004.

[33] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: the Concepts and Technology Behind Search*, Addison Wesley, 2011.

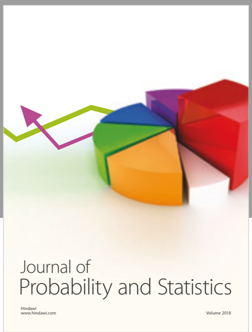
[34] J. J. Rocchio, *Document Retrieval systems – Optimization and Evaluation, [Ph.D. Thesis]*, Harvard University, 1966.

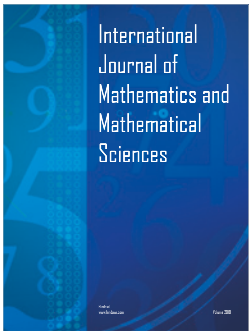
Advances in
Operations Research


Advances in
Decision Sciences

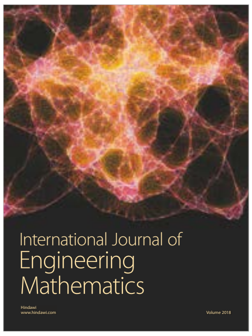

Journal of
Applied Mathematics

The Scientific
World Journal

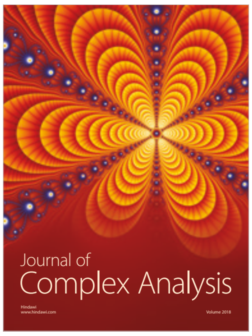

Journal of
Probability and Statistics

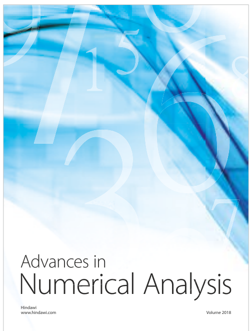

International
Journal of
Mathematics and
Mathematical
Sciences

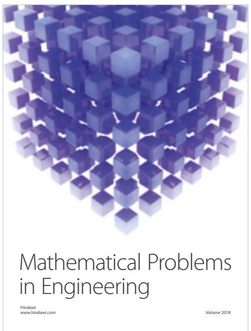
Journal of
Optimization

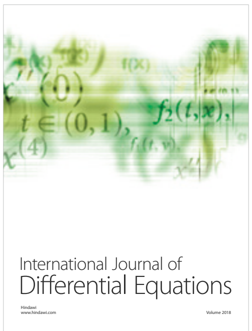

International Journal of
Engineering
Mathematics

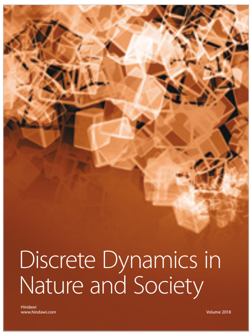
International Journal of
Analysis

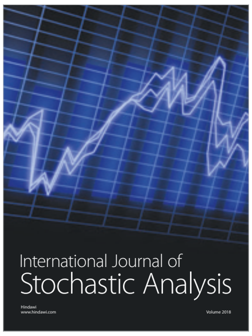

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

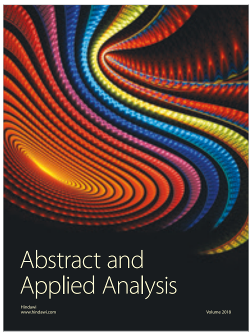

International Journal of
Differential Equations

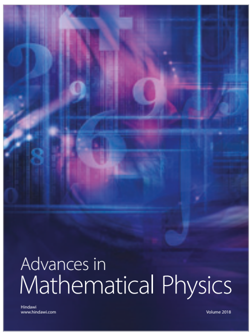

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics