

Medical Domain Based Feature Selection Using Rough Set Reduct Algorithm

T. Keerthika,

Assistant professor, Department of Information
Technology,
Sri Krishna College of Engineering and Technology.

Dr. K. Premalatha,

Professor, Department of Computer Science and
Engineering,
Bannari Amman Institute of Technology,
Sathyamangalam, Erode.

Abstract—The real time data will dynamically increase in size. To achieve it effectively and efficiently an incremental technique has to be proposed, which stimulates the result. Feature selection refers to the problem of selecting those input features that are most predictive of a given outcome. In particular, this has found successful application in tasks that involve datasets containing huge numbers of features, which would be impossible to process further. Recent examples include text processing and web content classification. Rough set theory has been used as such a dataset preprocessor with much success, but current methods are inadequate at finding *minimal* reductions.

Index Terms—Dynamic data sets, incremental algorithm, feature selection, rough set theory

I. INTRODUCTION

The problem of reducing dimensionality has been investigated for a long time in a wide range of fields, e.g., statistics, pattern recognition, machine learning, and knowledge discovery. In order to reduce the input dimensionality, there exist two main approaches, i.e., feature extraction and feature selection (FS). Feature extraction maps the primitive feature space into a new space with a lower dimensionality. Two of the most popular feature extraction approaches include Principal Components Analysis, and Partial Least Squares. There are numerous applications of feature extraction in the literature, such as image processing, visualization, and signal processing. In contrast, the FS approach chooses the most informative features from the original features according to a selection method, e.g., t-statistic, f-statistic, correlation, reparability correlation measure, or information gain. The irrelevant and redundant features in the dataset lead to slow learning and low accuracy. Finding the subset of features that are enough informative is NP complete. Some heuristic algorithms are proposed to search through the feature space. The selected subset can be evaluated from some issues, such as the complexity of the learning algorithm and the accuracy.

The Rough Set (RS) theory can be used as a tool to reduce the input dimensionality and to deal with vagueness and uncertainty in datasets. The reduction of attributes is based on data dependencies. The RS theory partitions a dataset into some equivalent (indiscernibility) classes, and approximates uncertain and vague concepts based on the partitions. The measure of dependency is calculated by a function of the approximations. The dependency measure is employed as a heuristic to guide the FS process. In order to

obtain a significant measure, proper approximations of the concepts are required. Hence, the initial partitions play an important rule. Given a discrete dataset, it is possible to find the indiscernibility classes; however, in case of datasets with real-valued attributes, it is impossible to say whether two objects are the same, or to what extent they are the same, using the indiscernibility relation. A number of research groups extended the RS theory using the tolerant or similarity relation (termed tolerance-based Rough Set).

The similarity measure between two objects is delineated by a distance function of all attributes. Two objects are considered to be similar when their similarity measure exceeds a similarity threshold value. Finding the best threshold boundary is both important and challenging. Used genetic algorithms to find the best similarity threshold. Used fuzzy similarity to cope with real-valued attributes.

1.1 ROUGH SET FOR FEATURE SELECTION

Rough set theory is a new mathematical approach to imprecision, vagueness and uncertainty. In an information system, every object of the universe is associated with some information. Objects characterized by the same information are indiscernible with respect to the available information about them. Any set of indiscernible objects is called an elementary set. Any union of elementary sets is referred to as a crisp set- otherwise a set is rough (imprecise, vague). Vague concepts cannot be characterized in terms of information about their elements. A rough set is the approximation of a vague concept by a pair of precise concepts, called lower and upper approximations. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset. Relative to a given set of attributes, a set is rough if its lower and upper approximations are not equal.

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. A *reduct* is 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, so the set R of all reducts is defined as:

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

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 $R_{min} \subseteq R$.

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

The reduct and minimal reduct sets for the example are:

$$R = \{\{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{b, d\}, \{c, d\}\}$$

$$R_{min} = \{\{b, d\}, \{c, d\}\}$$

The problem of finding a minimal reduct of an information system has been the subject of much research. The most basic solution to locating such a reduct is to simply generate all possible reducts and choose any with minimal cardinality. Obviously, this is an expensive solution to the problem and is only practical for very simple datasets. Most of the time only one minimal reduct is required, so all the calculations involved in discovering the rest are pointless. To improve the performance of the above method, an element of pruning can be introduced. By noting the cardinality of any prediscovered reducts, the current possible reduct can be ignored if it contains more elements. However, a better approach is needed – that one will avoid wasted computational effort.

Algorithm attempts to calculate a 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 the rough set dependency metric, until this produces its maximum possible value for the dataset. According to the algorithm, the mean dependency of each attribute subset is calculated and the best candidate is chosen:

$$\gamma_p(a) = \frac{|\text{POS}_p(a)|}{|U|}, \quad \forall a \in A.$$

1.2 PSO FOR FEATURE SELECTION

Particle swarm optimization (PSO) is an evolutionary computation technique the original intent was to graphically simulate the graceful but unpredictable movements of a flock of birds. Initial simulations were modified to form the original version of PSO. Later, Shi introduced inertia weight into the particle swarm optimizer to produce the standard PSO. PSO is initialized with a population of random solutions, called ‘particles’. Each particle is treated as a point in an S-dimensional space. The *i*th particle is represented as $x_i = (x_{i1}, x_{i2}, x_{i3} \dots \dots x_{iS})$. The best previous position (pbest, the position giving the best fitness $p_i = (p_{i1}, p_{i2}, \dots, p_{iS})$. The index of the best particle among all the particles in the population is represented by the symbol ‘gbest’. The rate of the position change (velocity) for particle *i* is represented as $V_i = (v_{i1}, v_{i2}, \dots, v_{iS})$. The particles are manipulated according to the following equation: value) of any particle is recorded and represented

$$v_{id} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * Rand() * (p_{gd} - x_{id})$$

$$x_{id} = x_{id} + v_{id}$$

Where $d = 1, 2, \dots, S$, w is the inertia weight, it is a positive linear function of time changing according to the generation iteration. Suitable selection of the inertia weight provides a balance between global and local exploration, and results in less iteration on average to find a sufficiently optimal solution. The acceleration constants c_1 and c_2 in equation represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest positions. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement toward, or past, target regions. $rand()$ and $Rand()$ are two random functions in the range.

Particle’s velocities on each dimension are limited to a maximum velocity, V_{max} . It determines how large steps through the solution space each particle is allowed to take. If V_{max} is too small, particles may not explore sufficiently beyond locally good regions. They could become trapped in local optima. On the other hand, if V_{max} is too high particles might fly past good solutions.

The first part of equation provides the “flying particles” with a degree of memory capability allowing the exploration of new search space areas. The second part is the “cognition” part, which represents the private thinking of the particle itself. The third part is the “social” part, which represents the collaboration among the particles. PSO is used to calculate the particle’s new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group’s best experience. Then the particle flies toward a new position according to equation.

1.3 ANT COLONY OPTIMIZATION FOR FEATURE SELECTION

Swarm Intelligence (SI) is the property of a system whereby the collective behaviors of simple agents interacting locally with their environment cause coherent functional global patterns to emerge. SI provides a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model. One area of interest in SI is Particle Swarm Optimization, a population-based stochastic optimization technique. Here, the system is initialized with a population of random solutions, called particles. Optima are searched for by updating generations, with particles moving through the parameter space towards the current local and global optimum particles. At each time step, the velocities of all particles are changed depending on the current optima.

Ant Colony Optimization (ACO) is another area of interest within SI. In nature, it can be observed that real ants are capable of finding the shortest route between a food source and their nest without the use of visual information and hence possess no global world model, adapting to changes in the environment. The deposition of pheromone is the main factor in enabling real ants to find the shortest routes over a period of time. Each ant probabilistically prefers to follow a direction rich in this chemical. The pheromone decays over time, resulting in much less pheromone on less popular paths. Given that over time the shortest route will have the higher rate of ant traversal, this path will be reinforced and the others

diminished until all ants follow the same, shortest path (the "system" has converged to a single solution). It is also possible that there are many equally short paths. In this situation, the rates of ant traversal over the short paths will be roughly the same, resulting in these paths being maintained while others are ignored. Additionally, if a sudden change to the environment occurs (e.g. a large obstacle appears on the shortest path), the ACO system can respond to this and will eventually converge to a new solution. Based on this idea, artificial ants can be deployed to solve complex optimization problems via the use of artificial pheromone deposition.

ACO is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time. Additionally, it can be the case that ants discover the best feature combinations as they proceed throughout the search space. This section discusses how ACO may be applied to the difficult problem of finding optimal feature subsets and, in particular, fuzzy-rough set-based reducts.

The feature selection task may be reformulated into an ACO-suitable problem. ACO requires a problem to be represented as a graph here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Illustrates this setup - the ant is currently at node a and has a choice of which feature to add next to its path (dotted lines). It chooses feature b next based on the transition rule, then c and then d . Upon arrival at d , the current subset $fa; b; c; dg$ is determined to satisfy the traversal stopping criteria (e.g. a suitably high classification accuracy has been achieved with this subset, assuming that the selected features are used to classify certain objects). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction.

1.4 GENETIC ALGORITHM FOR FEATURE SELECTION

Genetic algorithm (GA) is a search heuristic, used to generate solutions to optimization problems following the techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In the genetic algorithm,

- ❖ A population of strings (called chromosomes), which encode candidate solution to an optimization problem is taken.
- ❖ A proper fitness function is constructed, and fitness of the current population is evaluated.
- ❖ Two fittest chromosomes are chosen as the parents and (a) crossing over between them or (b) mutation of a parent is performed to produce new children and a new population.
- ❖ Again the fitness function for the new population is estimated.
- ❖ The process recurs as long as the fitness function keeps on improving or termination condition is attained.

The algorithm of a genetic programming begins with a population that is a set of randomly created individuals. Each

individual represents a potential solution that is represented as a binary tree. Each binary tree is constructed by all possible compositions of the sets of functions and terminals. A fitness value of each tree is calculated by a suitable fitness function. According to the fitness value, a set of individuals having better fitness will be selected. These individuals are used to generate new population in next generation with genetic operators. Genetic operators generally also include reproduction, crossover, mutation and others that are used to evolve functional expressions. After the evolution of a number of generations, we can obtain an individual with good fitness value. If the fitness value of such individual still does not satisfy the specified conditions of the solution, the process of evolution will be repeated until the specified conditions are satisfied.

II. LITERATURE REVIEW

2.1. A Rough-Set Based Incremental Approach for Updating Approximations under Dynamic Maintenance Environments

Approximations of a concept by a variable precision rough-set model (VPRS) usually vary under a dynamic information system environment. It is thus effective to carry out incremental updating approximations by utilizing previous data structures. This paper focuses on a new incremental method for updating approximations of VPRS while objects in the information system dynamically alter. It discusses properties of information granulation and approximations under the dynamic environment while objects in the universe evolve over time. The variation of an attributes domain is also considered to perform incremental updating for approximations under VPRS. Finally, an extensive experimental evaluation validates the efficiency of the proposed method for dynamic maintenance of VPRS approximations.

2.2. A Novel Dynamic Incremental Rules Extraction Algorithm Based on Rough Set Theory

The incremental rules extraction is a focus problem of KDD. In this paper, a novel incremental rules extraction algorithm which is called "RDBRST" (Rule Derivation Based on Rough Set And Search Tree) is proposed. It is one kind of width first heuristic search algorithms. The incremental rules are extracted and the existing rule set is updated based on this algorithm. We present an example to illustrate characteristics of this new incremental algorithm.

2.3. Incremental Induction of Decision Rules from Dominance-Based Rough Approximations

It is extended to handle preference-ordered domains of attributes (called criteria) within Variable Consistency Dominance-based Rough Set Approach. It deals, moreover, with the problem of missing values in the data set. The algorithm has been designed for medical applications which require: (i) a careful selection of the set of decision rules representing medical experience and (ii) an easy update of these decision rules because of data set evolving in time, and (iii) not only a high predictive capacity of the set of decision

rules but also a thorough explanation of a proposed decision. To satisfy all these requirements, we propose an incremental algorithm for induction of a satisfactory set of decision rules and a post-processing technique on the generated set of rules.

2.4. A Distance Measure Approach to Exploring the Rough Set Boundary Region for Attribute Reduction

This paper examines a rough set FS technique which uses the information gathered from both the lower approximation dependency value and a distance metric which considers the number of objects in the boundary region and the distance of those objects from the lower approximation. The use of this measure in rough set feature selection can result in smaller subset sizes than those obtained using the dependency function alone. This demonstrates that there is much valuable information to be extracted from the boundary region.

2.5. Incremental Learning of Decision Rules Based on Rough Set Theory

In this paper, based on the rough set theory, the concept of ∂ -indiscernibility relation is put forward in order to transform an inconsistent decision table to one that is consistent, called ∂ -decision table, as an initial preprocessing step. Then, the ∂ -decision matrix is constructed. On the basis of this, by means of a decision function, an algorithm for incremental learning of rules is presented. The algorithm can also incrementally modify some numerical measures of a rule.

III. FEATURE SELECTION

The main aim of feature selection (FS) is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features. In real world problems FS is a must due to the abundance of noisy, irrelevant or misleading features. For instance, by removing these factors, learning from data techniques can benefit greatly. Given a feature set size n , the task of FS can be seen as a search for an "optimal" feature subset through the competing $2n$ candidate subsets. The definition of what an optimal subset is may vary depending on the problem to be solved. Although an exhaustive method may be used for this purpose, this is quite impractical for most datasets.

Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimality of the final feature subset is often reduced.

The usefulness of a feature or feature subset is determined by both its *relevancy* and *redundancy*. A feature is said to be relevant if it is predictive of the decision feature(s), otherwise it is irrelevant. A feature is considered to be redundant if it is highly correlated with other features. Hence, the search for a good feature subset involves finding those features that are highly correlated with the decision feature(s), but are uncorrelated with each other.

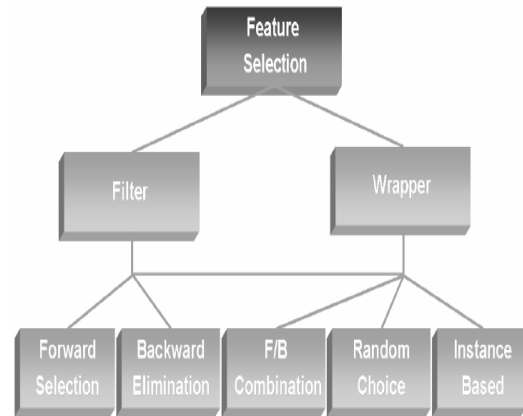


Figure 1.1 Aspects of feature selection

Determining subset optimality is a challenging problem. There is always a trade-off in non-exhaustive techniques between subset minimality and subset suitability - the task is to decide which of these must suffer in order to benefit the other. For some domains (particularly where it is costly or impractical to monitor many features), it is much more desirable to have a smaller, less accurate feature subset. In other areas it may be the case that the modeling accuracy (e.g. the classification rate) using the selected features must be extremely high, at the expense of a non-minimal set of features.

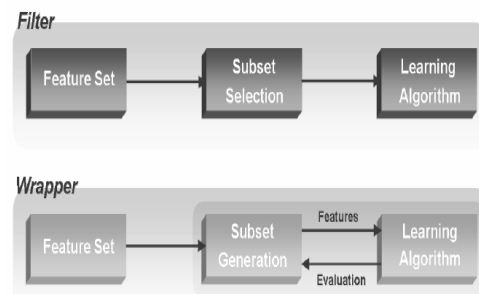


Figure 1.2 Filter and wrapper methods

Feature selection algorithms may be classified into two categories based on their evaluation procedure (see Figure 1.2). If an algorithm performs FS independently of any learning algorithm (i.e. it is a completely separate preprocessor), then it is a *filter* approach. In effect, irrelevant attributes are filtered out before induction. Filters tend to be applicable to most domains as they are not tied to any particular induction algorithm.

If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the FS algorithm employs the *wrapper* approach. This method searches through the feature subset space using the estimated accuracy from an induction algorithm as a measure of subset suitability. Although wrappers may produce better results, they are expensive to run and can break down with very large numbers of features. This is due to the use of learning algorithms in the evaluation of subsets, some of which can encounter problems when dealing with large datasets.

IV. ROUGH SET-BASED FEATURE SELECTION

Rough set theory (RST) can be used as a tool to discover data dependencies and to reduce the number of attributes contained in a dataset using the data alone, requiring no additional information. Over the past ten years, RST has become a topic of great interest to researchers and has been applied to many domains. Given a dataset with discretized attribute values, it is possible to find a subset (termed a *reduct*) of the original attributes using RST that are the most informative; all other attributes can be removed from the dataset with minimal information loss. From the dimensionality reduction perspective, informative features are those that are most predictive of the class attribute.

There are two main approaches to finding rough set reducts: those that consider the degree of dependency and those that are concerned with the discernibility matrix. This section describes the fundamental ideas behind both approaches. To illustrate the operation of these, an example dataset (Table 1.1) will be used.

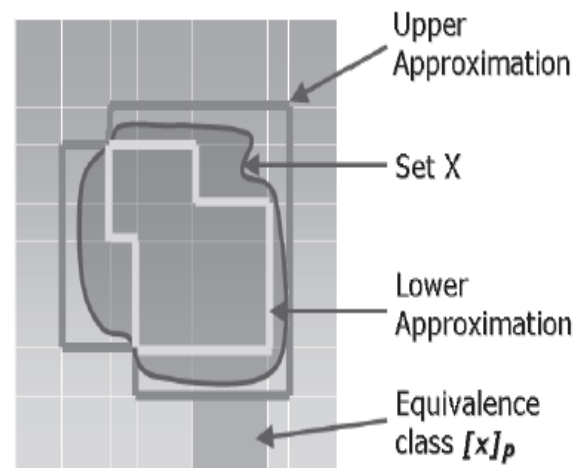


Figure 1.3 A Rough Set

Table 1.1 An example dataset

xU	A	b	c	d	e
0	1	0	2	2	0
1	0	1	1	1	2
2	2	0	0	1	1
3	1	1	0	2	2
4	1	0	2	0	1
5	2	2	0	1	1
6	2	1	1	1	2
7	0	1	1	0	1

a) Rough Set Attribute Reduction

Central to Rough Set Attribute Reduction (RSAR) is the concept of indiscernibility. Let $I = (U, A)$ be an information system, where U is a non-empty set of finite objects (the universe) and A is a non-empty finite set of attributes such that $a:U \rightarrow V_a$ for every $a \in A$. V_a is the set of values that attribute a may take. With any $P \in A$ there is an associated equivalence relation $IND(P)$:

b) Information and Decision Systems

An information system can be viewed as a table of data, consisting of objects (rows in the table) and attributes (columns). In medical datasets, for example, patients might be represented as objects and measurements such as blood pressure, form attributes. The attribute value for a particular patient is their specific reading for that measurement. Throughout this paper, the terms attribute, feature and variable are used interchangeably.

An information system may be extended by the inclusion of decision attributes. Such a system is termed a decision system. For example, the medical information system mentioned previously could be extended to include patient classification information, such as whether a patient is ill or healthy. A more abstract example of a decision system can be found in table 1. Here, the table consists of four conditional features (a; b; c; d), a decision feature (e) and eight objects. A decision system is consistent if for every set of objects whose attribute values are the same, the corresponding decision attributes are identical.

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

The partition of U generated by $IND(P)$ is denoted $U/IND(P)$ (or U/P). If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P . The equivalence classes of the P -indiscernibility relation are denoted $[x]P$. For the illustrative example, if $P = \{b, c\}$, then objects 1, 6 and 7 are indiscernible; as are objects 0 and 4. $IND(P)$ creates the following partition of U :

$$\begin{aligned} \frac{U}{IND(P)} &= U/IND(\{b\}) \otimes U/IND(\{c\}) \\ &= \{\{0,2,4\}, \{1,3,6,7\}, \{5\}\} \otimes \{\{2,3,5\}, \{1,6,7\}, \{0,4\}\} = \{\{2\}, \{0,4\}, \{3\}, \{1,6,7\}, \{5\}\} \end{aligned}$$

c) Lower and Upper Approximations

Let $X \subseteq U$. X can be approximated using only the information contained within P by constructing the P-lower and P-upper approximations of X :

$$\underline{P}X = \{x \mid [x]_P \subseteq X\}$$

$$\overline{P}X = \{x \mid [x]_P \cap X \neq \emptyset\}$$

d) Positive, Negative and Boundary Regions

Let P and Q be equivalence relations over U , then the positive region can be defined as:

$$POS_P(Q) = \bigcup_{x \in U/Q} \underline{P}x$$

The positive region contains all objects of U that can be classified to classes of U/Q using the information in attributes P . For example, let $P = \{b, c\}$ and $Q = \{e\}$, then

$$POS_P(Q) = U \setminus \{\emptyset, \{2,5\}, \{3\}\} = \{\emptyset, \{2,3,5\}\}$$

Using this definition of the positive region, the rough set degree of dependency of a set of attributes Q on a set of attributes P is defined in the following way:

For $P, Q \subseteq A$, it is said that Q depends on P in a degree k ($0 \leq k \leq 1$), denoted $P \rightarrow_k Q$, if

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

In the example, the degree of dependency of attribute $\{e\}$ from the attributes $\{b, c\}$ is:

$$\begin{aligned} \gamma_{\{b,c\}}(\{e\}) &= \frac{|POS_{\{b,c\}}(\{e\})|}{|U|} \\ &= \frac{|\{\emptyset, \{2,3,5\}\}|}{|\{0,1,2,3,4,5,6,7\}|} = \frac{3}{8} \end{aligned}$$

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 predictive capability of the decision feature as the original. A reduct R is defined as a subset of minimal cardinality of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$.

V. CONCLUSIONS

Feature Selection is an important research direction of rough set application. However, this technique often fails to find better reducts. This project starts with the fundamental concepts of rough set theory and explains basic techniques: Quick Reduct. These methods can produce close to the minimal reduct set. The swarm intelligence methods have been used to guide this method to find the minimal reducts. Here three different computational intelligence based reducts are used: Genetic algorithm, Ant colony optimization and PSO. Though these methods are performing well, there is no consistency since they are dealing with more random parameters. All these methods are analyzed using medical datasets. As shown in the results, our proposed method exhibits consistent and better performance than the other methods.

REFERENCES

- [1] H.M. Chen, T.R. Li, D. Ruan, J.H. Lin, and C.X. Hu, "A Rough-Set Based Incremental Approach for Updating Approximations under Dynamic Maintenance Environments," IEEE Trans. Knowledge and Data Eng., vol. 25, no. 2, pp. 274-284, Feb. 2013.
- [2] J.Y. Liang, F. Wang, C.Y. Dang, and Y.H. Qian, "An Efficient Rough Feature Selection Algorithm with a Multi-Granulation View," Int'l J. Approximate Reasoning, vol. 53, pp. 912-926, 2012.
- [3] J.F. Pang and J.Y. Liang, "Evaluation of the Results of Multi- Attribute Group Decision-Making with Linguistic Information," Omega, vol. 40, pp. 294-301, 2012.
- [4] Q.H. Hu, D.R. Yu, W. Pedrycz, and D.G. Chen, "Kernelized Fuzzy Rough Sets and Their Applications," IEEE Trans. Knowledge and Data Eng., vol. 23, no. 11, pp. 1649-1667, Nov. 2011.
- [5] N. Parthalaïn, Q. Shen, and R. Jensen, "A Distance Measure Approach to Exploring the Rough Set Boundary Region for Attribute Reduction," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 3, pp. 305-317, Mar. 2010.
- [6] Y.H. Qian, J.Y. Liang, W. Pedrycz, and C.Y. Dang, "Positive Approximation: An Accelerator for Attribute Reduction in Rough Set Theory," Artificial Intelligence, vol. 174, pp. 597-618, 2010.
- [7] W. Wei, J.Y. Liang, Y.H. Qian, F. Wang, and C.Y. Dang, "Comparative Study of Decision Performance of Decision Tables Induced by Attribute Reductions," Int'l J. General Systems, vol. 39, no. 8, pp. 813-838, 2010.
- [8] S.Y. Zhao, E.C.C. Tsang, D.G. Chen, and X.Z. Wang, "Building a Rule-Based Classifier-a Fuzzy-Rough Set Approach," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 5, pp. 624-638, May 2010.
- [9] M. Kryszkiewicz and P. Lasek, "FUN: Fast Discovery of Minimal Sets of Attributes Functionally Determining a Decision Attribute," Trans. Rough Sets, vol. 9, pp. 76-95, 2008.
- [10] M.Z. Li, B. Yu, O. Rana, and Z.D. Wang, "Grid Service Discovery with Rough Sets," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 6, pp. 851-862, June 2008.