Optimistic Rough Sets Attribute Reduction using Dynamic Programming

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

Optimistic Rough Sets Attribute Reduction using Dynamic Programming

Walid MOUDANI[1], Ahmad SHAHIN[2], Fadi CHAKIK[2], and Félix Mora-Camino[3]
[1] Dept. of Business Information System, Faculty of Business, Lebanese University, Lebanon
[2] LaMA – Liban, Lebanese University, Lebanon.
[3] Air 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 propose the 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 (RB). 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 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 Retail Business Company stores and generates tremendous amounts of raw and heterogeneous data that provides rich fields for Data Mining (DM). 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.

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 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. DATA MINING AND ITS APPLICATIONS IN RETAIL BUSINESS (RB)

II.1 Introduction to Data mining

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 [5]. 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 or "mining" 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 [2, 3]. 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.

II.2 Contributions of Data Mining in the case of Retail Business

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 [3]. 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 [2], 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.

III. ROUGH SET THEORY

In 1982, Pawlak introduced the theory of RS [23, 24]. Rough set theory is an efficient technique for knowledge discovery in databases. 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. 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 [31]. However, a rough set 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.

III.1 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 $X_1, X_2 \cdots, X_n$ 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:

$$R_A = \{X_1, X_2 \cdots, X_m\} \text{ where } X_i \text{ denote the elementary sets.}$$

Concepts:

$$R_{Class} = \{Y_1, Y_2 \cdots, Y_k\} \text{ where } Y_i \text{ refer to concepts}$$

Lower approximation (of a concept): 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.

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

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)

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

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

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

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

$$BND_A(Y_i) = \overline{R}_A(Y_i) - \underline{R}_A(Y_i)$$

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%!)

$$NEG_A(Y_i) = U - \overline{R}_A(Y_i)$$

Equivalence Set: Equivalence set is a set of attributes whose elements are related to each other by an equivalence relation. Two objects are in the same class if they have the same value for attribute in S.

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:

$$POS_A(Y) = \bigcup_{i=1}^{k} POS_A(Y_i)$$
$$BND(Y) = \bigcup_{i=1}^{k} BND_A(Y_i)$$
$$NEG_A(Y) = \cup - \bigcup_{i=1}^{k} \overline{R}_A(Y_i)$$

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 $X_i \Rightarrow Y_j$). The interpretation of the rules in words is as follows: If the attributes in A have the values given by the elementary set X_i, then the class is the class of the concept Y_j. For any $X_i \in BND_A(Y_j)$ we can generate a <100% confidence rule of the form: If X_i then Y_j (or $X_i \Rightarrow Y_j$), with confidence given as:

$$conf = \frac{\left|X_i \cap Y_j\right|}{X_i}$$

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)

$$\lambda_A(class) = \frac{POS_A(class)}{\left|U\right|}$$

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

$$\beta_A(class) = \frac{\left|U - BND_A(class)\right|}{\left|U\right|} = \frac{\left|POS_A(class) \cup NEG_A(class)\right|}{\left|U\right|}$$

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

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

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

III.2 Rough Set Based Attribute Reduction

III.2.1 Literature overview

The benefits of feature selection are twofold: it considerably decreased the computation time of the induction algorithm and increased the accuracy of the resulting mode [31]. 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 [11] each feature is given a relevance weighting that reflects its ability to discern between decision class labels. Focus [27], 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 [17]. SCRAP [28] is an instance based filter, which determines feature relevance by performing a sequential search within the instance space. EBR [7] based on the entropy heuristic employed by machine learning techniques such as C4.5. EBR is concerned with examining a dataset and determining those attributes that provide the most gain in information. FDR [29] is a novel approach to feature selection based on the concept of fractals – the self-similarity exhibited by data on different scales.

In the wrapper approach [10], 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 [18] is a wrapper method based on LVF algorithm. 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 [29] is employed for backward elimination in the search for optimal subsets.

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 [15]. 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 [20]. 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 [18]. 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. N. Zhong and A. Skowron [32] have applied Rough Sets with Heuristics (RSH) and Rough Sets with Boolean Reasoning (RSBR) for attribute selection and discretization of real-valued attributes.

III.2.2 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 [15]. 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. Feature selection has been studied intensively for the past one decade [8, 10, 11, 16, 21]. This approach provides a mathematical tool that can be used to find out all possible reducts. However, this process is NP-hard [9, 13], if the number of elements of the universe of discourse is large. The Rough Set Attribute Reduction (RSAR) has as central concept the indiscernibility [31]. 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:

$$a:\ U \rightarrow V_a$$

$\forall\ a \in A, V_a$ 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 $IND(P)$:

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

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 ($0 \le k \le 1$), denoted $P \Rightarrow_k Q$, if:

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|}, \quad where \begin{cases} \text{Q depends totally on P} & \text{if } k = 1 \\ \text{Q depends partially on P} & \text{if } 0 < k < 1 \\ \text{Q does not depend on P} & \text{if } k = 0 \end{cases}$$

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:

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

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:

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

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 $R_{\min} \subseteq R$:

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

The most basic solution to locating such a subset is to simply generate all possible subsets and retrieve those with a maximum rough set 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 [16], 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 [31]. 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 $\gamma_R(D) = card(POS_R(D))/card(U)$
$T \leftarrow R \cup \{x\}$
$R \leftarrow T$
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)* [19].

IV. SOLVING APPROACH FOR THE DYNAMIC ROUGH SETS ATTRIBUTE REDUCTION (DRSAR)

IV.1 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 [34]. 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.
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 assignment of Reduct Attributes to build the minimal subsets of attributes by the proposed scheme 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 $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \cdots \tilde{x}_j)$ 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 $\hat{x} = (\hat{x}_1, \hat{x}_2, \cdots \hat{x}_J)$ that starts from the initial state " e_0 " brings us to the state " e_J " while maximizing the following function:

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

The principle of optimality of dynamic programming [39], 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:

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

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 Dynamic Programming, 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 Dynamic Programming can be implemented directly

VI.2 Complexity

The algorithm based on the pattern resolution by the Dynamic Programming consists of three key parameters to evaluate its performance [35]. 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.

VI.2.1 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:

$$\sim O\left(I \times J^2\right)$$

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

$$\sim O\left(I^3 \times J^2\right)$$

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

$$\sim O\left(I^3 \times J^3\right)$$

VI.2.2 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:

$$\sim O\left(I \times J\right)$$

V. CASE STUDY: IMPLEMENTATION AND RESULTS

V.1 Numerical case

The proposed solution strategy has been adapted to a large Retailer Business. We consider the case of an international retailer having many stores with a daily average of 3000 transactions by store. 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 in order to get the new optimal reduced sets of attributes which leads 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. 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.

In order to test our solution approach, we have considered three problem cases that may interest the 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 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 # of transactions in which an item is sold in single, etc. These new calculated attributes have distinct importance relative to the experts.

V.2 Performance evaluation

This section describes some characteristics of tests conducted using the solution approach adopted in order to generate dynamically the different optimal RSAR. We proceed to evaluate the performance of the proposed approach by analyzing the responding time and some various sensitivity analyses that can be conducted through the use of some metrics measure (accuracy, precision, recall) to evaluate the performance of the prediction quality (table 2).

The DRSAR solution method has been developed using the Visual C++ on a PC computer equipped with a Pentium-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 Artificial 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. Concerning the classification process, several experiments were realized in order to test and compare the classification algorithm for each case based on RS technique applied on the initial number of attributes defined by experts and on the dynamic RSAR after running the DRSAR solution approach. The results shown in the below tables (2, 3, 4) focuses on the result before applying the DRSAR and after applying it. It presents the number of records for each case, the initial number of attributes, the reduced number of attributes retained when applying DRSAR, and finally the number of optimal reduced subsets obtained. We report also some metrics measure (accuracy, precision, and recall) to evaluation the quality of the predictive model. We show that the number of attributes is dramatically reduced. Also, the classification quality is very slightly assigned when applying the DRSAR solution approach. So, it is clear that our approach is efficient while the 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.

VI. 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. 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.

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

Table 1: Comparing the related features by using DRSAR

Cases	*Initial number of concept attributes*	*# of optimal subsets of attributes*	*Minimum Reduced attributes*	*Computing time (sec.)*
A-Customers classification	*28*	*2*	*19*	*1.65*
B-Items classification	*52*	*3*	*41*	*8.81*
C-Applying discount on item	*83*	*5*	*68*	*32.35*

Table 2: Confusion matrix results for Customers classification before and after applying DRSAR

# records:	*417.200*	***# Initial set of attributes***	*28*	***# of attributes in the reduced DRSAR***	*19*	***# RSAR***	*2*

Count		***Predicted class***	
		Solvent	*Insolvent*
Actual class	*Solvent*	*319535*	*7705*
	Insolvent	*8650*	*81310*

Count		***Predicted class***	
		Solvent	*Insolvent*
Actual class	*Solvent*	*318675 (99.73%)*	*8920 (86.38%)*
	Insolvent	*7595 (88.19%)*	*82010 (99.75%)*

Accuracy	*96.03*	***Error rate***	*0.92%*
Precision	*97.75*	***Recall***	*97.17*

Table 3: Confusion matrix results for Items classification before and after applying DRSAR

# records:	*933.820*	***# Initial set of attributes***	*52*	***# of attributes in the reduced DRSAR***	*41*	***# RSAR***	*3*

Count		***Predicted class***	
		Attractive	*Non-Attractive*
Actual class	*Attractive*	*822430*	*3145*
	Non-Attractive	*6725*	*101520*

Count		***Predicted class***	
		Attractive	*Non-Attractive*
Actual class	*Attractive*	*822217 (99.74%)*	*3133 (99.62%)*
	Non-Attractive	*8740 (77.05%)*	*99730 (98.73%)*

Accuracy	*98.72*	***Error rate***	*0.43%*
Precision	*98.95*	***Recall***	*99.62*

Table 4: Confusion matrix results for Applying Discount on item before and after applying DRSAR

# records:	*933.820*	***# Initial set of attributes***	*83*	***# of attributes in the reduced DRSAR***	*68*	***# RSAR***	*5*

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*

REFERENCES

[1] Ian H. Witten, Eibe F.: Data mining: Practical machine learning tools and techniques, 2005.
[2] Dorian Pyle, Morgan Kaufmann Publishers, Business Modeling and Data Mining, 2003.
[3] Micheal J.A. Berry, Gordon S. Linoff: Data mining techniques for marketing, sales and customer relationship management, 2004.
[4] Awad, M., Latifur., K.: Design and Implementation of data mining tools, 2009.
[5] Ezawa, K.J., Norton, S.W.: Constructing Bayesian networks to predict uncollectible telecommunications accounts, IEEE Volume 11, Issue 5, Oct. 1996
[6] Carlo Vercellis: Business Intelligence: Data Mining and Optimization for Decision Making, 2009.

[7] Jensen, R., Shen, Q.: A Rough Set – Aided system for Sorting WWW Bookmarks, Web Intelligence: Research and Development, 95-105, 2001
[8] Jensen, R., Shen, Q.: Fuzzy-rough attribute reduction with application to web categorization. Fuzzy Sets and Systems, 141(3): 469-485, 2004.
[9] Pawlak, Z.:Rough Sets:Theoretical aspects of reasoning about data.Kluwer Acad. Pub., 1991.
[10] John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. Proc. of 11th Intl. Conf. on Machine Learning, 121-129, 1994.
[11] Kira, K., Rendell, L.A. : The Feature selection Problem: Traditional Methods and a New Algorithm. Proceedings of AAAI, MIT Press, 129-134, 1992.
[12] Kohavi, R.: Useful feature subsets and Rough set reducts, Proceedings of the 3rd Intl. Workshop on Rough Set and Soft Computing, 310-317, 1994.
[13] Untsch, I. D., Gediga, G., Nguyen, H.S.: Rough set data anaylsis in the KDD process, 1999.
[14] Kusiak, A.: Rough Set Theory:A Datamining Tool for Semiconductor Manufacturing. IEEE Trans. on Electronics Packaging Manufacturing, 24(1), 2001.
[15] Shen, Q., Chouchoulas, A: A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems. Eng. Applications of Artificial Intelligence, 13(3):263-278, 2000.
[16] Liu, H., Motoda, H.: Feature Extraction Construction and Selection: A Data mining Perspective, Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, 1998.
[17] Liu, H., Setiono, R.: A probabilistic approach to feature selection: a filter solution. Proceedings of the 9th International conference on Industrial and Eng. Applications of AI and ES, 284-292, 1996.
[18] Liu, H., Setiono, R.: Feature selection and classification–A probabilistic wrapper approach. Proceedings of the 9th Intl. Conf. on Indust. and Eng. Applications of AI and ES, 419-424, 1996.
[19] Zhong, N., Dong, J., Ohsuga, S.: Using Rough Sets with Heuristics for Feature Selection. Journal of Intelligent Information systems, 16:199-214, 2001.
[20] Hu, X., Lin, T.Y., Jianchao, J.: A New Rough Sets Model Based on Database Systems. Fundamental Informatica, 1-18, 2004.
[21] Modrzejewski, M.: Feature Selection Using Rough Sets Theory. Proceedings of the 11th International Conference on Machine Learning, 213-226, 1993.
[22] Orlowska, E.: Incomplete Information: Rough Set Analysis. Physica-Verlag, Heidelberg, 1998.
[23] Pawlak, Z.: Rough Sets. Intl. Journal of Computer and Information Sciences, 11(5): 341- 356, 1982.
[24] Pawlak, Z.: Rough Sets:Theoritical Aspects and Reasoning about Data. Kluwer Academic Pub., 1991.
[25] Peters, J. F., Skowron, A.: Transactions on Rough Sets 1, Springer-Verlag, Berlin, 2004.
[26] Polkowski, L., Tsumoto, S., Lin, T.Y.: Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Physica-Verlag, Heidelberg, 2000.
[27] Almuallim, H., Dietterich, T.G.: Learning with many irrelevant features. 9th National Conference on Artificial Intelligence, MIT Press, 547-552, 1991.
[28] Raman, B., Loerger, T.R.: Instance-based filter for feature selection. Journal of Machine Learning Research, l1-23, 2002.
[29] Traina, C., Wu, L., Faloutsos, C.: Fast Feature selection using the fractal dimension. Proc. of the 15th Brazilian Symp. on DB (SBBD), P.158, 2000.
[30] Yang, Y., Chiam, T. C.: Rule Discovery Based On Rough Set Theory. Proc. of ISIF, TuC4, 11-16, 2000.
[31] Thangavel, K., Shen, Q., Pethalakshmi, A.: Application of Clustering for Feature selection based on rough set theory approach, AIML Journal, Volume 6, Issue 1, January, 2006, pp. 19-27
[32] Zhong, N., Skowron, A.: A Rough Set-Based Knowledge Discovery Process. Intl. Journal of App. Mathematics and Comp. Sc., 11(3):603-619, 2001.
[33] Almuallim, A., Dietterich, T.G.: Learning with many irrelevant features. 9th National Conference on Artificial Intelligence, MIT Press, 547-552, 1991.
[34] Bellman, R.E.: Dynamic Programming. Princeton University Press, 1957.
[35] EL Moudani, W., Mora-Camino, F.: A Dynamic Approach for Aircraft Assignment and Maintenance Scheduling by Airlines. Journal of Air Transport Management, Vol. 4, N° 1, pp. 233-237, 2000.