

Dynamic rough sets features reduction

Walid Moudani, Ahmad Shahin, Fadi Chakik, Felix Antonio Claudio
Mora-Camino

► **To cite this version:**

Walid Moudani, Ahmad Shahin, Fadi Chakik, Felix Antonio Claudio Mora-Camino. Dynamic rough sets features reduction. International Journal of Computer Science and Information Security, IJCSIS, 2011, 9 (4), pp 1-10. <hal-01021589>

HAL Id: hal-01021589

<https://hal-enac.archives-ouvertes.fr/hal-01021589>

Submitted on 31 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic Rough Sets Features Reduction

Walid MOUDANI¹, Ahmad SHAHIN², Fadi CHAKIK², and Félix Mora-Camino³

¹Lebanese University, Faculty of Business, Dept. of Business Information System, Lebanon

²LaMA – Liban, Lebanese University, Lebanon

³Air Transportation Department, ENAC, 31055 Toulouse, France

Abstract—Nowadays, and with the current progress in technologies and business sales, databases with large amount of data exist especially in retail companies. The main objective of this study is to reduce the complexity of the classification problems while maintaining the prediction classification quality. We propose to apply the promising technique Rough Set theory which is a new mathematical approach to data analysis based on classification of objects of interest into similarity classes, which are indiscernible with respect to some features. Since some features are of high interest, this leads to the fundamental concept of “Attribute Reduction”. The goal of Rough set is to enumerate good attribute subsets that have high dependence, discriminating index and significance. The naïve way of is to generate all possible subsets of attribute but in high dimension cases, this approach is very inefficient while it will require $2^d - 1$ iterations. Therefore, we apply Dynamic programming technique in order to enumerate dynamically the optimal subsets of the reduced attributes of high interest by reducing the degree of complexity. Implementation has been developed, applied, and tested over a 3 years historical business data in Retail Business. Simulations and visual analysis are shown and discussed in order to validate the accuracy of the proposed tool

Keywords- Data Mining; Business Retail; Rough Sets; Attribute Reduction; Classification; Dynamic Programming.

I. INTRODUCTION

Retail Business (RB) Company looks for increasing its benefit by providing all facilities services to its customers. The estimated benefits amount to several millions of dollars when the Retail Business Company organizes and offers to its customers the most related items. The RB Company stores and generates tremendous amounts of raw and heterogeneous data that provides rich fields for Data Mining (DM) [1, 2]. This data includes transactions Details (customers/providers) describing the content such as items, quantity, date, unit price, reduction, and other events such as the holidays, special activities, etc. Moreover, the profile of customers and their financial transactions contribute in personalizing some special services to each customer. This leads the research community to study deeply this field in order to propose a new solution approach for these companies. Moreover, these companies should analyze their business data in order to predict the appropriate services to be proposed to its customers. This approach is one of the main objectives of the retailer company. In order to build such a non trivial model, many researches were carried out on the feasibility of using the DM techniques, which raised from the need of analyzing high volumes of data

collected by the retailer companies and related to different kinds of transactions between the company and its customers/providers. Our contribution aims to reduce the complexity of the classification process by reducing the number of attributes that should be considered in order to discover the fruitful knowledge required by decision makers of RB.

The 1990s has brought a growing data glut problem to many fields such as science, business and government. Our capabilities for collecting and storing data of all kinds have far outpaced our abilities to analyze, summarize, and extract knowledge from this data [9]. Traditional data analysis methods are no longer efficient to handle voluminous data sets. How to understand and analyze large bodies of data is a difficult and unresolved problem. The way to extract the knowledge in a comprehensible form for the huge amount of data is the primary concern. DM refers to extracting knowledge from databases that can contain large amount of data describing decisions, performance and operations. However, analyzing the database of historical data containing critical information concerning past business performance, helps to identify relationships which have a bearing on a specific issue and then extrapolate from these relationships to predict future performance or behavior and discover hidden data patterns. Often the sheer volume of data can make the extraction of this business information impossible by manual methods. DM treats as synonym for another popularly used term, Knowledge Discovery in Databases. KDD is the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. DM is a set of techniques which allows extracting useful business knowledge, based on a set of some commonly used techniques such as: Statistical Methods, Case-Based Reasoning, Neural Networks, Decision Trees, Rule Induction, Bayesian Belief Networks, Genetic Algorithms, Fuzzy Sets, Rough Sets, and Linear Regression [4, 36]. DM commonly used in a variety of domains such as: marketing, surveillance and fraud detection in telecommunications, manufacturing process control, the study of risk factors in medical diagnosis, and customer support operations through a better understanding of customers in order to improve sales.

In commerce, RB is defined by buying goods or products in large quantities from manufacturers or importers, either directly or through a wholesaler, and then sells individual items or small quantities to the general public or end user customers. RB is based on the sale of goods from fixed

locations, these locations could be physical (shop or store) and/or virtual over the web. Retailing may include several types of services that can go along with the sale, such as delivery of goods, processing and tracking loyalty card functionality. The process goes from buying products in large quantities from manufacturers, and then sells smaller quantities to the end-user. From a business perspective, DM is mainly used in the Customer Relationship Management (CRM) area, specifically marketing. DM today's applications provide the tool for retailers or decision maker to get precious knowledge that covers the requested field of interest and make sense of their customer data and apply it to business such as: the sales/marketing domain and other business-related areas [4]. It contributes to predict customer purchasing behavior and perform target marketing by using demographic data and historical information, to drive sales suggestions for alternate or related items during a purchase transaction, to identify valuable customers, allowing the CRM team to target them for retention, to point out potential long-term customers who can be a potential target through marketing programs [36], to identify people behavior who are likely to buy new products based on their item categories purchased, to assess the products which are bought together.

This paper is organized as follows: in section 2, the background of DM and its relationship with RB is presented and highlighted by specifying the main major problems faced by retailer. In section 3, we present the Rough Sets (RS) technique and the Rough Sets Attribute Reduction (RSAR) problem followed by a general overview of the literature and a mathematical formulation. Therefore, in section 4, we present a new dynamic solution approach for the RSAR problem based on the Dynamic Programming technique followed by a study of its complexity. In section 5, we describe our solution approach through a numerical example using some well-known datasets followed by discussion and analysis of the results obtained. And finally, we ended by a conclusion concerning this new approach and the related new ideas to be tackled in the future.

II. ROUGH SET THEORY

Pawlak has introduced the theory of RS which is an efficient technique for knowledge discovery in databases [33, 34]. It is a relatively new rigorous mathematical technique to describe quantitatively uncertainty, imprecision and vagueness. It leads to create approximate descriptions of objects for data analysis, optimization and recognition. It is shown to be methodologically significant in the domains of Artificial Intelligence and cognitive science, especially in respect of the representation and of the reasoning with imprecise knowledge, machine learning, and knowledge discovery. In RS theory, the data is organized in a table called decision table. Rows of the decision table correspond to objects, columns correspond to attributes, and class label indicates the class to which each row belongs. The class label is called as decision attribute, the rest of the attributes are the condition attributes. Therefore, the partitions/classes obtained from condition attributes are called elementary sets, and those from the decision attribute(s) are called concepts. Let's consider C for the condition attributes, D for the decision attributes, where $C \cap D = \phi$, and t_j denotes

the j^{th} tuple of the data table. The goal of RS is to understand or construct rules for the concepts in terms of elementary sets, i.e., mapping partitions of condition attributes to partitions of decision attribute [41]. However, a RS is a formal approximation of a crisp set in terms of a pair of sets which give the lower and the upper approximation of the original set. Once the lower and upper approximation is calculated, positive, negative, and boundary regions can be derived from the approximation. Therefore, RS theory defines five regions based on the equivalent classes induced by the attribute values. Lower approximation contains all the objects, which are classified surely based on the data collected, Upper approximation contains all the objects which can be classified probably, Negative region contains the set of objects that cannot be assigned to a given class, Positive region contains the objects that can be unambiguously assigned to a given class, while the Boundary is the difference between the upper approximation and the lower approximation which contains the objects that can be ambiguously (with confidence less than 100%) assigned to a given class.

A. Elements of the rough sets

To illustrate clearly the RS technique, let's consider the main elements of RS theory. Let U be any finite universe of discourse. Let R be any equivalence relation defined on U , which partitions U . Here, (U, R) is the collection of all equivalence classes. Let $X_1, X_2 \dots, X_n$ be the elementary sets of the approximation space (U, R) . This collection is known as knowledge base. Let A be a subset of U .

Elementary sets:

$$R_A = \{X_1, X_2 \dots, X_m\} \text{ where } X_i \text{ denote the elementary sets.} \quad (1)$$

Concepts:

$$R_{Class} = \{Y_1, Y_2 \dots, Y_k\} \text{ where } Y_i \text{ refer to concepts.} \quad (2)$$

Lower approximation: Thus the lower approximation of a concept is the set of those elementary sets that are contained within subset of the concept with probability of 1.

$$\underline{R}_A(Y_i) = \cup X_j, \text{ where } X_j \subseteq Y_i \quad (3)$$

Upper approximation: The upper approximation of a concept is the set of those elementary sets that share some objects with the concept (non-zero probability).

$$\overline{R}_A(Y_i) = \cup X_j, \text{ where } X_j \cap Y_i \neq \phi \quad (4)$$

Positive region: Thus the positive region of a concept is the set of those elementary sets that are subset of the concept. Positive region would generate the strongest rule with 100% confidence.

$$POS_A(Y_i) = \underline{R}_A(Y_i) \quad (5)$$

Boundary region: Thus the boundary region of a concept is the set of those elementary sets that have something to say about the concept, excluding the positive region. It consists of those objects that can neither be ruled in nor ruled out as members of the target set. These objects can be ambiguously (with confidence less than 100%) assigned the class denoted by Y_i . Hence, it is trivial that if $BND_A = \phi$, then A is exact. This approach provides a mathematical tool that can be used to find out all possible reducts.

$$BND_A(Y_i) = \bar{R}_A(Y_i) - \underline{R}_A(Y_i) \quad (6)$$

Negative region: Thus the negative region of a concept is the set of those elementary sets that have nothing to say about the concept. These objects cannot be assigned the class denoted by Y_i (their confidence of belonging to class Y_i is in fact 0%)

$$NEG_A(Y_i) = U - \bar{R}_A(Y_i) \quad (7)$$

Concept Set: Concept set is the equivalence relation from the class and elementary set are equivalence relation from attributes. As mentioned above, the goal of the rough set is to understand the concept in term of elementary set. In order to map between elementary set and concept, lower and upper approximation must first defined. Then positive, boundary and negative regions can be defined based on the approximations to generate rules for categorization. Once the effect of subclass of concept is defined, the last step before rule generation is to define the net effect on entire set of concepts. Given effect of subset of concept $POS_A(Y_i)$, the net effect on entire set of concepts is defined as:

$$\begin{aligned} POS_A(Y) &= \bigcup_{i=1}^k POS_A(Y_i) \\ BND(Y) &= \bigcup_{i=1}^k BND_A(Y_i) \\ NEG_A(Y) &= U - \bigcup_{i=1}^k \bar{R}_A(Y_i) \end{aligned} \quad (8)$$

Generating rules: There are two kinds of rules that can be generated from the POS and the BND regions respectively. For any $X_i \in POS_A(Y_j)$, we can generate a 100% confidence rule of the form: If X_i then Y_j (or $X_i \Rightarrow Y_j$). For any $X_i \in BND_A(Y_i)$ we can generate a <100% confidence rule of the form: If X_i then Y_j (or $X_i \Rightarrow Y_j$), with confidence given as:

$$conf = \frac{|X_i \cap Y_j|}{|X_i|} \quad (9)$$

Assessment a rule: As mentioned above, the goal of the RS is to generate a set of rules that are high in dependency, discriminating index, and significance. There are three methods of assessing the importance of an attribute:

- **Dependency:** How much does a class depends on A (subset of attribute)

$$\lambda_A(class) = \frac{POS_A(class)}{|U|} \quad (10)$$

- **Discriminating Index:** Attributes A's ability to distinguish between classes

$$\begin{aligned} \beta_A(class) &= \frac{|U - BND_A(class)|}{|U|} \\ &= \frac{|POS_A(class) \cup NEG_A(class)|}{|U|} \end{aligned} \quad (11)$$

- **Significance:** How much does the data depend on the removal of A

$$\delta_A(class) = \delta_{A_1, A_2, \dots, A_d}(class) - \delta_{A_1, A_2, \dots, A_d - A}(class) \quad (12)$$

Significance of A is computed with regard to the entire set of attributes. If the change in the dependency after removing A is large, then A is more significant.

B. Rough Set Based Attribute Reduction

1) Literature overview

Attribute or feature selection is to identify the significant features, eliminate the irrelevant of dispensable features to the learning task, and build a good learning model. It refers to choose a subset of attributes from the set of original attributes. Attribute or feature selection of an information system is a key problem in RS theory and its applications. Using computational intelligence tools to solve such problems has recently fascinated many researchers. Computational intelligence tools are practical and robust for many real-world problems, and they are rapidly developed nowadays. Computational intelligence tools and applications have grown rapidly since its inception in the early nineties of the last century [5, 8, 16, 24]. Computational intelligence tools, which are alternatively called soft computing, were firstly limited to fuzzy logic, neural networks and evolutionary computing as well as their hybrid methods [16, 40]. Nowadays, the definition of computational intelligence tools has been extended to cover many of other machine learning tools. One of the main computational intelligence classes is Granular Computing [25, 40], which has recently been developed to cover all tools that mainly invoke computing with fuzzy and rough sets.

However, some classes of computational intelligence tools, like memory-based heuristics, have been involved in solving information systems and DM applications like other well-known computational intelligence tools of evolutionary computing and neural networks. One class of the promising computational intelligence tools is memory-based heuristics, like Tabu Search (TS), which have shown their successful performance in solving many combinatorial search problems [10, 32]. However, the contributions of memory-based heuristics to information systems and data mining applications are still limited compared with other computational

intelligence tools like evolutionary computing and neural networks.

A decision table may have more than one reduct. Anyone of them can be used to replace the original table. Finding all the reducts from a decision table is NP-Hard [37]. Fortunately, in many real applications it is usually not necessary to find all of them and it is enough to compute one such reduct is sufficient [45]. A natural question is which reduct is the best if there exist more than one reduct. The selection depends on the optimality criterion associated with the attributes. If it is possible to assign a cost function to attributes, then the selection can be naturally based on the combined minimum cost criteria. In the absence of an attribute cost function, the only source of information to select the reduct is the contents of the data table [26, 27]. For simplicity, we adopt the criteria that the best reduct is the one with the minimal number of attributes and that if there are two or more reducts with same number of attributes, then the reduct with the least number of combinations of values of its attributes is selected. Zhong et al. have applied Rough Sets with Heuristics (RSH) and Rough Sets with Boolean Reasoning (RSBR) for attribute selection and discretization of real-valued attributes [44]. Calculation of reducts of an information system is a key problem in RS theory [20, 21, 34, 38]. We need to get reducts of an information system in order to extract rule-like knowledge from an information system. Reduct is a minimal attribute subset of the original data which has the same discernibility power as all of the attributes in the rough set framework. Obviously, reduction is an attribute subset selection process, where the selected attribute subset not only retains the representational power, but also has minimal redundancy. Many researchers have endeavored to develop efficient algorithms to compute useful reduction of information systems, see [25] for instance. Besides mutual information and discernibility matrix based attribute reduction methods, they have developed some efficient reduction algorithms based on computational intelligence tools of genetic algorithm, ant colony optimization, simulated annealing, and others [16, 20, 21]. These techniques have been successfully applied to data reduction, text classification and texture analysis [25]. Actually, the problem of attribute reduction of an information system has made great gain from rapid development of computational intelligence tools.

In the literature, much effort has been made to deal with the attribute reduction problem [6, 15, 17, 19, 20, 21, 38, 39, 43]. In their works, four computational intelligence methods, GenRSAR, AntRSAR, SimRSAR, and TSAR have been presented to solve the attribute reduction problem. GenRSAR is a genetic-algorithm-based method and its fitness function takes into account both the size of subset and its evaluated suitability. AntRSAR is an ant colony-based method in which the number of ants is set to the number of attributes, with each ant starting on a different attribute. Ants construct possible solutions until they reach a RS reduct. SimRSAR employs a simulated annealing based attribute selection mechanism. SimRSAR tries to update solutions, which are attribute

subsets, by considering three attributes to be added to the current solution or to be removed from it. Optimizing the objective function attempts to maximize the RS dependency while minimizing the subset cardinality. The TSAR method proposed in [15] is based on using the Tabu Search (TS) neighborhood search methodology for searching reducts of an information system. TS is a heuristic method originally proposed by Glover in [11]. It has primarily been proposed and developed for combinatorial optimization problems [10, 12, 13], and has shown its capability of dealing with various difficult problems [10, 32]. Moreover, there have been some attempts to develop TS for continuous optimization problems [14]. TS neighborhood search is based on two main concepts; avoiding return to a recently visited solution, and accepting downhill moves to escape from local maximum information. Some search history information is reserved to help the search process to behave more intelligently. Specifically, the best reducts found so far and the frequency of choosing each attribute are saved to provide the diversification and intensification schemes with more promising solutions. TSAR invokes three diversification and intensification schemes; diverse solution generation, best reduct shaking which attempts to reduce its cardinality, and elite reducts inspiration.

The benefits of attribute reduction or feature selection are twofold: it considerably decreased the computation time of the induction algorithm and increased the accuracy of the resulting mode [41]. All feature selection algorithms fall into two categories: the filter approach and the wrapper approach. In the filter approach, the feature selection is performed as a preprocessing step to induction. The filter approach is ineffective in dealing with the feature redundancy. Some of the algorithms in the Filter approach methods are Relief, Focus, Las Vegas Filter (LVF), Selection Construction Ranking using Attribute Pattern (SCRAP), Entropy-Based Reduction (EBR), Fractal Dimension Reduction (FDR). In Relief each feature is given a relevance weighting that reflects its ability to discern between decision class labels [23]. Orłowska, in [30], conducts a breadth-first search of all feature subsets to determine the minimal set of features that can provide a consistent labeling of the training data. LVF employs an alternative generation procedure that of choosing random features subsets, accomplished by the use of a Las Vegas algorithm [26, 27]. SCRAP is an instance based filter, which determines feature relevance by performing a sequential search within the instance space [31]. Jensen et al. proposed EBR which is based on the entropy heuristic employed by machine learning techniques such as C4.5 [18]. EBR is concerned with examining a dataset and determining those attributes that provide the most gain in information. FDR is a novel approach to feature selection based on the concept of fractals – the self-similarity exhibited by data on different scales [42]. In the wrapper approach [22], the feature selection is “wrapped around” an induction algorithm, so that the bias of the operators that defined the search and that of the induction algorithm interact mutually. Though the wrapper approach suffers less from feature interaction, nonetheless, its running time would make the wrapper approach infeasible in practice, especially if there are many features, because the wrapper approach keeps running the induction algorithms on different subsets from the entire

attributes set until a desirable subset is identified. We intend to keep the algorithm bias as small as possible and would like to find a subset of attributes that can generate good results by applying a suite of DM algorithms. Some of the Wrapper approach methods are Las Vegas Wrapper (LVW) and neural network-based feature selection. The LVW algorithm is a wrapper method based on LVF algorithm [20, 21]. This again uses a Las Vegas style of random subset creation which guarantees that given enough time, the optimal solution will be found. Neural network-based feature selection is employed for backward elimination in the search for optimal subsets [42].

2) Mathematical modeling

The purpose of the Rough Set Attribute Reduction (RSAR) has been employed to remove redundant conditional attributes from discrete-valued datasets, while retaining their information content [37]. Attribute reduction has been studied intensively for the past one decade [20, 21, 22, 23, 28, 29]. This approach provides a mathematical tool that can be used to find out all possible reducts. However, this process is NP-hard [34], if the number of elements of the universe of discourse is large. The RSAR has as central concept the indiscernibility [41]. Let $I = (U, A)$ be an information system, where U is a non-empty set of finite objects (the universe of discourse); A is a non-empty finite set of attributes such that:

$$a: U \rightarrow V_a \quad (13)$$

$\forall a \in A, V_a$ being the value set of attribute a . In a decision system, $A = \{C \cup D\}$ where C is the set of conditional attributes and D is the set of decision attributes. With any $P \subseteq A$ there is an associated equivalence relation $IND(P)$:

$$IND(P) = \{x, y \in U^2 / \forall a \in P, a(x) = a(y)\} \quad (14)$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P . An important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes Q depends totally on a set of attributes P , denoted $P \Rightarrow Q$, if all attribute values from Q are uniquely determined by values of attributes from P . Dependency can be defined in the following way:

For $P, Q \subseteq A$, Q depends on P in a degree k ($0 \leq k \leq 1$), denoted $P \Rightarrow_k Q$, if:

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|}$$

$$\text{where } \begin{cases} Q \text{ depends totally on } P & \text{if } k = 1 \\ Q \text{ depends partially on } P & \text{if } 0 < k < 1 \\ Q \text{ does not depend on } P & \text{if } k = 0 \end{cases}$$

By calculating the change in dependency when an attribute is removed from the set of considered conditional attributes, a measure of the significance of the attribute can be obtained.

The higher the change in dependency, the more significant the attribute is. If the significance is 0, then the attribute is dispensable. More formally, given P, Q and an attribute $x \in P$, the significance of attribute x upon Q is defined by:

$$\sigma_P(Q, x) = \gamma_P(Q) - \gamma_{P-\{x\}}(Q) \quad (15)$$

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same quality of classification as the original. In the context of decision systems, a reduct is formally defined as a subset R of the conditional attribute set C such that $R(D) = C(D)$. A given dataset may have many attribute reduct sets, and the collection of all reducts is denoted by:

$$R = \{X: X \subseteq C, \gamma_X(D) = \gamma_C(D)\} \quad (16)$$

The intersection of all the sets in R is called the core, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the dataset. In RSAR, a reduct with minimum cardinality is searched for; in other words an attempt is made to locate a single element of the minimal reduct set $R_{\min} \subseteq R$:

$$R_{\min} = \{X: X \subseteq R, \forall Y \in R, |X| \leq |Y|\} \quad (17)$$

The most basic solution to locating such a subset is to simply generate all possible subsets and retrieve those with a maximum RS dependency degree. Obviously, this is an expensive solution to the problem and is only practical for very simple datasets. Most of the time only one reduct is required as, typically, only one subset of features is used to reduce a dataset, so all the calculations involved in discovering the rest are pointless. Another basic way of achieving this is to calculate the dependencies of all possible subsets of C . Any subset X with $\gamma_X(D) = 1$ is a reduct; the smallest subset with this property is a minimal reduct. However, for large datasets this method is impractical and an alternative strategy is required.

An algorithm called "QuickReduct" algorithm, borrowed from [28], attempts to calculate a minimal reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in $\gamma_P(Q)$, until this produces its maximum possible value for the dataset (usually 1). However, it has been proved that this method does not always generate a minimal reduct, as $\gamma_P(Q)$ is not a perfect heuristic. It does result in a close to minimal reduct, though, which is still useful in greatly reducing dataset dimensionality. In order to improve the performance of the "QuickReduct" algorithm, an element of pruning can be introduced [41]. By noting the cardinality of any pre-discovered reducts, the current possible subset can be ignored if it contains more elements. However, a better approach is needed in order to avoid wasted computational effort. The pseudo code of the "Quickreduct" is given below:

QUICKREDUCT(C, D)

C, the set of all conditional features;

D, the set of decision features.

$R \leftarrow \{ \}$

do

$T \leftarrow R$

$\forall x \in (C - R)$

if $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$ where $\gamma_R(D) = \text{card}(POS_x(D)) / \text{card}(U)$

$T \leftarrow R \cup \{x\}$

$R \leftarrow T$

until $\gamma_R(D) = \gamma_C(D)$

return R

An intuitive understanding of “QuickReduct” implies that, for a dimensionality of n, n! evaluations of the dependency function may be performed for the worst-case dataset. From experimentation, the average complexity has been determined to be approximately $O(n)$ [44].

III. DYNAMIC ROUGH SETS ATTRIBUTE REDUCTION APPROACH

A. Solving approach by Dynamic Programming

An intelligent approach using Dynamic Programming (DP) is applied to deal with the optimization problem of RSAR where the constraints are involved in verifying the validity of developed solution. In fact, as shown in the choice of the criterion, it is to maximize the dependence degree in our solution which in principle meets all the constraints level. Using DP technique leads to generate dynamic equivalence subsets of attributes. It becomes a problem of discrete combinatorial optimization and applying DP approach leads to get an exact solution. This can be effective for the treatment of combinatorial optimization problems, in a static, dynamic or stochastic, but only if the level constraints are present in limited numbers [3]. Indeed, scaling constraints level lead to address every step of the optimization process exponentially growing number of states within the parameters sizing the problem, making it impossible to process numerically the problem of consequent dimensions. The proposed method, called Dynamic Rough Sets Attribute Reduction (DRSAR), shows promising and competitive performance compared with some other computational intelligence tools in terms of solution qualities since it produces optimistic reduct attribute subsets.

To implement an approach based on DP technique, it is necessary to define two key elements: the states and the stages and the various possible levels of constraints associated with dynamic allocation. Solving the problem of dynamic attributes reduction to build the minimal subsets of attributes by the proposed schema leads to the following mathematical formulation:

J : is the number of stages which is associated to the number of attributes;

I : is the number of states which is based on the super set of attributes;

E_j : is the number of states associated to stage j;

X_j : represents the decision vector taken at stage j;

$\sum_{j=1}^J p_{ij} x_{ij}$: represents the sum of weighted associated to a

sequence of decisions $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_j)$ which starts from the initial state e_0 to the current state e_j ;

$TR_{ij}(e_{i,j-1}, x_{ij}) = e_{ij}$: represents the state transition ($DEP_{ij} = DEP_{i,j-1} + p_{ij}x_{ij}$) where DEP represents the dependency related to a transition.

Therefore, solving this problem involves finding an optimal sequence $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_j)$ that starts from the initial state e_0 brings us to the state e_j while maximizing the following function:

$$MAX \left\{ \sum_{j=1}^J p_{ij} x_{ij} / x_{ij} \in X_j; e_{ij} = TR_{ij}(e_{i,j-1}, x_{ij}), \forall j = 1 \dots J \right\} \quad (18)$$

The principle of optimality of dynamic programming, shows that whatever the decision in stage J brings us from state $e_{j-1} \in E_{j-1}$ to state $e_j \in E_j$, the portion of the policy between e_0 and e_{j-1} must be optimal. However, applying this principle of optimality, we can calculate step by step $AFF(J, e_j)$ using the following recurrence equation:

$$AFF(j, e_j) = \left\{ x_{ij} \in X_j / e_{ij} = TR_{ij}(e_{i,j-1}, x_{ij}) \right\} \left\{ p_{ij} x_{ij} + AFF(j-1, e_{j-1}) \right\} \quad (19)$$

with $AFF(0, e_0) = 0$

However, if the weights p_{ij} should be such that they take into account the dependence degree reached at the tree of the solutions deployed by DP, it seems that for each state of each stage it is necessary to reassess the weights effective following the path leading to it. Thus, an exact resolution scheme by DP can be implemented directly.

B. Complexity

The algorithm based on the pattern resolution by the DP consists of three key parameters to evaluate its performance [7]. These three parameters are the number of states, the number of stages and the number of calls to the procedure that calculates the dependence weights associated with each path in the tree solutions. Let I be the number of states which is based on super set of attributes, and J is also the number of stages associated to attributes. Remember also, that a calculation of dependency weight must be made for each path in the graph. Since the solution algorithm follows the scheme of solving the DP, then it is to treat the problem as belonging to a family of similar problems and linking them through the principle of optimality.

1) Temporal Complexity

The effectiveness of the algorithm described above is assessed by temporal complexity depending on the number of iterations needed to obtain the solution (s). The evaluation of the number of iterations is done in the worst case. Indeed, it is impossible in the general case to count the exact number of paths to build in order to solve the optimization problem. The number of paths traversed in each stage is estimated to I^2 .

A set of constraints must be checked at each stage in the process of resolution, even to each path. A subset of these constraints is considered in our case. The computation time required to check all of these constraints is of the order of:

$$\sim O(I \times J^2) \quad (20)$$

Thus, the temporal complexity associated with each step in resolution (a step involves I^2 possible paths) is the order of:

$$\sim O(I^3 \times J^2) \quad (21)$$

The temporal complexity associated with treating the whole problem ("J" stages) is the order of:

$$\sim O(I^3 \times J^3) \quad (22)$$

2) Space Complexity

The memory space required for the algorithm developed here depends on the number of states and the number of stages considered. Indeed, the number of states set the maximum number of vertices to be considered in one step. This number multiplied by the number of stages defined here also helps to set the maximum number of vertices in the graph solutions. Thus, the number of variables to remember throughout the resolution process is the order of:

$$\sim O(I \times J) \quad (23)$$

IV. CASE STUDY: IMPLEMENTATION AND RESULTS

A. Numerical case

The proposed solution strategy has been adapted to a large retailer business. It considers the case of an international retailer having many stores with a daily average of 3000 transactions by store. We are using a large database having a large number of attributes and which cover the transactions of last 3 years. It contributes in dealing with any critical classification process. A growing RB market, where items' numbers and relationships are becoming more and more complex, is highly important since it is closely related to optimization of profit. The aim of this study is to reduce the initial number of attributes leading to reduce the complexity while preserving approximately the pattern of the predictive model. Simulations and visual analysis will be used to validate the accuracy of the improved approach. In our case, we have considered three problems that may interest the large retailer

business such as: classifying the customers, classifying the items, and applying discount on item. Our algorithm simulates these real business cases by allowing the experts to define a number of attributes that describe the business case in order to be able to get the appropriate decisions. These attributes can be related to pertinent information such as: products, products category, customers, personal information, suppliers, times and seasons, price, quantity, events, and others related attributes gathered from appropriate databases. Moreover, the experts express their thoughts as added inputs to our algorithm beside the statically defined input. Therefore, data corresponding to the appropriate set of attributes are gathered and collected from a rich data warehouse oriented business based on experts' opinions. For example, experts may define some features deduced such as: the amount paid for advertising for an item over a period, the number of transactions containing an item, the percentage of transactions related to other items of the same category, the number of transactions in which an item is sold in single, etc. These new calculated attributes have distinct importance relative to the experts.

B. Performance evaluation

This section describes some characteristics of tests conducted using the DRSAR solution in order to generate dynamically the different optimal RSAR. We proceed to evaluate the performance of this new solution by analyzing the responding time and some various sensitivity features that can be conducted through the use of some metrics measure (accuracy, precision, recall). Also, we propose a comparison with some computational intelligence tools retained from the literature in order to compare the performance of the DRSAR regarding the existing ones.

The DRSAR solution method has been developed using Visual C++ on a PC computer equipped with a P-IV processor. Concerning the response time consumed by the system and which is stated in table 1, it presents a much shorter computing time than with pre-existent computational intelligence or mathematical programming methods and this response time is compatible with online use in an operations management environment. The solutions obtained by the proposed method have appeared to be significantly superior to those obtained from lengthy manual procedures or those based on some computational intelligence tools such as: genetic algorithms, simulated annealing, tabu search, ant colony, etc. Several experiments were realized in order to test and compare the classification algorithm for three cases based on a set of attributes defined by experts before and after applying DRSAR. The results are shown in the tables (2, 3, 4). For each case, it presents the number of records, the initial number of attributes, and the reduced number of attributes achieved after applying DRSAR. We report also some metrics measure (accuracy, precision, and recall) to evaluate the quality of the predictive model. We show that the number of attributes is dramatically reduced without assigning the quality of the classification. So, it is clear that our approach is efficient while its complexity is decreased by reducing the number of

attributes. Moreover, the metrics measures show a slight modification while the optimal subsets are dealt with instead of considering the whole attributes defined by the experts.

In order to achieve the performance evaluation of the DRSAR, we compare it with the some intelligence computational tools developed in the literature and which dealt with the reduction of attribute sets in RS such as: Ant Colony optimization for Rough Set Attribute Reduction (AntRSAR) [19, 20, 21]; Simulated Annealing for Rough Set Attribute Reduction (SimRSAR) [19]; Genetic Algorithm for Rough Set Attribute Reduction (GenRSAR) [19, 20, 21]; and Tabu Search Attribute Reduction (TSAR) [15]. The results of this comparison are reported in Table 5 and figures (1, 2). The results in Table 5 focus on the reduced number of attributes achieved by each method after several runs and the corresponding dependency (Dep.) degree function.

The results shown in the above table show that the DRSAR approach is the best since it is based on an optimistic method while the others are of type greedy heuristics. DRSAR outperforms all the considered methods TSAR, AntRSAR, GenRSAR, and SimRSAR for any datasets (Figure 1). The performance of TSAR and AntRSAR is comparable since there is no significant difference between them for any datasets. We note here that TSAR outperforms AntRSAR for dataset 2, while it is not the case for dataset 1. TSAR and AntRSAR outperform GenRSAR and SimRSAR methods for all tested datasets. SimRSAR outperforms GenRSAR for any dataset except the dataset 2. Concerning the dependency function degree, we note here that the degree of dependency associated to the reduced number of attributes is optimal while using DRSAR. AntRSAR and TSAR are more performance than GenRSAR and SimRSAR (Figure 2).

We conclude that the proposed method, shows promising and competitive performance compared with others computational intelligence tools in terms of solution qualities. Moreover, DRSAR shows a superior performance in saving the computational costs.

TABLE I. COMPARING THE RELATED FEATURES BY USING DRSAR

Cases	Initial number of concept attributes	Minimum Reduced attributes	Computing time (sec.)
A-Customers classification	28	19	1.65
B-Items classification	52	41	8.81
C-Applying discount on item	83	68	32.35

TABLE II. CONFUSION MATRIX RESULTS FOR CUSTOMERS CLASSIFICATION BEFORE/AFTER DRSAR

# records:	417.200	# Initial set of attributes	28	# of attributes in the reduced DRSAR	19			
Count		Predicted class		Count		Predicted class		
Actual class		<i>Solvent</i>	<i>Insolvent</i>	Actual class	<i>Solvent</i>	<i>Insolvent</i>		
	<i>Solvent</i>	319535	7705		<i>Solvent</i>	318675 (99.73%)	8920 (86.38%)	
	<i>Insolvent</i>	8650	81310		<i>Insolvent</i>	7595 (88.19%)	82010 (99.75%)	
Accuracy	96.03	Error rate	0.92%	Precision	97.75	Recall	97.17	

TABLE III. CONFUSION MATRIX RESULTS FOR ITEMS CLASSIFICATION BEFORE/AFTER DRSAR

# records:	933.820	# Initial set of attributes	52	# of attributes in the reduced DRSAR	41			
Count		Predicted class		Count		Predicted class		
Actual class		<i>Attractive</i>	<i>Non-Attractive</i>	Actual class	<i>Attractive</i>	<i>Non-Attractive</i>		
	<i>Attractive</i>	822430	3145		<i>Attractive</i>	822217 (99.74%)	3133 (99.62%)	
	<i>Non-Attractive</i>	6725	101520		<i>Non-Attractive</i>	8740 (77.05%)	99730 (98.73%)	
Accuracy	98.72	Error rate	0.43%	Precision	98.95	Recall	99.62	

TABLE IV. CONFUSION MATRIX RESULTS FOR APPLYING DISCOUNT ON ITEM BEFORE/AFTER DRSAR

# records:	933.820	# Initial set of attributes	83	# of attributes in the reduced DRSAR	68
------------	---------	-----------------------------	----	--------------------------------------	----

Count		Predicted class	
		Yes	No
Actual class	Yes	12746	170
	No	98	2986

Count		Predicted class	
		Yes	No
Actual class	Yes	12739 (99.94%)	166 (97.65%)
	No	127 (77.16%)	2968 (99.74%)

Accuracy	98.16	Error rate	0.36%
Precision	99.01	Recall	98.71

TABLE V. REPORTED RESULTS BASED ON THE NUMBER OF ATTRIBUTES AND DEPENDENCY DEGREE FUNCTION

# records	# Initial Sets of attributes	DRSAR		GenRSAR		AntRSAR		TSAR		SimRSAR	
		# attr.	Dep.	# attr.	Dep.	# attr.	Dep.	# attr.	Dep.	# attr.	Dep.
417.200	28	19	1	24	0.68	21	0.78	22	0.77	23	0.69
933.820	52	41	1	45	0.64	43	0.72	43	0.74	47	0.66
933.820	83	68	1	78	0.59	73	0.64	72	0.69	74	0.61

Figure 1. Comparison of methods in RSAR based on the # of attributes

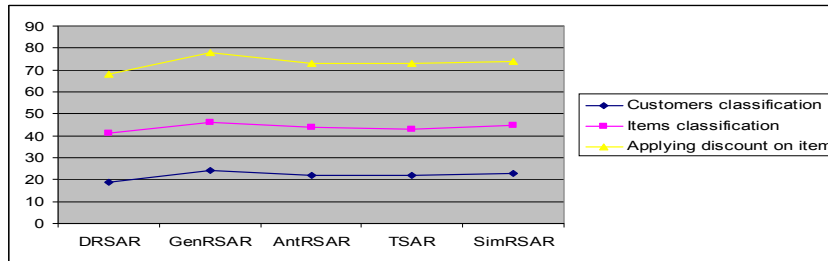
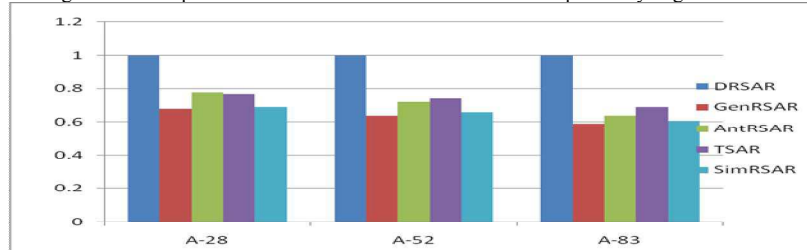


Figure 2. Comparison of methods in RSAR based on the dependency degree function



V. CONCLUSION AND PERSPECTIVES

In this communication, a new solution approach is proposed in order to reduce the complexity of the classification problems faced by Retailer business. Moving on from traditional heuristic methods, an optimal one based on Dynamic Programming, called DRSAR, is proposed. The proposed approach produces an exact solution in mathematical terms and appears to be quite adapted, if necessary, to the operational context of the retailer business and provides, through a comprehensive process for the decision-makers, improved legible solutions. This technique provides a dynamic solution that can be executed on any classification problems without taking into consideration the classification techniques that will

be used later. It permits to explore the optimal sets of significant attributes that can drive the profit of the company and reduced the process complexity. Numerical experiments on three classification problem cases have been considered and performed in order to validate the proposed solution approach for retailer business. It had been tested on a real database with 3 years historical data. The obtained results had been found plausible. Comparisons with other computational intelligence tools have revealed that DRSAR is promising and it is less expensive in computing the dependency degree function.

In perspectives, a Decision Support System should integrate many other aspects that may be highly relevant such as:

Customer Retention, Buyer Behavior, Cost/Utilization, Halo and Cannibalization, Detect positive and negative correlation among items, Quality Control, Inventory, etc. This is performed in order to improve the efficiency of business retailer operations.

REFERENCES

- [1] T. Bhavani, K. Latifur, A. Mamoun, and W. Lei, "Design and Implementation of data mining tools. Data Mining Techniques and Applications", Web Data Management and Mining, 2009.
- [2] C. Verceles, "Business Intelligence: Data Mining and Optimization for Decision Making", Wiley & Sons, 2009, ISBN: 978-0-470-51138-1.
- [3] R.E. Bellman, "Dynamic Programming", Princeton University Press, 1957.
- [4] G. Linoff and M. Berry, "Data mining techniques for marketing, sales and customer relationship management", 3rd Ed., Wiley & Sons, 2004.
- [5] E.K. Burke and G. Kendall, "Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques", Springer-Verlag, Berlin, 2005.
- [6] A. Chouchoulas and Q. Shen, "Rough set-aided keyword reduction for text categorisation", Applied Artificial Intelligence, 2001. Vol. 15, pp. 843-873.
- [7] W. Moudani and F. Mora-Camino, "A Dynamic Approach for Aircraft Assignment and Maintenance Scheduling by Airlines", Journal of Air Transport Management, 2000, Vol. 4 (1), pp. 233-237.
- [8] A.P. Engelbrecht, "Computational Intelligence: An Introduction", John Wiley & Sons, Chichester, England, 2003.
- [9] K.J. Ezawa and S.W. Norton, "Constructing Bayesian networks to predict uncollectible telecommunications accounts", IEEE Intelligent Systems, 1996, Vol. 11 (5), pp. 45-51.
- [10] F. Glover and M. Laguna, "Tabu Search", Kluwer Academic Publishers, Boston, MA, USA, 1997.
- [11] F. Glover, "Future paths for integer programming and links to artificial intelligence", Computers and Operations Research, 1986, Vol. 13, pp. 533-549.
- [12] F. Glover, "Tabu search-Part I", ORSA Journal on Computing, 1989, Vol. 1, pp.190-206.
- [13] F. Glover, "Tabu search-Part II", ORSA Journal on Computing, 1990, Vol. 2, pp. 4-32.
- [14] A. Hedar and M. Fukushima, "Tabu search directed by direct search methods for nonlinear global optimization", European Journal of Operational Research, 2006, Vol. 170, pp. 329-349.
- [15] A. Hedar, J. Wangy, and M. Fukushima, "Tabu search for attribute reduction in rough set theory", Journal of Soft Computing - A Fusion of Foundations, Methodologies and Applications, Springer-Verlag Berlin, Heidelberg, 2008, Vol. 12 (9).
- [16] W. Moudani and F. Mora-Camino, "Management of Bus Driver Duties using data mining," International Journal of Applied Metaheuristic Computing (IJAMC), 2011, Vol 2 (2)
- [17] J. Jelonek, K. Krawiec, and R. Slowinski, "Rough set reduction of attributes and their domains for neural networks", Computational Intelligence, 1995, Vol. 11, pp. 339-347.
- [18] R. Jensen and Q. Shen, "A Rough Set - Aided system for Sorting WWW Bookmarks", Web Intelligence: Research and Development, 2001, pp. 95-105.
- [19] R. Jensen and Q. Shen, "Finding rough set reducts with ant colony optimization". Proceedings of the 2003 UK Workshop on Computational Intelligence, 2003, pp. 15-22.
- [20] R. Jensen and Q. Shen, "Fuzzy-rough attribute reduction with application to web categorization", Fuzzy Sets and Systems, 2004, Vol. 141 (3), pp. 469-485.
- [21] R. Jensen and Q. Shen, "Semantics-preserving dimensionality reduction: Rough and fuzzy rough-based approaches", IEEE Transactions on Knowledge and Data Engineering, 2004, Vol. 16, pp.1457-1471.
- [22] G.H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem". Proceedings of 11th Intl. Conf. on Machine Learning, 1994, pp.121-129.
- [23] K. Kira and L.A. Rendell, "The Feature selection Problem: Traditional Methods and a New Algorithm". Proceedings of AAAI, MIT Press, 1992, pp. 129-134.
- [24] A. Konar, "Computational Intelligence: Principles, Techniques and Applications", Springer-Verlag, Berlin, 2005.
- [25] T.Y. Lin, Y.Y. Yao, and L.A. Zadeh, "Data Mining, Rough Sets and Granular Computing", Springer-Verlag, Berlin, 2002.
- [26] H. Liu and R. Setiono, "A probabilistic approach to feature selection: a filter solution". Proceedings of the 9th International conference on Industrial and Eng. Applications of AI and ES, 1996, pp. 284-292.
- [27] H. Liu and R. Setiono, "Feature selection and classification-A probabilistic wrapper approach", Proceedings of the 9th Intl. Conf. on Indust. and Eng. Applications of AI and ES, 1996, pp. 419-424.
- [28] H. Liu and H. Motoda, "Feature Extraction Construction and Selection: A Data mining Perspective", Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, 1998.
- [29] M. Modrzejewski, M., "Feature Selection Using Rough Sets Theory", Proceedings of the 11th International Conference on Machine Learning, 1993, pp. 213-226.
- [30] E. Orłowska, "Incomplete Information: Rough Set Analysis", Physica-Verlag, Heidelberg, 1998.
- [31] B. Raman and T.R. Loerger, "Instance-based filter for feature selection", Journal of Machine Learning Research, 2002, pp. 11-23.
- [32] C. Rego and B. Alidaee, "Metaheuristic Optimization via Memory and Evolution", Springer-Verlag, Berlin, 2005.
- [33] Z. Pawlak, "Rough Sets", Intl. Journal of Computer and Information Sciences, 1982, Vol. 11(5), pp.341- 356.
- [34] Z. Pawlak, "Rough Sets: Theoretical aspects of reasoning data", Kluwer Academic Publishers, 1991.
- [35] J.F. Peters and A. Skowron, "Transactions on Rough Sets 1", Springer-Verlag, Berlin, 2004.
- [36] D. Pyle, "Business Modeling and Data Mining", Morgan Kaufmann Publishers, 2003.
- [37] Q. Shen and A. Chouchoulas, "A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems", Eng. Applications of Artificial Intelligence, 2000, Vol. 13(3), pp. 263-278.
- [38] R.W. Swiniarski and A. Skowron, "Rough set methods in feature selection and recognition", Pattern Recognition Letters, 2003, Vol. 24, pp. 833-849.
- [39] S. Tan, "A global search algorithm for attributes reduction", Advances in Artificial Intelligence, G.I. Webb and X. Yu (eds.), LNAI 3339, 2004, pp. 1004-1010.
- [40] Tettamanzi, A., Tomassini, M., and Janben, J. (2001) Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems. Springer-Verlag, Berlin
- [41] K. Thangavel, Q. Shen, and A. Pethalakshmi, "Application of Clustering for Feature selection based on rough set theory approach", AIML Journal, 2006, Vol. 6 (1), pp.19-27.
- [42] C. Traina, L. Wu, and C. Faloutsos, "Fast Feature selection using the fractal dimension", Proceeding of the 15th Brazilian Symposium on Databases (SBBD), 2000.
- [43] L.Y. Zhai, L.P. Khoo, and S.C. Fok, "Feature extraction using rough set theory and genetic algorithms- an application for simplification of product quality evaluation", Computers & Industrial Engineering, 2002, Vol. 43, pp. 661-676.
- [44] N. Zhong and A. Skowron, "A Rough Set-Based Knowledge Discovery Process", Intl. Journal of App. Mathematics and Computer Sciences, 2001, Vol. 11 (3), pp.603-619.
- [45] X. Hu, T.Y. Lin, and J. Jianchao, "A New Computation Model for Rough Sets Based on Database Systems", Lecture Notes in Computer Science, Vol. 2737/2003, pp. 381-390, 2003.