On the Role of Feature Space Granulation in Feature Selection Processes

Marek Grzegorowski, Andrzej Janusz, Dominik Ślęzak, Marcin Szczuka
Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
Email: {m.grzegorowski,janusza,slezak,szczuka}@mimuw.edu.pl

***Abstract*—Information granulation plays an important role in the process of scaling up modern machine learning and knowledge discovery algorithms. By employing compact descriptions of granules – whereby granules are defined as collections of original data elements gathered together by means of their similarity, proximity or functionality – one can drastically accelerate computations and, moreover, make the results of those computations more meaningful for domain experts.**

In this paper, we summarize some of the feature space granulation approaches introduced by now. We discuss the meaning of similarity, proximity and functionality while considering the granules of physically existing or potentially derivable attributes. We also show several examples of utilization of the granulation structures defined over the feature spaces in the feature selection algorithms. As a case study, we consider the algorithms developed within the theory of rough sets, aimed at finding irreducible subsets of attributes that are sufficient to distinguish between the cases belonging to different target decision classes.

***Index Terms*—Feature Selection, Information Granulation, Rough Sets, Attribute Clustering, Attribute Hierarchies**

I. Introduction

Information granulation and the construction of a granular system for a given data set is frequently portrayed as a procedure of zooming in and out on the data or, in other words, changing the data "resolution". Depending on the chosen level of granularity, some data items (objects, cases, instances) become indistinguishable. Hence, the "length" of the data is altered, which corresponds to possible reduction of the storage and processing resources. Operating with *data granules* is common in physics, photography and many other fields. It becomes present in machine learning and data mining as well. It is also worth mentioning that the idea of zooming in and out – i.e., switching between different levels of information granularity – is popular in the area of online analytical processing (OLAP) in databases. However, one should remember that *data granularity* can have different meanings. In traditional databases, by *granular* data one usually means the most detailed, low level, exact data representation. On the other hand, in the field of granular computing (GrC), the term *granular* corresponds rather to the overall methodology of working with the granules of data or information.

The granular approach to dealing with (massive) information systems does not have to be limited to just the length/volume dimension of the data set. It can also be used to modify, reduce and transform the "width" and "depth" of information. In GrC this is sometimes called *variable granulation* and *concept granulation*. Just like in a case of the "classical" granulation, where data objects are combined into more complex entities, attributes in data can be granulated by using similarity, distance or correlation between them. In particular, by constructing granules over the space of attributes in the data set it is possible to reduce dimensionality. In the simplest form it can be used to replace multiple features/dimensions by just one representative of the corresponding granule. A more complex, yet still similar approach is represented by a reduction based on an information function and discernibility, typical for the theory of rough sets, where the original set of attributes is replaced by a reduct, i.e., a subset that carries the same amount of important information.

The attribute granules can take various forms. It is possible to group or cluster features on the basis of their relationship, and it can be done in a parameterized manner. For example, we can produce various versions of granulations depending on the choice of cutoff value after the original attributes are hierarchically clustered. In this context, it is important to have the means of assessment of the resulting granules, similar to those developed for standard data clustering. By making the feature selection process aware of the underlying granular structure of attribute space one can make better use of the knowledge contained therein. This in turn may lead to selecting the sets of features that are not only optimal from the perspective of some mathematical criteria but are also more useful for interpreting knowledge hidden in the data.

In this paper we discuss, using some examples of real-life applications, how the concept of granulation can be made useful in selecting and engineering features on large and possibly complex data sets. We show how to utilize the intrinsic properties of the data and underlying problem as well as background/domain knowledge for the purpose of building granular representation of attributes. All the provided tools and examples are devised to work with data sets that are very large in terms of the number of objects, as well as the number and complexity of features. Thus, we address at least some of the challenges posed by the Big Data paradigm.

Particular contributions made by this paper are concentrated around two aspects. First, we put forward a framework for expressing granules in an attribute space. Therein we include original ideas for discovering and managing similarities between attributes for the purpose of constructing granules. Feature granules can be induced by, e.g., hierarchical clustering

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

on attributes or analysis of so-called *heat maps* that convey the knowledge about attribute interchangeability. On the other hand, we show that meaningful granulations can be derived also according to other prerequisites, such as proximity or common functionality of the considered features.

The second contribution refers to a general algorithmic framework for performing feature selection on top of a granular representation of attribute space. Our methodology is devised in such a way that it caters for various types of granules and various goals of feature selection. The purpose is to perform a kind of granular attribute selection that exploits to the fullest semantical relationships between variables. The proposed methods are designed in such a way that it is possible to deal with large and complex data sets. By taking into account a given granulation of attributes, we can configure our algorithms to achieve faster convergence and, moreover, we obtain natural means to make use of efficient, parallelized computational schemes such as MapReduce.

The paper reviews the existing ideas related to handling massive and complex data sets by means of their granulation (decomposition, clustering, etc.) and simplification (reduction, summarization, etc.). It starts with outlining relevant prior research in Section II. In Section III, we lay foundation for our granular approach to feature selection by explaining how the granules can be formed and interpreted. Section IV brings the centerpiece of this work – a feature selection framework that is able to take into account the given attribute granulations. Finally, Section V outlines some ideas related to combining the proposed framework with the elements of iterative MapReduce. Section VI concludes the paper.

II. Related Work

While the approach presented in this paper was not considered in the past, at least not in the same extent, it is by no means detached from the current research in the areas of GrC and Big Data [1], [2]. In the domain of GrC there is much work devoted to understanding the granulation process and the underlying dependencies in data, which has also an influence on different ways of expressing the notions of relevance and redundancy in the considered spaces of attributes/features/variables/dimensions [3], [4]. Further in the paper we concentrate on various aspects of constructing granules gathering some subsets of the elements of such spaces and refer to several examples of the corresponding applications [5]–[8].

From the perspective of the Big Data analytics, an introduction of some hierarchies of granularity into the spaces of investigated attributes can make the feature selection and extraction processes more efficient. Tackling the complexity of large data sets is an issue noticed by many researchers [9], [10]. The typical challenges associated with Big Data, as symbolized by the presence of "Five Vs", make things even more complicated. Besides the complexity and scale of calculations that affect the required amounts of resources, the superfluous features may negatively influence the understanding of the data by the analysts, therefore, affecting their ability to monitor and tune the knowledge discovery processes [11], [12].

The demand for efficiency and effectiveness in Big Data scenarios resulted in a number of approaches to massively parallel feature reduction [13], [14], as well as highly scalable instance selection and deduplication [15], [16]. Popular code libraries like Spark or Mahout provide parallel implementations of well-known feature selection methods [17]. There are also approximate implementations of standard algorithms, which derive heuristic feature evaluation scores from granulated data summaries [18]. The speed of the feature and instance selection processes becomes especially important in interactive approaches [19], whereby, additionally, granular hierarchies of attributes may help the users to navigate through rich feature spaces. Introducing approximate computations into the feature selection processes is – in combination with making them highly parallel – an example of a more general trend in machine learning and knowledge discovery [20].

Identification of subgroups of similar variables is especially important for high-dimensional data exploration [21], [22]. In this context it is frequently useful to apply the modern algorithms aimed at big data clustering. Several instance clustering algorithms, like k-means [23] or expectation maximization [24], have already been implemented in the scalable environments. There are also some prior results reported on the feature clustering algorithms that are of particular interest in this paper [25]. The hierarchies of granules/groups of features can be constructed using some interactive clustering methods as well [26]. It is also important to realize that the feature similarity measures employed in the above clustering approaches should somehow correspond to the ultimate goal of finding the groups of attributes that can play mutually comparable roles in the constructed decision models [27], [28].

In the next sections we discuss the advantages of pregrouping of attributes from the perspective of feature selection, with particular emphasis on the reduct-based decision models originating from the theory of rough sets [29], [30]. It is noteworthy that, just as for other popular feature selection methods, there were some interesting attempts to perform reduct derivation within the MapReduce framework [31]. The ideas of scalable performance of feature extraction, in particular reduct calculation, are most commonly related to decomposing computations with respect to rows/instances [32]. However, by introducing the elements of granulation into the feature spaces we can additionally scale up the algorithms in an "attribute-oriented" fashion. Surely, such granulation-related ideas could be considered – besides the algorithms originating from the theory of rough sets – within the scope of other popular feature selection/engineering solutions as well [33], [34].

III. Prerequisites for Feature Space Granulation

In the GrC literature, a granule may have different meanings. One general definition of this basic notion is a collection of entities which are related through a similarity, proximity, indiscernibility or functionality [2], [35]. Indeed, if we consider a granulation of the attribute space, we may find examples of granules formed using each of those criteria.

In the context of attribute granulation, two attributes are usually regarded as similar if they convey similar information about objects described in the data. For instance, one may consider similar two attributes whose values in the data are highly correlated. In fact, Pearson and Spearman correlation coefficients are commonly used as measures of attribute similarity for the purpose of attribute clustering [25], [36]. There are, however, some other possibilities as well. For instance, further in this section we examine an idea of building similarity of attributes by means of their ability to replace each other in the constructed decision models. Namely, if an attribute can be replaced by another without losing important information about investigated objects, it means that they complement in the same way the remaining attributes.

The proximity of attributes may have a few meanings as well. Typically, this term is used as a synonym of similarity. However, when it comes to granules of attributes, it may also be understood as a "physical proximity". For example, in coal mines, there are many sensors monitoring the safety of miners, which constantly gather data about the conditions underground [11]. When analyzing this type of data, it is important to consider locations of sensors, since readings from closely co-located devices are inherently correlated [37]. Moreover, events observed by one group of sensors are detected by other groups after some time and the delay, as well as the order, in which different sensors denote the event, often correspond to the ventilation scheme of the mine. For this reason, as noted in [7], it is often worth to consider the whole chunks of attributes corresponding to such proximate sensors. In this way, it is not only possible to improve readability of the resulting decision models, but also increase the performance of the whole data processing chain due to a more efficient utilization of local buffers for reading data streams [38]. Another practical consideration is the aspect of model robustness and fault tolerance. In this context, proximity of attributes may be regarded as a degree of dependency on a specific hardware equipment. For instance, if one sensor is faulty, all attributes whose values are dependent on its readings will be unreliable.

It may also be desirable to consider granules of attributes that share some higher-level properties or that are tied by constraints imposed by a given application area [5]. Typically, domain experts associate such attributes with similar functionalities of investigated objects. Let us consider an example of the brain MRI data set investigated in [8], whereby features derived using some parameterized image processing procedures may be associated with groups of attributes that take different values for particular objects (these values depend on particular parameter settings) but describe the same aspect of the data. Another example of this type of situation is apparent in the analysis of a stock market. Many financial experts use technical indices to describe the behavior of stock prices in time. Such indices (e.g. moving averages, moving variance, RSI, TDI, stochastic oscillators and many more) have many parameters, such as the considered time window size. However, experienced traders would avoid including more than one or two realizations of any particular index in their models.

TABLE I
AN EXEMPLARY DATA TABLE $\mathbb{A}$ WITH A BINARY DECISION.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	d
u_1	1	2	2	0	0	1	0	1	1
u_2	0	1	1	1	1	0	1	0	1
u_3	1	2	0	1	0	2	1	0	1
u_4	0	1	0	0	1	0	0	1	0
u_5	2	0	1	0	2	1	0	0	1
u_6	1	0	2	0	2	0	0	2	0
u_7	0	1	1	2	0	2	1	0	1
u_8	0	0	0	2	1	1	1	1	0
u_9	2	1	0	0	1	1	0	0	0

Instead, they would "granulate" the attribute search space and focus on finding the right sets of parameters (that correspond to specific attributes) within each attribute group.

The above considerations lead toward several observations. First, the spaces of features/attributes that require to be granulated can be more complex than a set of columns in a tabular data. Let us explain it using a terminology of decision tables that is common for the rough set methods [29]. Therein, data sets are represented as triples $\mathbb{A} = (U, A \cup \{d\})$ with U denoting a universe of objects/rows, A denoting a set of attributes/columns and d referring to a distinguished decision attribute – a target variable. In some real-life scenarios, the set A may require granulation because of its high cardinality. An example of such situation can be found e.g. in [26], where an interactive GUI-based approach for grouping genes-attributes was introduced. However, in other scenarios the set A may not exist in a materialized form. We can rather think about a set A^* gathering all derivable attributes/features, e.g., wavelet coefficients in the case of EEG signal analysis [39] or JSON-driven aggregates defined for a semi-structured data set [40]. Thus, one could think about A^* as a space of all outcomes of the feature engineering/extraction techniques applied in a given application area. We shall treat A^* (sometimes taking a simple form of A) as our granulation domain.

The second observation is about the meaning of granules built over A^* (or A) from the perspective of data understanding and decision model construction, including feature selection studied in Section IV. With respect to data understanding, it is implicitly assumed that features dropping into the same granules should be assessed by domain experts as having some kind of common background, by means of physical, functional or information-specific comparability. In particular, the information level of comparability may correspond to the way, in which particular features contribute to decision models aimed at classifying or distinguishing between different states of target variables. This aspect, as previously mentioned, seems to be close to the ideas of adapting various data clustering methods for the purpose of grouping together similarly acting or replaceable/interchangeable attributes [27]. However, we also need to remember that all of the above flavors of similarity need to be coupled with some tangible criteria for assessing the quality of pre-defined or produced granules [41].

Let us now present two specific examples of the granu-

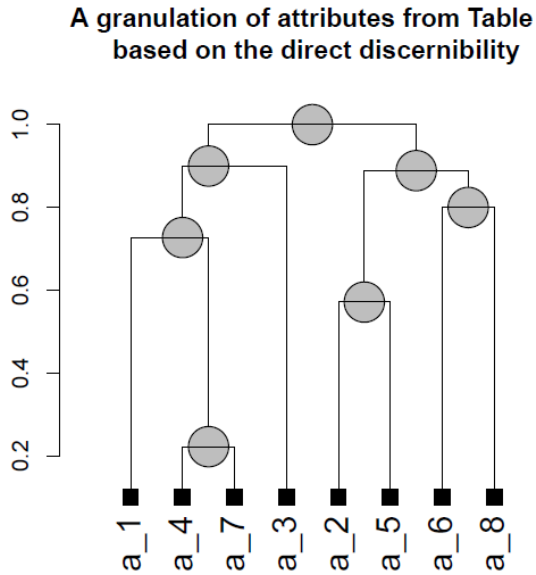

Fig. 1. An attribute clustering tree for the data in Table I, obtained using agglomerative nesting combined with *direct discernibility*. A cut at any given tree height produces attribute granulation, e.g., 0.85 results in four granules $G_1 = \{a_1, a_4, a_7\}$, $G_2 = \{a_3\}$, $G_3 = \{a_2, a_5\}$ and $G_4 = \{a_6, a_8\}$.

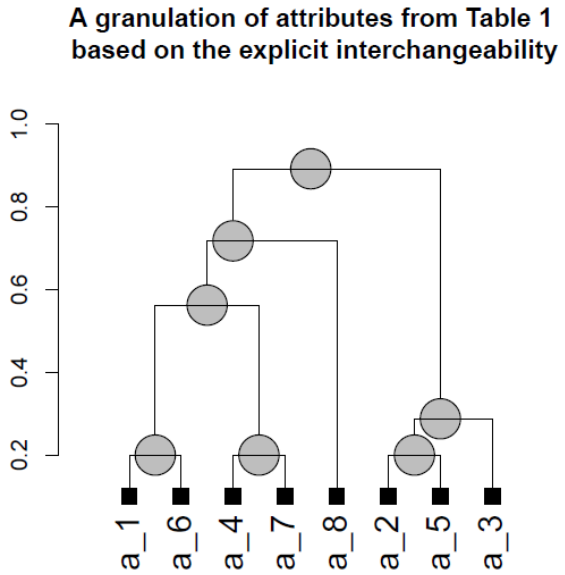

Fig. 2. An attribute clustering tree for the decision Table I, obtained by combining the agglomerative nesting algorithm with the *explicit interchangeability* function. A cut at height 0.85 would result in only two granules $G'_1 = \{a_1, a_4, a_6, a_7, a_8\}$ and $G'_2 = \{a_2, a_3, a_5\}$.

lation based on the attribute interchangeability. Herein we use illustrations related to the original sets of attributes A, although similar analysis can be repeated for A^*. In the first example, two attributes are considered interchangeable if they let us distinguish or *discern* between similar sets of pairs of objects from a data set. This method strongly refers to the principles of the already-mentioned theory of rough sets, where the notion of discernibility plays a fundamental role in deriving dependencies from the data [42]. The following attribute similarity function, further called *direct discernibility*, expresses a ratio between a number of pairs of objects from different decision classes that are discerned by *exactly one* attribute from the considered pair, to a number of such objects discerned by *at least one* of the compared attributes.

$$direct(a,b) = \frac{|\{(u,u'):d(u)\neq d(u') \wedge a(u)\neq a(u') \wedge b(u)\neq b(u')\}|}{|\{(u,u'):d(u)\neq d(u') \wedge (a(u)\neq a(u') \vee b(u)\neq b(u'))\}|}$$

Let us compare a_2 and a_5 in Table I. In the case of the most of the pairs of objects from different decision classes, a_5 discerns them only if a_2 does. This may indicate that there might be relatively many pairs of attribute subsets that on one hand, preserve information about the discernibility and on the other hand, have a form of $B \cup \{a_2\}$ and $B \cup \{a_5\}$, $B \subseteq A \setminus \{a_2, a_5\}$. Subsets $\{a_2, a_3\}$ and $\{a_3, a_5\}$ are a good illustration of this kind of interchangeability. The attributes that are likely to be interchangeable can be identified by studying a dendrogram generated by a hierarchical clustering algorithm. An example of such a tree generated for the data from Table I is shown in Figure 1. As expected, the attributes a_2 and a_5 are merged into a single granule as the second pair. In [6], it was demonstrated in experiments with the microarray data that the attribute granulation obtained using this method can facilitate the search for concise and informative subsets of attributes.

A slightly different approach is centered around the notion of *explicit interchangeability* of features in attribute subsets that are small in size but sufficient to model the target decision classes/labels. In the above-mentioned theory of rough sets, such attribute subsets are usually referred as *decision reducts*. Intuitively, if two attributes rarely belong to the same subset but they both often appear together with similar groups of other attributes, they may be considered interchangeable. In the opposite situation, when two attributes often belong to the same subset or appear in a company of completely different features, it seems reasonable to assume that they convey different information and thus are not similar. More formally, this type of attribute interchangeability can be measured using a co-occurrence frequency matrix F, whose entry in $i\text{-}th$ row and $j\text{-}th$ column equals $f_{i,j}$:

$$f_{i,j} = \frac{|\{k:a_i \in AS_k \wedge a_j \in AS_k\}|}{|\{k:a_i \in AS_k\}|}$$

where a_i, a_j are attributes, $i \neq j$ and AS_k is the $k\text{-}th$ pre-computed attribute subset. All values at diagonal of F are set to 0. The final values of attribute interchangeability can be computed as a difference between the similarity of corresponding feature sets and the frequency, with which the given features co-occur:

$$I(a_i, a_j) = cosine(f_{i,\cdot}, f_{j,\cdot}) - f_{i,j} \quad (1)$$

In this formula, $f_{i,\cdot}$ and $f_{j,\cdot}$ are vectors of values from $i\text{-}th$ and $j\text{-}th$ rows of F, respectively. An example of attribute granulation based on the explicit interchangeability measure

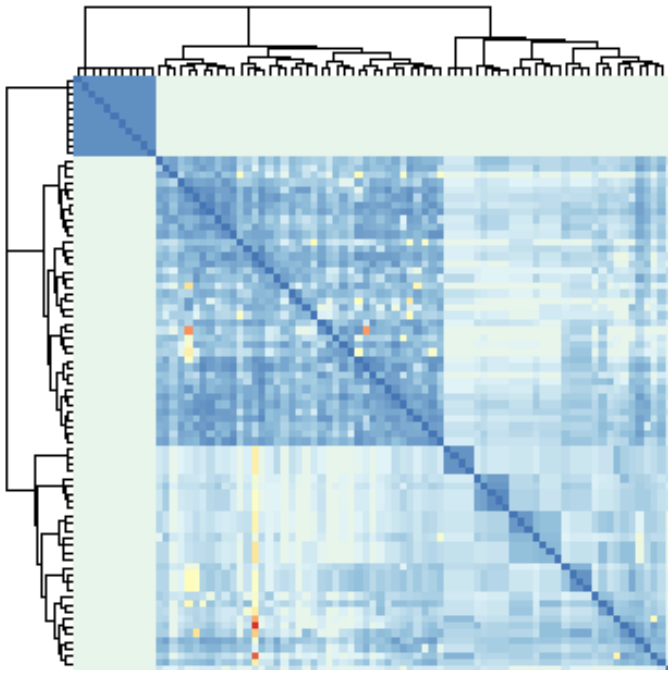

Fig. 3. A fragment of a heat map expressing interchangeability of risk factors (represented as attributes) taken from the AAIA'14 Data Mining Competition. Granules of attributes are arranged along the diagonal of the matrix.

is shown in Figure 2. In that example, attribute subsets were defined as all decision reducts of Table I.

The above approach was successfully applied in [43], where the task was to analyze results of a data mining challenge aiming at identifying key risk factors for firefighters during fire&rescue actions. Participants of that challenge submitted solutions in a form of relevant attribute subsets (each attribute in the data was interpreted as a factor influencing the risk of injury). Typically, submitted subsets were small but the analysis of thousands of solutions using the formula (1) allowed us to construct a granulation of the risk factors into groups that were meaningful to domain experts. Figure 3 depicts a heat map of a distance matrix that was used for this task in a combination with an agglomerative clustering algorithm (the distance was computed as $1 - I(a_i, a_j)$). Analogous heat maps could be computed using ensembles of possibly diverse *approximate* decision reducts that preserve information about the decision classes only to some extent and, thus, they can utilize different groups of attributes to concentrate on different aspects of approximate data dependencies [44].

IV. Feature Selection with Attribute Granules

The process of feature selection aims at exploring the given attribute space A (or A^*) and extracting a relatively small subset $R \subseteq A$ of attributes that, on the one hand, are the most relevant and, on the other hand, are sufficient to solve the investigated problem. Such selection/extraction process is often conducted by applying statistical tests in order to determine, which attributes contribute to the constructed decision model [33]. In some approaches, the attributes are also analyzed with respect to their interdependencies observed within decision models [45]. Actually, one may regard the idea of looking at co-occurrences of attributes in approximate decision reducts – as mentioned in the end of Section III – as useful for the purposes of feature selection as well. This shows that the goals of feature clustering/granulation and selection can be quite complementary to each other.

In this section, we focus on a slightly different aspect of combining the above ideas. Namely, we examine to what extent a pre-computed/pre-defined feature granulation can guide the process of choosing the most appropriate collections of attributes. Although the standard feature selection algorithms are not configured for attributes that are structured or bound by relationships, it seems to be relatively easy to take such an additional aspect into account. The knowledge about attribute granulation can have an important impact on the final subset composition and, hence, we argue that it should influence the order, in which we investigate attributes.

For the purpose of further discussion, let us concentrate on a common approach to conducting a feature selection. Certainly, we do not claim that all possible methods follow the scheme below. Nevertheless, in our opinion it is sufficiently general to explain the benefits of working with attribute granules. For a given input set of attributes A (or A^*), let us consider a criterion function $\mathbb{C} : 2^A \to \{0, 1\}$ whose purpose is simply to indicate, which subsets of A are already rich enough to serve as the outcomes of the selection process. In practice, $\mathbb{C}$ may correspond to a collection of criteria reflecting different requirements. Additionally, consider an arbitrary heuristic quality function $Q : 2^A \to \mathbb{R}$ that can be utilized iteratively to add the most "promising" elements to the constructed feature subset. Let us note that Q can combine various aspects of relationships between the selected attributes and a target variable [34], [46]. Let us also mention that the last item of the following procedure has strong roots in the theory of rough sets, where there is a particular focus on the simplification of decision models learnt from the data [29], [47].

1) While the criterion $\mathbb{C}(R)$ is not met by the selected feature subset R continue the following:
   a) Select candidate sets of (subsets of) features $B_1, .., B_k$ to be added to R
   b) Evaluate $B_1, .., B_k$ with the desired attribute subset quality measure Q
   c) If the best B_x contributes to R, then $R \leftarrow R \cup B_x$
   d) Verify if the criterion $\mathbb{C}(R)$ is met
2) Eliminate superfluous attributes from R

Algorithm 1 reflects our generic idea of embedding the additional knowledge about attribute granulation into the above-described feature selection process. In each iteration of the main loop, in order to limit the attribute space A, the subset of granules $\{G_1, .., G_m\} \subseteq \mathbb{G}$ is selected with respect to the granulation preferences expressed by, e.g., a permutation $\sigma_{\mathbb{G}} : \{G_{\sigma(1)}, G_{\sigma(2)}, ..\}$ (which means that the granule $G_{\sigma(1)}$ is most preferred to draw attributes from). By limiting the search space using the additional knowledge about attribute granula-

Algorithm 1 General framework for granular feature selection

Input: $\mathbb{G}$ – set of granules, A – attribute space,
$\mathbb{C}$– criterion function, $\sigma_{\mathbb{G}}$ – granule preferences
Output: R – selected attribute subset
Initialization:
1: $R \leftarrow \emptyset$
2: **while** R does not satisfy $\mathbb{C}(R)$ **do**
3: $B \leftarrow \emptyset$
4: Select granules $\{G_1, .., G_m\} \subseteq \mathbb{G}$ with respect to $\sigma_{\mathbb{G}}$
5: Limit attribute space $A_G \leftarrow A \cap \bigcup_{1 \leq i \leq m} G_i$
6: Generate candidates $B_1, .., B_k \subseteq A_G$
7: Evaluate candidates $\{B_1, .., B_k\}$
8: $B \leftarrow selectBestCandidate(\{B_1, .., B_k\}, ...)$
9: **if** B contributes to R **then**
10: $R \leftarrow R \cup B$
11: **end if**
12: **end while**
13: $R \leftarrow eliminateSuperfluousAttributes(R)$
14: **return** R

tion, we may quickly generate a set of candidates $\{B_1, .., B_k\}$. After the evaluation of candidates with the correlation, gini index or other implementation of the function Q, the feature subset R may be extended if only the selected B contributes to R. The loop continues until a "good enough" R is collected or all combinations/candidates are explored. Finally, we conduct a backward elimination of superfluous attributes.

The presented framework does not enforce any particular interpretation of the information granules and, thus, different implementations may vary in a way of their utilization. In some cases, it may be preferred to select features that belong to only one, specific granule. For example, the analysis of coal mine sensor readings [11] may be oriented on the one, particular mine shaft. In that case, the analysts could generate granules on the basis of a sensor location and introduce a constraint that the finally selected attributes should/must belong to the particular granule(s). In other applications, it may be convenient to generate an attribute subset that contains attributes from multiple granules in order to provide higher robustness [27]. Regardless of the way that we use the attribute granulation, the general framework is still the same. To give a better understanding of the abstracted phases of Algorithm 1, Figure 4 presents its exemplary iteration for the data presented in Table I and attribute granulation outlined in Figure 1.

Attribute granulation may also influence a feature selection process with respect to the expected robustness and resilience of decision models. In real-life applications, we may observe various anomalies in explored data sets, which cause a model over-fitting. Some researchers emphasize the role of appropriate granulation of attributes during feature engineering in achieving higher stability of the created models. With that respect, we may refer to several techniques using, e.g., histograms [48] or the already-discussed clustering [25]. During the decision model construction, there are also some non-functional factors that could impact the continuity of analysis like, e.g., temporal or permanent unavailability of some sources during on-line data collection [49]. From this perspective, it is advisable to use diverse feature subsets and ensemble methods, whereby each of separate decision models is based on a few attributes but, overall, many attributes are involved [6]. Thus, it is important to combine the feature selection approaches relying on the attribute granulation with some feature subset diversification methods.

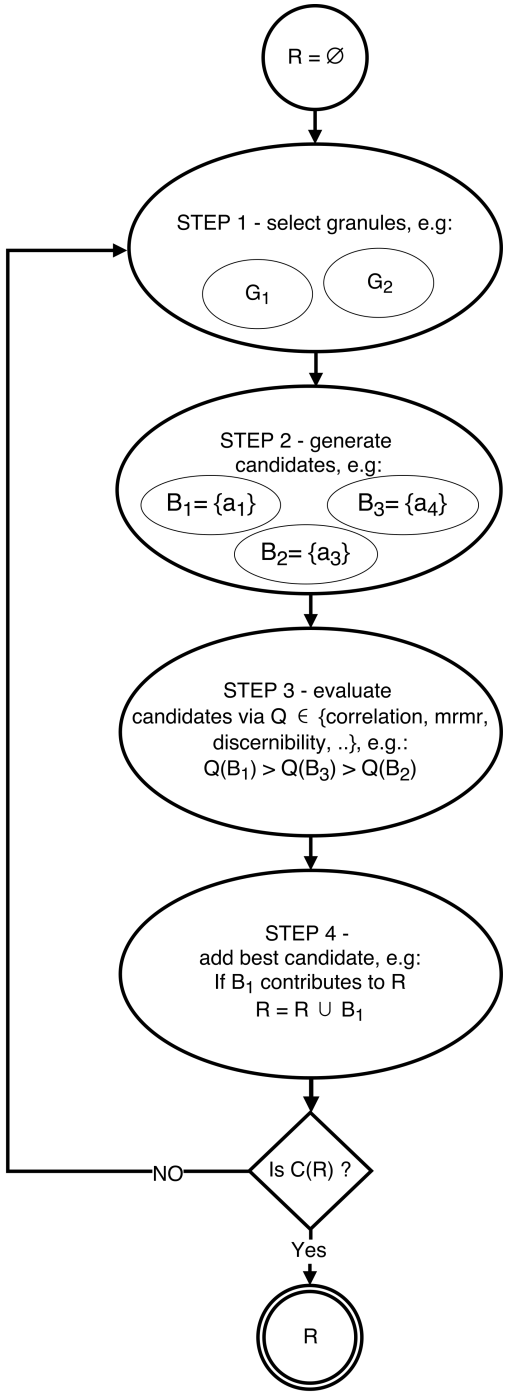

Fig. 4. A single loop of a feature selection algorithm for the data referred in Figure 1, taking into account the knowledge about attribute granulation.

Algorithm 2 Full-granule-oriented version of Algorithm 1

Input: $\mathbb{G}$ – set of granules, A – attribute space,
Q– quality function, $\mathbb{C}$– criterion function
Output: R – selected attribute subset
Initialization :
1: $R \leftarrow \emptyset$
2: **while** R does not satisfy $\mathbb{C}(R) = 1$ **do**
3: Select granules $\{G_1, .., G_m\} \subseteq \mathbb{G}$
4: Evaluate granules $\{G_1, .., G_m\}$ and pick the best G_x
5: **if** G_x contributes to R **then**
6: $R \leftarrow R \cup G_x$
7: **end if**
8: **end while**
9: $R \leftarrow eliminateSuperfluousGranules(R)$
10: **return** R

In this context, the analysts could utilize feature granulation in order to achieve more robust and resilient results due to, e.g., exploitation of attributes extracted from diverse sources. In particular, the method outlined by Algorithm 1 could be used to compose an attribute subset R as a collection of features from diverse granules. In this case, the attribute reduction algorithm should aim at achieving feature subsets of minimal cardinality $|R|$ and also ensure the diversity of granules by, e.g., maximization of $|\{G \in \mathbb{G} : R \cap G \neq \emptyset\}|$. Accordingly, a specialized configuration of the main loop in the presented framework can take into account, both, the so-far-selected features and the granules that are used less often, i.e., granules G_i that minimize the quantity of $|G_i \cap R|$.

The feature selection methods should be also able to operate on the whole granules or their subsets instead of individual attributes. To some extent, it corresponds to the idea of so-called *decision systems with constraints* – the enriched data representation proposed in [7]. The goal of this approach is not only to record the presence of granules (called constraints) but also to make it possible to apply various computational methods that make use of them. Let us consider Algorithm 2, where the overall scheme is aligned with Algorithm 1, though one can notice some simplifications like selecting particular granules $G_1, .., G_m$ as the candidate subsets B_i. Similarly, the backward elimination concerns removal of the whole granules instead of individual attributes. In such approaches, as it was observed also by other researchers, the properties of selected attribute subsets can depend a lot on coarsening or refining granules [50]. Therefore, as it was discussed in Section III, there is a need for a framework allowing the domain experts and algorithm designers to assess the results of feature selection/granulation processes from different perspectives.

As we could see above, Algorithm 1 can be treated as a general umbrella for various approaches aiming at utilization of the attribute space granulation for the purpose of enhancing the feature selection process. Surely, there are still several details to be discussed. First, it is useful to look at different strategies of validating whether a given attribute sufficiently *contributes* to the result R [51]. Second, it is interesting

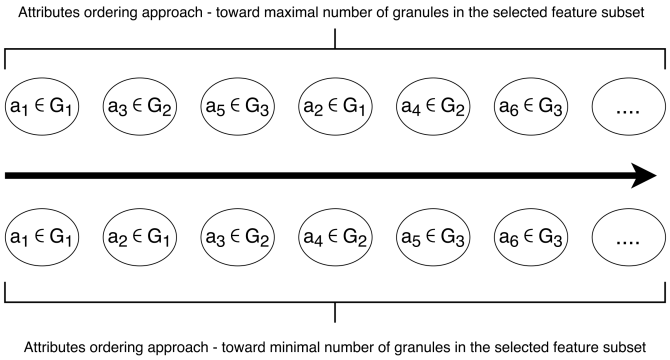

Fig. 5. A diagram with two significantly different attribute ordering strategies that take into account granulation of attributes.

to compare the proposed framework with methods based on attribute orderings. The main idea behind this class of methods is to iterate along diversified permutations σ_A over A. Such permutations can be induced partially with respect to some heuristic function Q or they can be generated fully randomly [30]. In the latter case, the procedure is repeated a number of times and the best of the obtained attribute subsets (or a bigger ensemble of subsets) is eventually selected.

Figure 5 shows how we can use the knowledge about granules to influence permutations, e.g., by arranging the elements of the same granule within consecutive subsequences or mixing them together as much as possible (by following a preference permutation $\sigma_{\mathbb{G}} : \{G_{\sigma(1)}, G_{\sigma(2)}, ..\}$). It is important to note that such two semi-randomized strategies are in a correspondence to the ideas of operating with regular granules (Algorithm 2) and maximally diverse attribute subsets, respectively. This shows that the attribute granulation is easily applicable to the ordering-based feature selection algorithms, without a necessity to modify their code. On the one hand, the described scenarios of "granular ordering" are conceptually aligned with Algorithm 1. On the other hand, the phase of selecting granules/candidates can be performed implicitly at a level of generation of attribute permutations.

V. MapReduce over Attribute Granulations

There are at least two more aspects of utilizing the attribute granulations to improve the feature selection processes, especially in the context of Big Data. The high velocity and volume of still-incoming records are often a curse of storage systems and machine learning algorithms. Furthermore, raw records are often insufficient for the purpose of predictive analysis and the process of feature engineering (i.e., switching to A^*) is commonly employed to construct more relevant attributes [52]. The massively parallel feature engineering methods may be efficiently performed via the MapReduce programming model what, in turn, may multiply the initial number of explored attributes. Still, the question remains how to choose which attributes should be evaluated. As suggested in [8], the actual feature selection process can be performed at a level of general labels of some attribute granules, whereby specific elements of those granules are not materialized prior to the algorithm's

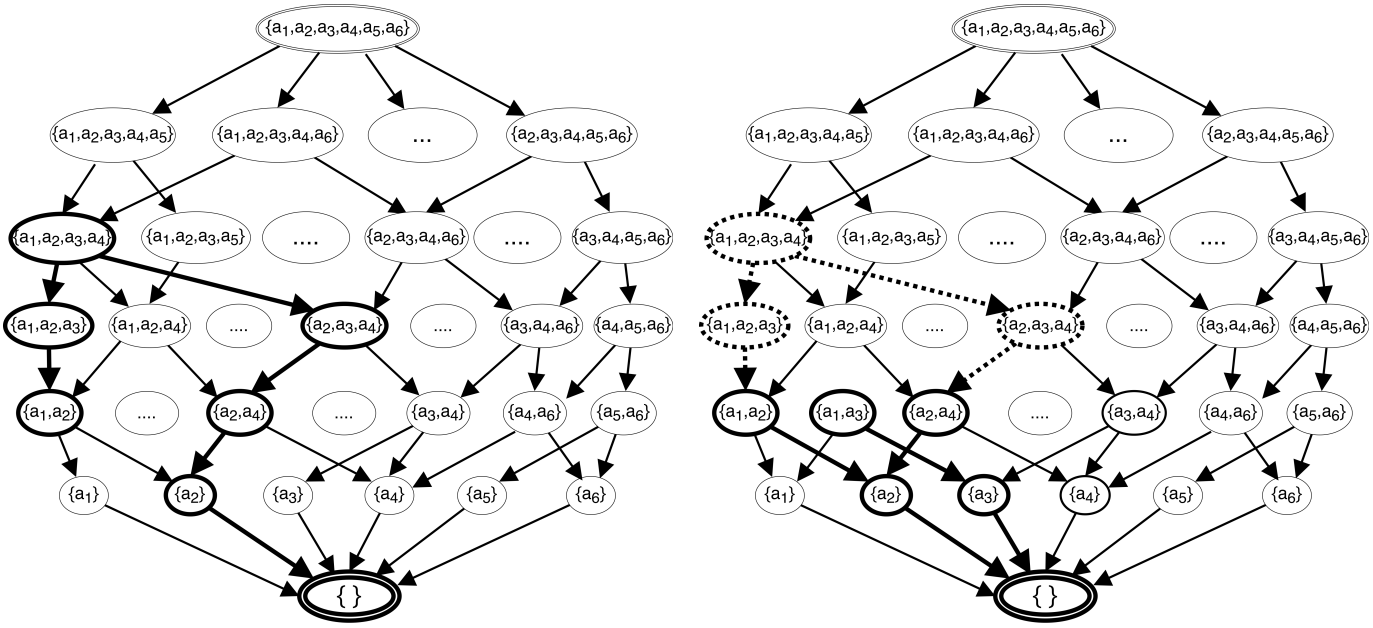

Fig. 6. The attribute lattices reflecting possible feature subsets generated for the exemplary data set. Bold ovals and arrows visualize paths explored during forward propagation and backward elimination phases. The left lattice corresponds to a single-threaded algorithm. The right lattice shows a parallel version. Dotted lines emphasize steps omitted in a parallel algorithm comparing to a single-threaded version.

start. This style of hierarchical feature space exploration fits perfectly Algorithm 1 and its specific configurations.

The execution of a feature selection method may be expressed as searching through the attribute lattice, as presented in Figure 6. The left lattice visualizes a stand-alone selection process, with only a few features involved into the process. This clearly refers to the above idea of avoiding extracting/-materializing too many attributes in advance. Let us recall that our algorithms actually work over the set A^* of features that are extractable from the stored data. Thus, the process of feature engineering should result in computing values of only those features, which are recognized as important based on the additional knowledge about their granulation. In other words, although the corresponding lattice is practically "infinite", it does not mean that a given algorithm's run requires materialization of all potentially involved features.

The second aspect corresponds to parallelization of calculations, leading towards shortening the time of computations. This naturally responds to the necessity to handle the enormous velocity and volume of Big Data. Nevertheless, Big Data is also connected with various issues with respect to the quality and (in)consistency, that may hamper processing and eventually affect the accuracy of the analysis. When we take a look at the attribute lattice presented on the right side of Figure 6, we see that multiple (bold) paths are explored and multiple attribute subsets are examined in parallel. As a result, the parallelization of feature selection allows us to evaluate more potentially important candidates, which results in a higher quality of the final outcome. Moreover, it goes well together with the above-mentioned idea of multiple executions of (differently initiated) feature selection loops, which are now conducted in parallel rather than in a serialized fashion.

Models and frameworks for parallel computing focus on various aspects of data processing. Some of them respond to high velocity of the data, which makes them closer to incremental stream processing [38]. Others concentrate on batch processing models and adapt well-known mechanisms, such as the apriori-based breadth first exploration of a feature space [53]. Herein, the MapReduce paradigm seems to be a good choice to consider [31], [54]. We may distinguish two popular approaches in this field. One of them implements the solution as a single job, whereas the other – iterative MapReduce – encompasses ℓ consecutive job runs that may be controlled automatically or manually [55], [56]. One can think about parallelization of the discussed granular feature selection methods using both of these approaches.

Let us briefly outline one of possible implementations of a massively parallel granular feature selection process as an iterative MapReduce program. Consider ℓ consecutive iterations, where each of them is based on Algorithm 3. We propose to work on the transmuted data, i.e., the mappers are executed on attributes a assigned to a granule G_a and having a vector V_a of values for objects/records in the analyzed data set. The outcome of a single iteration is a sorted set of candidate attribute subsets, whereas only n best intermediate outputs $\mathbb{R} = \{R_1, .., R_n\}$ are passed to the subsequent phase. The map functions are provided with the collection $\mathbb{R}$ and the vector V_d containing values of the decision attribute d. To each subset R_i there has been assigned granulation preferences $\sigma^i_{\mathbb{G}}$, whereby the diversification of granule-level permutations may

Algorithm 3 Granular feature selection with iterative MapReduce. In each of ℓ-phases the following program is executed.

Map(Key: $a \in A$**, Value:** G_a, V_a**)**
1: Given $\mathbb{R} = \{R_1, .., R_n\}$
2: **for all** $R_i \in \mathbb{R}$ **do**
3: **if** a is relevant to R_i **then**
4: $R_i \leftarrow R_i \cup \{a\}$
5: **emit**(sortAttributes$(R_i), \sigma^i_{\mathbb{G}}$, score)
6: **end if**
7: **end for**
Reduce(Key:$R_i, \sigma^i_{\mathbb{G}}$**, Value:** $\{score, score, ..\}$**)**
8: **emit(** $R_i, \sigma^i_{\mathbb{G}}$, score)

play a similar role as for the previously discussed attribute-level permutations. During the evaluation of a, we verify its relevance to every considered R_i, with respect to a quality function Q, preferences $\sigma^i_{\mathbb{G}}$, or any other factor of interest. If the performed assessment reveals that a is relevant for R_i (where relevance may be expressed as a mixture of preference, contribution, etc.), then the set $R_i \cup \{a\}$ is emitted. The role of reducers is then to aggregate subsets R_i and sort them according to their score. The whole process ends when the expected number of feature subsets satisfies $\mathbb{C}$.

The main objective of the above illustrative example of a MapReduce program is to evaluate a possibly large number of attribute subsets, in order to reach a higher quality, compactness and/or diversification of the produced outcomes. Obviously, parallel programming models allow to implement the granular feature selection framework in many other ways [13], [31]. Nevertheless, the major conclusion of this part of our paper is that the idea of operating on attribute granules – regardless of their origin discussed in Section III – is truly worth combining with the principles of parallelization of feature selection methods with respect to complex spaces of derivable features and their subsets.

VI. Final Remarks

We presented a particular take on the challenge of devising a more effective and efficient feature selection methodology. The main idea behind our approach is to make an intelligent use of the information granulation paradigm in the context of aggregating, selecting and engineering attributes (features/variables/dimensions) that describe the data. The resulting solution is meant to convey the granular knowledge that is in the data. At the same time, it is designed to deal with enormous amount of information that needs to be processed when facing the kinds of tasks typical for Big Data.

The gist is to operate on attribute granules that are formed through the use of various knowledge discovery algorithms, such as, e.g., clustering or interchangeability analysis through heat maps. In many instances, as exemplified by use cases discussed in Section IV, granules built over the attribute space may represent semantic relationships that are important for domain experts. The proposed feature selection framework, coupled with the granular structure of attribute space, facilitates discovering meaningful knowledge from the underlying data. This knowledge may be further leveraged in order to obtain a more comprehensible and user-friendly representation of the final decision model.

The proposed algorithms for both, granule construction and feature selection, can make use of various forms of problem decomposition and parallelization, as outlined in Section V. They are capable of tackling large spaces of derivable features and their subsets. Hence, they respond to demand for efficiency which is central for all approaches to vast amounts of information, and may have a significant impact on the feasibility of granular feature selection.

Some of the next steps towards practical use of granule-based methodology for dealing with data that is characterized by high dimensionality, large size and high complexity, may be directed towards incorporation of domain knowledge into the process of the granule construction and feature extraction. In particular, the long-term goal would be to devise methods and tools that automate this process and at the same time maintain acceptable level of transparency and human readability. A granular system capable of flexible, comprehensible and extensible interaction with data scientists who analyzes massive data sets could be an invaluable tool.

In a shorter perspective, next steps should involve integration with the existing technologies. While in the paper we have shown how the MapReduce principles can be employed, there is a plethora of other techniques that were developed over the years with Big Data in mind. For example, it could be helpful to integrate the proposed methods with the existing tools for management of massive relational data sets (such as Apache Hive or some approximate database engines). This way, we could embed the "zoom in/out" operations on attributes into a convenient RDBMS environment.

References

[1] J. Li and H. Liu, "Challenges of Feature Selection for Big Data Analytics," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 9–15, 2017.

[2] Y. Yao and N. Zhong, "Granular Computing," in *Wiley Encyclopedia of Computer Science and Engineering*, B. W. Wah, Ed. Wiley, 2008.

[3] Q. Hu, J. Liu, and D. Yu, "Mixed Feature Selection Based on Granulation and Approximation," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 294–304, 2008.

[4] F. Li, D. Miao, and W. Pedrycz, "Granular Multi-Label Feature Selection Based on Mutual Information," *Pattern Recognition*, vol. 67, pp. 410–423, 2017.

[5] P. Govindan, R. Chen, K. Scheinberg, and S. Srinivasan, "A Scalable Solution for Group Feature Selection," in *Proc. of IEEE BigData 2015*, pp. 2846–2848.

[6] A. Janusz and D. Ślęzak, "Rough Set Methods for Attribute Clustering and Selection," *Applied Artificial Intelligence*, vol. 28, no. 3, pp. 220–242, 2014.

[7] S. H. Nguyen and M. S. Szczuka, "Feature Selection in Decision Systems with Constraints," in *Proc. of IJCRS 2016*, pp. 537–547.

[8] S. Widz and D. Ślęzak, "Granular Attribute Selection: A Case Study of Rough Set Approach to MRI Segmentation," in *Proc. of PReMI 2013*, pp. 47–52.

[9] M. García-Torres, F. Gómez-Vela, B. Melián-Batista, and J. M. Moreno-Vega, "High-Dimensional Feature Selection via Feature Grouping," *Information Sciences*, vol. 326, no. C, pp. 102–118, 2016.

[10] J. Fan and J. Lv, "A Selective Overview of Variable Selection in High Dimensional Feature Space," *Statistica Sinica*, vol. 20, no. 1, pp. 101–148, 2010.

[11] A. Janusz, M. Grzegorowski, M. Michalak, L. Wróbel, M. Sikora, and D. Ślęzak, "Predicting Seismic Events in Coal Mines Based on Underground Sensor Measurements," *Engineering Applications of Artificial Intelligence*, vol. 64, pp. 83–94, 2017.

[12] M. H. Rehman, V. Chang, A. Batool, and T. Y. Wah, "Big Data Reduction Framework for Value Creation in Sustainable Enterprises," *International Journal of Information Management*, vol. 36, no. 6, pp. 917–928, 2016.

[13] J. Qian, P. Lv, X. Yue, C. Liu, and Z. Jing, "Hierarchical Attribute Reduction Algorithms for Big Data Using MapReduce," *Knowledge-Based Systems*, vol. 73, no. C, pp. 18–31, 2015.

[14] Z. Zhao, R. Zhang, J. Cox, D. Duling, and W. Sarle, "Massively Parallel Feature Selection: An Approach Based on Variance Preservation," *Machine Learning*, vol. 92, no. 1, pp. 195–220, 2013.

[15] M. S. Szczuka and D. Ślęzak, "How Deep Data Becomes Big Data," in *Proc. of IFSA/NAFIPS 2013*, pp. 579–584.

[16] I. Triguero, D. Peralta, J. Bacardit, S. García, and F. Herrera, "MRPR: A MapReduce Solution for Prototype Reduction in Big Data Classification," *Neurocomputing*, vol. 150, pp. 331–345, 2015.

[17] C. Eiras-Franco, V. Bolón-Canedo, S. Ramos, J. González-Domínguez, A. Alonso-Betanzos, and J. Touriño, "Multithreaded and Spark Parallelization of Feature Selection Filters," *Journal of Computational Science*, vol. 17, pp. 609–619, 2016.

[18] A. Chądzyńska-Krasowska, P. Betliński, and D. Ślęzak, "Scalable Machine Learning with Granulated Data Summaries: A Case of Feature Selection," in *Proc. of ISMIS 2017*, pp. 519–529.

[19] D. Ślęzak, M. Grzegorowski, A. Janusz, and S. Stawicki, "Toward Interactive Attribute Selection with Infolattices – A Position Paper," in *Proc. of IJCRS 2017, Part II*, pp. 526–539.

[20] A. Agrawal, J. Choi, K. Gopalakrishnan, S. Gupta, R. Nair, J. Oh, D. A. Prener, S. Shukla, V. Srinivasan, and Z. Sura, "Approximate Computing: Challenges and Opportunities," in *Proc. of ICRC 2016*, pp. 1–8.

[21] H. D. Bondell and B. J. Reich, "Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR," *Biometrics*, vol. 64, no. 1, pp. 115–123, 2008.

[22] B. Gu, G. Liu, and H. Huang, "Groups-Keeping Solution Path Algorithm for Sparse Regression with Automatic Feature Grouping," in *Proc. of KDD 2017*, pp. 185–193.

[23] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable K-Means++," *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 622–633, 2012.

[24] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google News Personalization: Scalable Online Collaborative Filtering," in *Proc. of WWW 2007*, pp. 271–280.

[25] M. A. Hall, "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning," in *Proc. of ICML 2000*, pp. 359–366.

[26] A. Grużdź, A. Ihnatowicz, and D. Ślęzak, "Interactive Gene Clustering – A Case Study of Breast Cancer Microarray Data," *Information Systems Frontiers*, vol. 8, no. 1, pp. 21–27, 2006.

[27] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys, "Robust Biomarker Identification for Cancer Diagnosis with Ensemble Feature Selection Methods," *Bioinformatics*, vol. 26, no. 3, pp. 392–398, 2010.

[28] T.-P. Hong, Y.-L. Liou, S.-L. Wang, and B. Vo, "Feature Selection and Replacement by Clustering Attributes," *Vietnam Journal of Computer Science*, vol. 1, no. 1, pp. 47–55, 2014.

[29] R. W. Świniarski and A. Skowron, "Rough Set Methods in Feature Selection and Recognition," *Pattern Recognition Letters*, vol. 24, no. 6, pp. 833–849, 2003.

[30] S. Stawicki, D. Ślęzak, A. Janusz, and S. Widz, "Decision Bireducts and Decision Reducts – A Comparison," *International Journal of Approximate Reasoning*, vol. 84, pp. 75–109, 2017.

[31] P. Li, J. Wu, and L. Shang, "Fast Approximate Attribute Reduction with MapReduce," in *Proc. of RSKT 2013*, pp. 271–278.

[32] P. Hońko, "Attribute Reduction: A Horizontal Data Decomposition Approach," *Soft Computing*, vol. 20, no. 3, pp. 951–966, 2016.

[33] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[34] H. Peng, F. Long, and C. Ding, "Feature Selection Based on Mutual Information Criteria of Max-dependency, Max-relevance, and Min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[35] W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems*. CRC Press, 2013.

[36] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering Cancer Gene Expression Data: A Comparative Study," *BMC Bioinformatics*, vol. 9, 2008.

[37] A. Appice, P. Guccione, D. Malerba, and A. Ciampi, "Dealing with Temporal and Spatial Correlations to Classify Outliers in Geophysical Data Streams," *Information Sciences*, vol. 285, pp. 162–180, 2014.

[38] M. Grzegorowski and S. Stawicki, "Window-based Feature Extraction Framework for Multi-Sensor Data: A Posture Recognition Case Study," in *Proc. of FedCSIS 2015*, pp. 397–405.

[39] M. S. Szczuka and P. Wojdyłło, "Neuro-Wavelet Classifiers for EEG Signals Based on Rough Set Methods," *Neurocomputing*, vol. 36, no. 1-4, pp. 103–122, 2001.

[40] A. Janusz, T. Tajmajer, and M. Świechowski, "Helping AI to Play Hearthstone: AAIA'17 Data Mining Challenge," in *Proc. of FedCSIS 2017*, pp. 121–125.

[41] A. Janusz and M. S. Szczuka, "Assessment of Data Granulations in Context of Feature Extraction Problem," in *Proc. of IEEE GrC 2014*, pp. 116–120.

[42] Z. Pawlak and A. Skowron, "Rough Sets and Boolean Reasoning," *Information Sciences*, vol. 177, no. 1, pp. 41–73, 2007.

[43] A. Janusz, A. Krasuski, S. Stawicki, M. Rosiak, D. Ślęzak, and H. S. Nguyen, "Key Risk Factors for Polish State Fire Service: A Data Mining Competition at Knowledge Pit," in *Proc. of FedCSIS 2014*, pp. 345–354.

[44] J. Wróblewski, "Ensembles of Classifiers Based on Approximate Reducts," *Fundamenta Informaticae*, vol. 47, no. 3-4, pp. 351–360, 2001.

[45] M. Dramiński, A. Rada-Iglesias, S. Enroth, C. Wadelius, J. Koronacki, and H. J. Komorowski, "Monte Carlo Feature Selection for Supervised Classification," *Bioinformatics*, vol. 24, no. 1, pp. 110–117, 2008.

[46] M. Dash and H. Liu, "Consistency-based Search in Feature Selection," *Artificial Intelligence*, vol. 151, no. 1-2, pp. 155–176, 2003.

[47] Y. Yao, Y. Zhao, and J. Wang, "On Reduct Construction Algorithms," *LNCS Transactions on Computational Science*, vol. 2, pp. 100–117, 2008.

[48] E. Zdravevski, P. Lameski, R. Mingov, A. Kulakov, and D. Gjorgjevikj, "Robust Histogram-based Feature Engineering of Time Series Data," in *Proc. of FedCSIS 2015*, pp. 381–388.

[49] M. Grzegorowski, "Governance of the Redundancy in the Feature Selection Based on Rough Sets' Reducts," in *Proc. of IJCRS 2016*, pp. 548–557.

[50] Y. Jing, T. Li, C. Luo, S.-J. Horng, G. Wang, and Z. Yu, "An Incremental Approach for Attribute Reduction Based on Knowledge Granularity," *Knowledge-Based Systems*, vol. 104, no. C, pp. 24–38, 2016.

[51] A. Janusz and D. Ślęzak, "Computation of Approximate Reducts with Dynamically Adjusted Approximation Threshold," in *Proc. of ISMIS 2015*, pp. 19–28.

[52] F. Ahmed, M. Samorani, C. Bellinger, and O. R. Zaïane, "Advantage of Integration in Big Data: Feature Generation in Multi-Relational Databases for Imbalanced Learning," in *Proc. of IEEE BigData 2016*, pp. 532–539.

[53] J. Xie, J. Wu, and Q. Qian, "Feature Selection Algorithm Based on Association Rules Mining Method," in *Proc. of ICIS 2009*, pp. 357–362.

[54] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, "Parallel Data Processing with MapReduce: A Survey," *SIGMOD Record*, vol. 40, no. 4, pp. 11–20, 2012.

[55] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, "Map-reduce for Machine Learning on Multicore," in *Proc. of NIPS 2006*, pp. 281–288.

[56] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: A Runtime for Iterative MapReduce," in *Proc. of HPDC 2010*, pp. 810–818.