ResearchGate

Game-Theoretic Rough Sets

Article *in* Fundamenta Informaticae · January 2011

DOI: 10.3233/FI-2011-423 · Source: DBLP

2 authors:

Joseph Herbert

Jingtao Yao

University of Regina

Game-Theoretic Rough Sets

Joseph P. Herbert
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada
herbertj@cs.uregina.ca

JingTao Yao
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada
jtyao@cs.uregina.ca

Abstract. This article investigates the Game-theoretic Rough Set (GTRS) model and its capability of analyzing a major decision problem evident in existing probabilistic rough set models. A major challenge in the application of probabilistic rough set models is their inability to formulate a method of decreasing the size of the boundary region through further explorations of the data. To decrease the size of this region, objects must be moved to either the positive or negative regions. Game theory allows a solution to this decision problem by having the regions compete or cooperate with each other in order to find which is best fit to be selected for the move. There are two approaches discussed in this article. First, the region parameters that define the minimum conditional probabilities for region inclusion can either compete or cooperate in order to increase their size. The second approach is formulated by having classification approximation measures compete against each other. We formulate a learning method using the GTRS model that repeatedly analyzes payoff tables created from approximation measures and modified conditional risk strategies to calculate parameter values.

Keywords: rough sets, set approximation, game theory, decision-theoretic model, variable precision model, probabilistic rough set model, game-theoretic model

Address for correspondence: Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2

1. Introduction

Probabilistic rough sets [28, 34] extend the original rough set model given by Pawlak in the early 1980s [19]. The major change is the consideration regarding probabilities of objects being in a set to determine inclusion in approximation regions. Two probabilistic thresholds are used to determine the division between the boundary-positive region and boundary-negative region. The decision-theoretic rough set model calculates these thresholds based on minimum risk of a classification action [29, 32]. Minimum risk is derived from a set of loss functions: functions representing a users notion of classification cost. We propose a game-theoretic rough set model that takes into account each models capability for deriving thresholds.

A fundamental challenge exists pertaining to how the probability thresholds may change in order to improve the performance of rough set classification. A decision problem is created when one realizes that we cannot simply change thresholds in an arbitrary manner [9, 25]. Likewise, we cannot change a users notion of risk without considering the ramifications of new risk beliefs will have in the application domain. We must look at the data itself in order to gain insight on improving classification performance. To this end, the Bayesian rough set model [21], parameterized rough set model [7], and the stochastic dominance-based rough set approach [14] all have contributed to this end. In Reference [21], the authors use a certainty gain measure to evaluate the performance of the model when new information is found. In Reference [7], Bayesian confirmation measures are used to determine to what degree an object's membership in an equivalence class gives to the likelihood of an object belonging to a set approximation. In Reference [14], "inconsistent" data, or objects that clearly have preferable attributes values to others but result in less-favorable classifications, are re-evaluated using the dominance relation. The game-theoretic rough set model introduced here allows for the simultaneous consideration of multiple classification approximation measures in order to determine suitable values for loss functions or parameters.

Game theory is a powerful method for mathematically formulating decision problems as competition between two or more entities [18]. These entities, or players, aspire to either achieve a dominant position over the other players or collaborate with each other in order to find a position that benefits all [6]. To formulate a decision problem in game theory, information regarding the players in the game (those with a unique position to directly influence the decision), the actions that can be performed to influence a decision, and the resulting payoff that is achieved signifying that a decision is nearer to being made [10].

In this article, we introduce game-theoretic rough sets. We make the connection that the parameter optimization decision problem is highly similar to that of a competitive game. Since reducing the size of the boundary region is our goal, we can formulate two different types of games with this model.

First, we consider when the two probabilistic thresholds are competing to reach the goal directly by reducing the boundary region. The upper threshold must compete to decrease its value, as a decrease will increase the positive region. The lower threshold must compete to increase its value, as this will increase the negative region. A coalition can be formed between the two thresholds in order to achieve a level of balance in region sizes.

Second, we consider when two classification approximation measures compete for dominance in a game. The classification ability of rough set analysis can considered by observing values for accuracy, precision, generality, etc [4]. In order to increase the classification ability of the system, actions in modifying the conditional risk of classifying an object into a region can be increased or decreased. In a game formulated in this manner, repetitive actions of modifying conditional risk lead to a learning method to derive the thresholds based on a target measure value. In addition, we offer a learning method of deriving loss functions from the relationship between classification ability and risk.

The competition and cooperation of parameters pairs and approximation measure pairs give us insight regarding the impact these simple concepts have on the decision support capability of rough sets. By using game theory, we are able to model a solution to the fundamental decision problem created through the use of probabilistic rough sets. The reduction of the boundary region is, however, restricted the usability of the knowledge gained from the data. Simply reducing the size of the boundary region to the point that is empty does not provide us with any increased certainty. In fact, the knowledge gained from such a case would be less useful as it would be too general to provide any meaningful descriptions.

This article is organized as follows: Section 2 reviews some background information regarding game theory and the two probabilistic models of variable-precision and decision-theoretic rough sets. Section 3 formulates the game-theoretic rough set (GTRS) model designed to aid in the decision problem mentioned earlier. Two approaches to competition are observed: competition between region parameters and competition between classification approximation measures. Section 4 presents an application of the model to time-series data. Section 5 concludes the article with further insights.

2. Background Information

2.1. Game Theory

Game theory [18] has been one of the core subjects of the decision sciences, specializing in the analysis of decision making in an interactive environment. Many applications can be expressed as a game between two or more players. Game theory is as an advanced problem solving technique that can be used in many domains. Economics [17, 20], machine learning [8], networking [1], and cryptography [5] are just some examples of it's use.

The basic assumption of game theory is that all players are considered rational in terms of attempting to maximize position of winning the game. Rational behavior more predictable than irrational behavior, as opposing parties are able to determine other party's strategies on the assumption will not knowingly make their situation worse than before.

In a simple game put into formulation consists of a set of players $O = \{o_1, \ldots, o_n\}$, a set of actions $S = \{a_1, \ldots, a_m\}$ for each player, and the respective payoff functions for each action $F = \{\mu_1, \ldots, \mu_m\}$. Each player chooses actions from S to be performed according to expected payoff from F, usually some a_i maximizing payoff $\mu(a_i)$ while minimizing other player's payoff.

Let us look at a classical example of the use of game theory: the prisoners' dilemma. In this game, there are two players, $O = \{o_1, o_2\}$. Each player has been captured by authorities in regards to a suspected burglary. While being interrogated by the police, the prisoners each have a choice of two actions they can perform: to confess and implicate the other prisoner for the crime, or not to confess to the burglary, i.e. $S = \{a_1, a_2\}$, where $a_1 =$ *confess* and $a_2 =$ *don't confess*. The payoff functions for each action correspond to the amount of jail time that prisoner will receive. These payoffs are expressed in Table 1, called a payoff table.

If prisoner o_1 confesses to the burglary and implicates the other, he will serve either a maximum of 10 years in jail or serve nothing, depending on whether or not prisoner o_2 confesses or not. If o_1 doesn't confess, he will serve at least 1 year in jail or a maximum of 20 years. If both confess, they each serve 10 years. If neither confesses, they both serve 1 year. Clearly, without knowing the other's action, a rational

		o_2	
		confess	don't confess
o_1	confess	o_1 serves 10 years, o_2 serves 10 years,	o_1 serves 0 years, o_2 serves 20 years,
	don't confess	o_1 serves 20 years, o_2 serves 0 years,	o_1 serves 1 year, o_2 serves 1 year,

Table 1. Payoff table for the Prisoners' Dilemma.

player would choose to confess, as it provides the smallest maximum jail term as well as the smallest overall term.

The above example demonstrates one of the strengths of game theory for aiding data analysis. It provides clarity to complex scenarios where multiple actions influence the outcome in a predictable manner. The decision-theoretic rough set approach to data analysis is one such method where classification ability is configurable by observing different values of conditional risk associated with an action. As we will see in the game-theoretic rough set model, we utilize payoff tables similar to that above to organize the strategies involved and for analyzing the results of these strategies for improving classification ability.

2.2. Probabilistic Rough Set Models

The traditional rough set model is often too strict when including objects into the approximation regions and may require additional information [15], or require several approximations [3]. A probabilistic model for rough sets can modify rough set regions in a universe. The first approach that we will look at in-depth, variable-precision, makes use of given probabilities as parameters to produce the regions. The second model, decision-theoretic, uses the cost of classifying an object either correctly or incorrectly to determine the different regions.

2.2.1. Variable-Precision Rough Sets

The variable-precision rough set (VPRS) approach can be used in many areas to support data analysis [13, 35, 36]. The VPRS approach improves on the traditional rough set approach by using bounds on conditional probabilities [34]. Two parameters, the lower-bound ℓ and the upper-bound u, are provided by the user or domain expert.

The parameter u signifies the least degree of conditional probability $P(A|[x])$ to include an object x with description $[x]$ into a set A. This is called the u-positive region,

$$\begin{aligned} POS_u(A) &= \{x \in A | P(A|[x]) \geq u\}, &(1)\\ NEG_\ell(A) &= \{x \in A | P(A|[x]) \leq \ell\}, &(2)\\ BND_{\ell,u}(A) &= \{x \in A | \ell < P(A|[x]) < u\}, &(3)\end{aligned}$$

That is, an object x is included in a set A if the probability, given its description, of it belonging to A is greater-than or equal-to an upper-bound probability specified by the user. The positive region of a set

A consists of all objects that meet this criterion. Likewise, the l-negative region $NEG_\ell(A)$ is controlled by the lower-bound ℓ. An object x is included in a set A if the probability, given its description, of it belonging to A is less-than or equal-to a lower-bound probability specified by the user. The negative region of set A consists of all objects that, again, meet this criterion. The boundary region is now potentially smaller in size since the u-positive and ℓ-negative regions increase the size of the positive and negative regions. The ℓ, u-boundary region classifies all remaining objects.

The quality of classification is user-driven since the ℓ and u parameters are given by the user. An expert provides values for these parameters based on their knowledge or their intuition [30]. An upper-bound u set too low decreases the certainty that any object is correctly classified. Likewise, a lower-bound l that is set too high suffers from the same outcome. The special case $u = 1$ and $l = 0$ results in this approach performing exactly like the Pawlak model.

Similar to VPRS with symmetric bounds in the respect that it uses a single fixed threshold to determine regions, the stochastic dominance-based rough set approach [14] utilizes the dominance relation instead of the indiscernibility relation to describe the relationships between objects. Classes are created through the unions of sets that either dominate (objects having attribute values more favorable that those in a lower class) or are dominated (objects having attribute values less favorable than those in an upper class) themselves.

The parameterized rough set approach given in [7] determines regions through the concurrent use of two pairs of parameters. The lower approximation is defined by those objects with conditional probabilities greater than or equal to that of a threshold t and a resulting confirmation measure greater than or equal to a selected threshold α within the range of that particular measure. Likewise, the upper approximation is defined by those objects with conditional probabilities strictly greater than a threshold q and a confirmation measure value greater than another threshold β, where $t < q$ and $\alpha \geq \beta$.

Even though the probabilities of each individual object are measurable from the data, inclusion of these objects into regions depends on the user's ability to correctly describe the region bounds. The decision-theoretic rough set approach calculates region boundaries based on the cost of a classification action, i.e. the loss incurred by correct and incorrect classifications, and thus, does not have this limitation.

2.2.2. The Decision-Theoretic Rough Set Approach

Using the Bayesian decision procedure, the decision-theoretic rough set (DTRS) approach allows for minimum risk decision making based on observed evidence. Let $\mathcal{A} = \{a_1, \ldots, a_m\}$ be a finite set of m possible actions and let $\Omega = \{w_1, \ldots, w_s\}$ be a finite set of s states. $P(w_j|\mathbf{x})$ is calculated as the conditional probability of an object x being in state w_j given the object description $\mathbf{x}$. $\lambda(a_i|w_j)$ denotes the loss, or cost, for performing action a_i when the state is w_j. The expected loss (conditional risk) associated with taking action a_i is given by [29, 32]:

$$R(a_i|\mathbf{x}) = \sum_{j=1}^{s} \lambda(a_i|w_j)P(w_j|\mathbf{x}). \tag{4}$$

Object classification with the approximation operators can be fitted into the Bayesian decision framework. The set of actions is given by $\mathcal{A} = \{a_P, a_N, a_B\}$, where a_P, a_N, and a_B represent the three actions in classifying an object into POS(A), NEG(A), and BND(A) respectively. To indicate whether an element is in A or not in A, the set of states is given by $\Omega = \{A, A^c\}$. Let $\lambda(a_\diamond|A)$ denote the loss incurred by taking action $a_\diamond$ when an object belongs to A, and let $\lambda(a_\diamond|A^c)$ denote the loss incurred by take the same action when the object belongs to A^c [25].

Let λ_{PP} denote the loss function for classifying an object in A into the POS region, λ_{BP} denote the loss function for classifying an object in A into the BND region, and let λ_{NP} denote the loss function for classifying an object in A into the NEG region. A loss function $\lambda_{\diamond N}$ denotes the loss of classifying an object that does not belong to A into the regions specified by $\diamond$.

Taking individual can be associated with the expected loss $R(a_\diamond|[x])$actions and can be expressed as:

$$\begin{aligned} R(a_P|[x]) &= \lambda_{PP}P(A|[x]) + \lambda_{PN}P(A^c|[x]), \\ R(a_N|[x]) &= \lambda_{NP}P(A|[x]) + \lambda_{NN}P(A^c|[x]), \\ R(a_B|[x]) &= \lambda_{BP}P(A|[x]) + \lambda_{BN}P(A^c|[x]), \end{aligned} \quad (5)$$

where $\lambda_{\diamond P} = \lambda(a_\diamond|A)$, $\lambda_{\diamond N} = \lambda(a_\diamond|A^c)$, and $\diamond = P$, N, or B. If we consider the loss functions $\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}$, and when $\alpha > \beta$, the following decision rules are formulated (P1, N1, B1) [25, 29]:

P1:	If $P(A|[x]) \geq \alpha$,	decide POS(A);
N1:	If $P(A|[x]) \leq \beta$,	decide NEG(A);
B1:	If $\beta < P(A|[x]) < \alpha$,	decide BND(A).

where,

$$\begin{aligned} \alpha &= \frac{\lambda_{PN} - \lambda_{BN}}{(\lambda_{BP} - \lambda_{BN}) - (\lambda_{PP} - \lambda_{PN})}, \\ \beta &= \frac{\lambda_{BN} - \lambda_{NN}}{(\lambda_{NP} - \lambda_{NN}) - (\lambda_{BP} - \lambda_{BN})}. \end{aligned} \quad (6)$$

The α and β values define the three different regions, giving us an associated risk for classifying an object. When $\alpha = \beta$, we can simplify the rules (P1-B1) into (P2-B2) [29], which divide the regions based solely on α:

P2	If $P(A|[x]) > \alpha$,	decide POS(A);
N2	If $P(A|[x]) < \alpha$,	decide NEG(A);
B2	If $P(A|[x]) = \alpha$,	decide BND(A).

A rough set risk analysis component for a Web-based Support System [24] was created using these minimum-risk decision rules as a foundation. Data mining [23], feature selection [27], information retrieval [16], and decision classifications [11, 33] are just some of the applications in which the DTRS approach has been successfully used.

Yao [29] has shown that the Pawlak and variable-precision approaches are derivable with the decision-theoretic model when the α and β parameters take on certain values. Thus, the variable-precision model can be considered as an intermediate step when using the decision-theoretic approach for rough analysis, albeit with different directions of focus. Yao [31] also identifies two semantic issues regarding the interpretation of parameters and rules in probabilistic rough sets. Determining how much the loss functions can change while maintaining sufficient classification abilities was recently formulated [25]. A learning method for calculating loss functions has been provided [12] and is briefly covered in Section 3.

Similar to DTRS in the respect that they do not necessarily require actual parameter values from user, Bayesian rough sets [21] determine region inclusion of objects by comparing the conditional probability of a set given it's description to that of the prior probability of the set itself. This approach is useful in those situations where any improvements in certainty are beneficial, rather than specifying constraints regarding the classification errors that determine the levels of certainty.

3. Game-Theoretic Rough Sets

Many applications or problems can be expressed as a game between two or more players, so that some aspects of game theory can be utilized [18]. We proposed a game-theoretic rough set model [10]. Decreasing the boundary region is the primary concern in this study. There are two ways to approach this problem: decreasing the boundary region through competition between parameters and decreasing the boundary region through competition between classification approximation measures.

3.1. General Model

Game theory [18] has been one of the core subjects of the decision sciences, specializing in the analysis of decision-making in an interactive environment. The disciplines utilizing game theory include economics [17, 20], networking [1], and machine learning [8].

When using game theory to help determine suitable loss functions, we need to correctly formulate the following: a set of players, a set of strategies for each player, and a set of payoff functions. Game theory uses these formulations to find an optimal strategy for a single player or the entire group of players if cooperation (coordination) is wanted. A single game is defined as,

$$G = \{O, S, F\}, \tag{7}$$

where G is a game consisting of a set of players O using strategies in S. These strategies are measured using individual payoff functions in F.

To begin, the set of players should reflect the overall purpose of the competition. In a typical example, a player can be a person who wants to achieve certain goals. For simplicity, we will be using competition between two players. The two approaches we discuss here are formulated in the same manner, but with different idealogies concerning the direction in which the problem of decreasing the boundary region is addressed.

First, decreasing the boundary region by directly manipulating the region sizes would lead to each player representing one parameter (α and β). A set of players is formulated as $O = \{\alpha, \beta\}$. Competition between the two exposes possible strategies for decreasing the boundary region.

With improved classification ability as the other approach, each player can represent a certain measure such as accuracy (ϕ) and precision ( ). With this in mind, a set of players is formulated as

$O = \{\phi, \ \}$. Through competition, optimal values are attempting to appear for each measure. Although we are measuring accuracy and precision, the choice of measures is ultimately up to the user to decide. We wish to analyze the amount of movement or compromise loss functions can have when attempting to achieve optimal values for these two measures.

Each parameter/measure from either approach is effectively competing with the other to win the "game". Here, the game is to decreasing the size of the boundary region. To compete, each parameter/measure in O has a set of strategies it can employ to achieve payoff. Payoff is the measurable result of actions performed using the strategies. These strategies are executed by the player in order to better their position in the future, e.g., maximize payoff. Individual strategies, when performed, are called *actions*. The strategy set $S_i = \{a_1, \dots, a_m\}$ for any measure i in O contains these actions. A total of m actions can be performed for this player. The strategic goal for ϕ would be along the lines of "acquire a maximal value for approximation accuracy as possible". Likewise, the strategy for α would be to "acquire as many equivalence classes from the boundary region as possible".

3.1.1. Measuring Action Payoff

Payoff, or utility, results from a player performing an action. For a particular payoff for player i performing action a_j, the utility is defined as $\mu_{i,j} = \mu(a_j)$.
A set of payoff functions F contains all μ functions acting within the game G. In this competition between accuracy and precision, $F = \{\mu_\phi, \mu \ \}$, showing payoff functions that measure the increase in accuracy and precision respectively.

A formulated game typically has a set of payoffs for each player. In our approach, given two strategy sets S_1 and S_2, each containing three strategies, the two payoff functions $\mu_\phi : S_1 \mapsto P_1$ and $\mu \ : S_2 \mapsto P_2$ are used to derive the payoffs for ϕ and containing:

$$P_1 = \{\phi_{1,1}, \phi_{1,2}, \phi_{1,3}\}, \qquad (8)$$

$$P_2 = \{ \ _{2,1}, \ _{2,2}, \ _{2,3}\}, \qquad (9)$$

reflecting payoffs from the results of the three actions, i.e., $\mu_\phi(a_j) = \phi_{1,j}$. This is a simple approach that can be expanded to reflect true causal utility based on the opposing player's actions. This means that not only is an action's payoff dependant on the player's action, but also the opposing player's strategy.

After modifying the respective loss functions, the function μ_ϕ calculates the payoff via approximation accuracy. Likewise, the payoff function $\mu \ $ calculates the payoff with approximation precision for deterministic approximations. More elaborate payoff functions could be used to measure the state of a game G, including entropy or other measures according to the player's overall goals [4].

The payoff functions imply that there are relationships between the measures selected as players, the actions they perform, and the probabilities used for region classification. These properties can be used to formulate guidelines regarding the amount of flexibility the user's loss function can have to maintain a certain level of consistency in the data analysis. As we see in the next section, the payoffs are organized into a payoff table in order to perform analysis.

3.1.2. Payoff Tables and Equilibrium

To find optimal solutions for ϕ and , we organize payoffs with the corresponding actions that are performed. A payoff table is shown in Table 5, and will be the focus of our attention.

The actions belonging to ϕ are shown row-wise whereas the strategy set belonging to are column-wise. In Table 5, the strategy set S_1 for ϕ contains three strategies, $S_1 = \{-R_P, +R_N, +R_B\}$, pertaining to actions resulting in a decrease in expected cost for classifying an object into the positive region and an increase in expected cost for classifying objects into the negative and boundary regions. The strategy set for contains the same actions for the second player.

Each cell in the table has a payoff pair $< \phi_{1,i}, \ {}_{2,j} >$. A total of 9 payoff pairs are calculated. For example, the payoff pair $< \phi_{3,1}, \ {}_{3,1} >$ containing payoffs $\phi_{3,1}$ and ${}_{3,1}$ correspond to modifying loss functions to increase the risk associated with classifying an object into the boundary region and to decrease the expected cost associated with classifying an object into the positive region. Measures pertaining to accuracy and precision after the resulting actions are performed for all 9 cases. These payoff calculations populate the table with payoffs so that equilibrium analysis can be performed.

In order to find optimal solutions for accuracy and precision, we determine whether there is equilibrium within the payoff table [6]. This intuitively means that both players attempt to maximize their payoffs given the other player's chosen action, and once found, cannot rationally increase this payoff.

A pair $< \phi^*_{1,i}, \ {}^*_{2,j} >$ is an equilibrium if for any action a_k, where $k \neq i, j$, $\phi^*_{1,i} \geq \phi_{1,k}$ and ${}^*_{2,j} \geq \ {}_{2,k}$. The $< \phi^*_{1,i}, \ {}^*_{2,j} >$ pair is an optimal solution for determining loss functions since no actions can be performed to increase payoff.

Thus, once an optimal payoff pair is found, the user is provided with the following information: a suggested tolerance level for the loss functions and the amount of change in accuracy and precision resulting from the changed loss functions. Equilibrium is a solution to the amount of change loss functions can undergo to achieve levels of accuracy and precision noted by the payoffs.

3.2. Competition Between Parameters

This approach requires that the game be formulated such that each player represents the region parameters α and β. The actions these players choose are summarized in Table 2. They either increase or decrease their actual probabilistic values by a small increment or decrement respectively. This will result in equivalence classes from the boundary region moving to either the positive region (decrease of α) or negative region (increase of β).

Constants $\{c_1, \ldots, c_3\}$ represent a percentage of increase/decrease in the probabilistic values of α and β. For each action to be distinct and non-redundant, the following should hold: $c_1 < c_2 < c_3$. For example, given the configuration $\{c_1 = 0.05, c_2 = 0.07, c_3 = 0.15\}$ (an increase/decrease of 5%, 7%, and 15%) with initial values of $\{\alpha = 0.75, \beta = 0.25\}$, new possible parameter region values given the selection of any action would be $\alpha = \{0.7125, 0.6975, 0.6375\}$ and $\beta = \{0.2625, 0.2675, 0.2875\}$. We can see that using the same constants for increasing and decreasing α and β respectively results in different magnitudes of growth. This gives an advantage to large initial values for either parameter since higher probabilities are more likely to influence a larger number of equivalence classes.

Modifying a larger probability will have more of an effect regarding the amount of equivalence classes being moved from from the BND region. This will have an adverse effect when trying to increase the NEG region, as $\beta < \alpha$ will be the case in most, if not all situations. We must level the

Table 2. The strategy scenario for decreasing the boundary region (winner-take-all). Each parameter can access three actions differentiated by different values of a scalar c.

Action (Strategy)	Method	Outcome
α_1 ($\downarrow\alpha$)	Decrease α by c_1	Larger POS region
α_2 ($\downarrow\alpha$)	Decrease α by c_2	
α_3 ($\downarrow\alpha$)	Decrease α by c_3	
β_1 ($\uparrow\beta$)	Increase β by c_1	Larger NEG region
β_2 ($\uparrow\beta$)	Increase β by c_2	
β_3 ($\uparrow\beta$)	Increase β by c_3	

playing field for all players through the use of normalized payoff functions. That is, we must formulate the payoff functions in such a way that the magnitude in parameter change is indicative of the number of equivalence classes that are moved from the BND region:

$$\mu(\alpha_i) \quad = \quad \frac{(|POS|' - |POS|)}{(\alpha - (\alpha * c_i))}, \quad \mu(\beta_i) = \frac{(|NEG|' - |NEG|)}{((\beta * c_i) - \beta)}, \tag{10}$$

where μ is the payoff function for evaluating a given action, $|POS|$ and $|NEG|$ is the size of the current positive and negative region, $|POS|'$ and $|NEG|'$ is the size of the new positive and negative regions given the action will be executed, and α and β are the probabilistic parameter values. If $|POS|' \geq |POS|$ or $|NEG|' \geq |NEG|$, μ is a monotonically increasing measure with respect to the number of objects. Likewise, if $|POS|' \leq |POS|$ or $|NEG|' \leq |NEG|$, μ is a monotonically non-increasing measure with respect to the number of objects. For other monotonic measures dealing with certainty gain and cost, a thorough review can be found in Reference [2]. We have now formulated competition between region parameters in such a way that each player represents a parameter, a set of actions have been identified in increase or decrease the parameter's value, and a normalized payoff function that results in an equal competitive footing for both players.

The winner-take-all scenario will result in only one parameter being modified. Forming a coalition between the parameters (allowing for cooperation between them) can result in both parameters being modified at the same time. This idea of cooperative game theory allows for a scenario that is very similar to winner-take-all approach. However, an additional strategy must be formed and added to each player's strategy set. This extra action will allow for some sacrifice to be made in order to improve the other player's position to reduce the BND region further.

In the case of decreasing and increasing the parameters α and β respectively through the three actions stated above, a fourth action to enable the increase of α and the decrease of β will be considered, as shown in Table 3.

This at first may seem non-constructive, since this action could result in a larger BND region - the opposite of what we wish to achieve. However, this action exists in order to allow more equivalence classes in the BND region to be available for movement to the other region. A sacrifice of some POS region classes will make them available to be eventually classified in the NEG region, if needed.

Table 3. The strategy for decreasing the boundary region (coalition). The original three actions are present with an additional fourth action for each parameter.

Action (Strategy)	Outcome
$\{\alpha_1, \ldots, \alpha_3\}$ (old)	Larger POS region, Smaller BND region
α_4 ($\uparrow\alpha$)	Smaller POS region, Larger BND region
$\{\beta_1, \ldots, \beta_3\}$ (old)	Smaller NEG region, Larger BND region
β_4 ($\downarrow\beta$)	Larger NEG region, Smaller BND region

Again, we may use the same payoff functions as the winner-take-all scenario, with the addition of a coalition function to evaluate the benefit of the fourth action:

$$\mu(\alpha_4) = \frac{(|POS| - |POS|')}{((\alpha * c^*) - \alpha)} - \frac{(|NEG|' - |NEG|)}{((\beta * c^*) - \beta)}, \tag{11}$$

$$\mu(\beta_4) = \frac{(|NEG| - |NEG|')}{(\beta - (\beta * c^*))} - \frac{(|POS|' - |POS|)}{(\alpha - (\alpha * c^*))}, \tag{12}$$

where c^* is the action for the respective parameter that is maximum of $(c_1, \ldots, c_3)$. The evaluation of the fourth action results in the following decision on whether or not execute it:

$$\mu(\alpha_4) \begin{cases} < 0 & \alpha \text{ selects } \alpha_4, \beta \text{ selects } c^* \\ \geq 0 & \alpha_4 \text{ is not selected} \end{cases}, \quad \mu(\beta_4) \begin{cases} < 0 & \beta \text{ selects } \beta_4, \alpha \text{ selects } c^* \\ \geq 0 & \beta_4 \text{ is not selected} \end{cases} \tag{13}$$

Essentially, the fourth action will only be selected if the other player truly benefits from an increased size of the BND region.

3.3. Competition Between Classification Measures

In this approach, a game is formulated in such a way that each player represents a classification approximation measure, such as approximation accuracy (ϕ) and approximation precision ( ). The actions these players choose are summarized in Table 4. They either increase or decrease the conditional risk by modifying the associated loss functions in (5). This, in turn, changes the sizes of the classification regions.

For a particular payoff for player i performing action a_j, the utility is defined as $\mu_{i,j} = \mu(a_j)$. A payoff is simply the benefit or cost each player acquires after performing a given action. A set of payoff functions F is a set of all μ functions used to derive payoff within the game. In this competition between accuracy and precision, $F = \{\mu_\phi, \mu\ \}$, showing payoff functions that measure the increase in accuracy and precision respectively [10].

A game typically has a set of strategies S_i for each player i. Given player 1 employs an accuracy-seeking strategy within S_1 and player 2 employs a precision-seeking strategy in S_2, the two payoff functions $\mu_\phi : S_1 \mapsto P_1$ and $\mu\ : S_2 \mapsto P_2$ are used to derive the payoffs for ϕ and containing:

Table 4. The strategy scenario of increasing approximation accuracy [10].

Action (Strategy)	Method	Outcome
a_1 ($\downarrow R_P$)	Decrease λ_{PP} or λ_{PN}	Larger POS region
a_2 ($\uparrow R_N$)	Increase λ_{NP} or λ_{NN}	Smaller NEG region
a_3 ($\uparrow R_B$)	Increase λ_{BP} or λ_{BN}	Smaller BND region

$$P_1 = \{\phi_{1,1}, \phi_{1,2}, \phi_{1,3}\}, \tag{14}$$
$$P_2 = \{ \ _{2,1}, \ _{2,2}, \ _{2,3}\}, \tag{15}$$

reflecting payoffs from the results of the three actions, i.e., $\mu_\phi(a_j) = \phi_{1,j}$ [25]. This simple approach can be expanded to reflect true causal utility based on the opposing player's actions. A view of the correspondence between players, strategies, and payoffs can seen in Table 5. It means that not only is an action's payoff dependant on the player's action, but also the opposing player's strategy.

After modifying the respective loss functions, the function μ_ϕ calculates the payoff via approximation accuracy. Likewise, the payoff function μ calculates the payoff with approximation precision for deterministic approximations. More elaborate payoff functions could be used to measure the state of a game, including entropy or other measures according to the player's overall goals [4].

The payoff functions imply that there are relationships between the measures selected as players, the actions they perform, and the probabilities used for region classification. These properties can be used to formulate guidelines regarding the amount of flexibility the user's loss function can have to maintain a certain level of consistency in the data analysis.

The players, actions, and payoffs are organized into a payoff table in order to perform analysis, as shown in Table 5. If one is interested in maximizing accuracy, all needed is to choose a suitable action that fits with his or her acceptable risk.

Approximation accuracy (ϕ), is defined as the ratio measured between the size of the lower approximation of a set A to the upper approximation of a set A. A large value of ϕ indicates that we have a small boundary region. To illustrate the change in approximation accuracy, suppose we have player ϕ taking two turns in the competition. For the first turn, player ϕ executes action a_1 from it's strategy set. When it is time to perform another turn, the player executes action a_2. Ultimately, since the player's goal is to increase approximation accuracy, we should measure that $\phi_{a_1} \leq \phi_{a_2}$. If this is not the case ($\phi_{a_1} > \phi_{a_2}$), the player has chosen a poor second action from it's strategy set.

The second player, approximation precision ( ), observes the relationship between the upper approximation and a set. In order to increase precision, we need to make $|\underline{apr}(A)|$ as large as possible. For non-deterministic approximations, Yao [28] suggested an alternative precision measure.

In general, the two measures ϕ and show the impacts that the loss functions have on the classification ability of the DTRS model. Modifying the loss functions contribute to a change in risk (expected cost). Determining how to modify the loss functions to achieve different classification abilities requires a set of risk modification strategies.

There is a limit to the amount of change allowable for loss functions. For example, the action of reducing the expected cost R_P. We can reduce this cost any amount and rule (PN) will be satisfied.

Table 5. Payoff table for ϕ, payoff calculation (deterministic).

		Precision ( ) S_2		
		$\downarrow R_P$	$\uparrow R_N$	$\uparrow R_B$
Accuracy (ϕ) S_1	$\downarrow R_P$	$\phi_{1,1}$, $_{1,1}$	$\phi_{1,2}$, $_{1,2}$	$\phi_{1,3}$, $_{1,3}$
	$\uparrow R_N$	$\phi_{2,1}$, $_{2,1}$	$\phi_{2,2}$, $_{2,2}$	$\phi_{2,3}$, $_{2,3}$
	$\uparrow R_B$	$\phi_{3,1}$, $_{3,1}$	$\phi_{3,2}$, $_{3,2}$	$\phi_{3,3}$, $_{3,3}$

However, the rules (NN) and (BN) are also sensitive to the modification of R_P, denoted R_P^*. R_P^* must satisfy $R_P^* \geq (R_N - R_P)$ and $R_P^* \geq (R_B - R_P)$. This results in upper limit of t_{PP}^{max} for λ_{PP} and lower limit of t_{PN}^{min} for λ_{PN} [10]. Assuming that $\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}$, we calculate the following,

$$t_{PP}^{max} \quad \leq \quad \frac{\lambda_{BP} - \lambda_{PP}}{\lambda_{PP}} \quad , \qquad t_{PN}^{min} < \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{PN}}. \tag{16}$$

That is, t_{PP}^{max} is the *tolerance* that loss function λ_{PP} can have (t_{PP}^{min} for λ_{PN}). Tolerance values indicate how much change a user can have to their risk beliefs (loss functions) in order to maintain accuracy and precision measures of $< \phi_{1,i}^*, \ {}_{2,j}^* >$. In brief, when selecting a strategy, i.e., ($\downarrow R_P$), the game calculates payoffs by measuring the approximation accuracy and prediction that result from modifying the loss functions λ_{PP} and λ_{PN}. The new loss functions, λ_{PP}^* and λ_{PN}^* are used to calculate a new expected loss R_P^*. In order to maintain the levels of accuracy and precision stated in the payoffs, the user must have new loss functions within the levels of t_{PP}^{max} for λ_{PP} and t_{PN}^{min} for λ_{PN} [25].

We will see in the next section that having this game repeat results in a sequence of loss function modifications that optimize the classification region sizes for a given classification approximation measure.

3.4. Learning Optimal Parameters

Game theory can be a powerful tool for analyzing the relationships between classification ability and risk. It could also provide a means for the user to change their beliefs regarding the types of decisions they can make [9]. This can occur by changing their risk beliefs instead of the actual probabilities. This is advantageous as many users cannot intuitively describe their decision needs in terms of probabilities.

3.4.1. Learning Procedure

The learning procedure utilizes game theory in conjunction with the DTRS model to aid in decision making [25]. There are five steps to be performed [12]:

1. Game Formulation.
2. Strategy Formulation.
3. Payoff Measurement.

4. Competition Analysis (**repeated**).
   - Every time a loss function is modified, competition analysis must be performed on updated measurements.
   - New payoff tables are created after each learning iteration.
   - Observation of the game within payoff tables and examining the relationships between the actions undertaken and the payoffs associated with those actions.
5. Result Acquisition (**repeated**).
   - This step is repeated so that the loss functions will be modified in a correct manner.
   - The action selected is used to learn new values of loss functions.
   - The result acquisition step interprets the results of the competition.

We must continually repeat Steps 4 and 5 in order for a given measure to be optimized by the loss functions. Once Step 5 occurs (meaning that a suitable action has been chosen and recorded), we re-calculate the payoff table to reflect new values that arise if more actions are performed. The actions remain the same [12]. This repetition continues until the next action that should be performed exceeds the acceptable levels of tolerance a user may have.

3.4.2. Repetitive Risk Modifications

Modifying the conditional risk changes the sizes of the regions. Doing so repeatedly allows for the regions to change in size in order to reflect the learning of loss functions. Referring to Table 4, choosing action $\downarrow R_P$ (decreasing R_P) will increase the size of the positive region [12]. Likewise, choosing action $\uparrow R_N$, will decrease the size of the negative region. The last option, choosing action $\uparrow R_B$ will decrease the size of the boundary region. Performing these actions repeatedly allows for the learning of loss functions.

The repetitive modification of the conditional risk associated with a given action can be thought of as a learning procedure. The new value for a loss function should exhibit a measurable change, dependant on its previous value, the probability that an object will be classified into that region, and the amount of classification ability changes.

3.4.3. The Parameter Learning Sequence

In order to find better values of classification approximation measures, we learn optimal values for loss functions through the use of game theory. We choose a sequence of strategies that will result in an increase in the classification approximation measure from the payoff tables. Recording these actions into a sequence of choices can give us learning criteria for adjusting these loss functions [12].

Let Γ be the measure we wish to optimize and $\mu(\Gamma)$ be the actual value of that measure given the current conditions. A new loss function, resulting from the modification of the current value, given the choice of action a_i and classification into a set A, is as follows [12]:

$$\lambda^*_{\diamond P} \quad = \quad \lambda_{\diamond P} \pm \big(\lambda_{\diamond P} \cdot P(A|[x]) \cdot (\mu(\Gamma) - \mu(a_i))\big), \tag{17}$$

Table 6. Region sizes with changes in region parameters. c_0 is initial value. As α decreases, the POS region increases in size. Likewise, as β increases, the NEG region increases in size.

						Region Size (% of universe)		
Strategy	Constant	α	β	ϕ		POS	NEG	BND
$\downarrow\alpha$	$c_0 = 1.0$	0.75	0.25	0.56	0.35	34%	38%	28%
	$c_1 = 0.05$	0.7125	0.25	0.61	0.38	38%	38%	24%
	$c_2 = 0.07$	0.6975	0.25	0.62	0.39	39%	38%	23%
	$c_3 = 0.15$	0.6375	0.25	0.73	0.46	46%	38%	16%
$\uparrow\beta$	$c_0 = 1.0$	0.75	0.25	0.56	0.35	34%	38%	28%
	$c_1 = 0.05$	0.75	0.2625	0.56	0.35	34%	38%	28%
	$c_2 = 0.07$	0.75	0.2675	0.56	0.35	34%	38%	28%
	$c_3 = 0.15$	0.75	0.2875	0.57	0.35	34%	40%	26%

where $\diamond = P$, N, or B. The original loss function is changed by the proportion of the difference in classification ability $(\mu(\Gamma) - \mu(a_i))$ multiplied by the expected cost $(\lambda_{\diamond P} \cdot P(A|[x]))$. This allows for gradual learning based on the significance of the objects and the degree of change in classification ability. Referring to Table 4, if we wish to increase the size of the positive region, we would choose action $\downarrow R_P$. The corresponding modification into a A's complement is given by [12]:

$$\lambda^*_{\diamond N} \quad = \quad \lambda_{\diamond N} \pm \big(\lambda_{\diamond N} \cdot P(A^c|[x]) \cdot (\mu(\Gamma) - \mu(a_i)\big). \tag{18}$$

From Table 4, we could decrease either λ_{PP} or λ_{PN}. $P(A^c|[x])$ is used since it has a loss function that measures the cost of classifying an object into a set's complement. This is done by solving either of the following two equations [12]:

$$\lambda^*_{PP} \quad = \quad \lambda_{PP} + \big(\lambda_{PP} \cdot P(A|[x]) \cdot (\mu(\Gamma) - \mu(a_i)\big), \tag{19}$$
$$\lambda^*_{PN} \quad = \quad \lambda_{PN} + \big(\lambda_{PN} \cdot P(A^c|[x]) \cdot (\mu(\Gamma) - \mu(a_i)\big). \tag{20}$$

This learning method modifies an initial loss function according to game theory analysis. By selecting a suitable action in the payoff table, the change in between that outcome in the one previously is scaled according to it's influence on the overall classification ability.

4. GTRS Experiment Results

In this section, we will look at experiment results regarding the application of the GTRS model with competition between parameters, competition between classification approximation measures, and learning parameters from the data itself.

Table 7. Region sizes with changes in conditional risk.

							Region Size (% of universe)		
Strategy	Method	α	β	λ	ϕ		POS	NEG	BND
$\downarrow R_p$	Decrease	0.75	0.25	4.0	0.56	0.35	34%	38%	28%
	λ_{PN}	0.722	0.25	3.6	0.61	0.38	38%	38%	24%
		0.673	0.25	3.06	0.65	0.41	41%	38%	21%
$\uparrow R_N$	Increase	0.75	0.25	4.0	0.56	0.35	34%	38%	28%
	λ_{NP}	0.75	0.227	4.4	0.55	0.35	34%	37%	29%
		0.75	0.198	5.06	0.53	0.35	34%	35%	31%
$\uparrow R_B$	Increase	0.75	0.25	1.0	0.56	0.35	34%	38%	28%
	λ_{BP}	0.714	0.263	1.2	0.61	0.38	38%	38%	24%
		0.685	0.276	1.38	0.66	0.41	41%	39%	20%

The data used is the stock market data from the New Zealand Exchange Limited (NZX). The NZX data begins July 31, 1991 and ends April 27, 2000. This data set gives us a wide variety of testing challenges in order to test our model. Data representing closing price, opening price, highest price reached during the day, and lowest price reached during the day is present.

New conditional attributes that conveyed statistical properties of the data accumulated by looking at the computational finance domain [22]. The following statistics were used in this study: Moving Average Convergence/Divergence (MACD), Moving Average over a 5-day period (MA5), Moving Average over a 12-day period (MA12), Price Rate of Change (PROC), and Wilder Relative Strength Index (RSI). The decision attribute is the trend associated closing price of the next day, with values of either -1, 0, or 1. The value 1 represents that the next day's price is higher than that of the current date, -1 lower and 0 no change.

4.1. Parameter Competition Results

Table 6 summarizes new region sizes (in total universe percentage) that result in changing region parameters given a chosen strategy. As we decrease α, we see the size of the POS region increase. This results in objects in the BND region moving into the POS region. In addition, we see increases in both accuracy of approximation (ϕ) and approximation precision ( ), with the largest increase occurring when $c_3 = 0.15$. Likewise, increasing β results in an increase in the size of the NEG region from the movement of objects from the BND region to NEG region, albeit at a much slower rate. Approximation precision is not affected and a very small increase in accuracy in noted due a small amount of equivalence classes being attached to the NEG region when $c_3 = 0.15$.

4.2. Measure Competition Results

Table 7 shows the progression of new region sizes (in total universe percentage) from changing loss functions given a chosen strategy [12]. As we decrease λ_{PN}, we see the parameter α decrease. This

Table 8. Region sizes with changes in conditional risk.

α	β	λ_{PN}	λ_{NP}	λ_{BP}	λ_{BN}	ϕ	
1	0	4	4	1	1	0.06	0.06
0.938	0	4	4	1.2	1	0.06	0.06
0.872	0	4	4	1.44	1	0.09	0.09
0.804	0	4	4	1.728	1	0.2	0.19
0.751	0	3.2	4	1.728	1	0.34	0.32
0.733	0.08	3.2	4	1.728	1.2	0.41	0.38
0.707	0.162	3.2	4	1.728	1.44	0.5	0.38
0.669	0.243	3.2	4	1.728	1.728	0.64	0.41
0.608	0.32	3.2	4	1.728	2.07	0.8	0.47
0.608	0.421	3.2	3.2	1.728	2.07	0.89	0.47

results in objects in the BND region moving into the POS region. Likewise, increasing λ_{NP} results in a decreased β and objects moving from the NEG region to BND region. The last strategy, increasing λ_{BP}, results in α decreasing, β increasing, and objects in the BND region moving into both the POS and NEG regions.

The remaining two regions are decreased or increased when increasing or decreasing a region size respectively. This is intuitive if one thinks that to increase the positive region, one may decrease the risk associated with classifying an object into that region. A decreased risk will result in more objects being classified to that region. Decreasing the risk of a correct classification will result in a increased risk of an incorrect classification. This is a side-effect. That is, if λ_{PN} is lowered, λ_{BP} and λ_{NP} will increase to some extent [12].

4.3. Learning Results

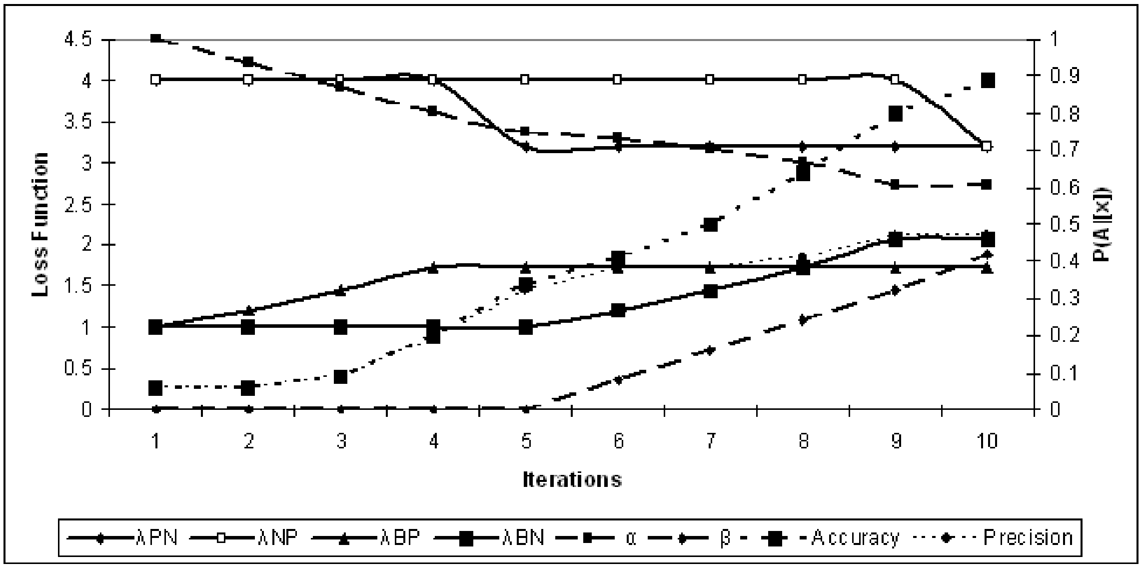
Table 8 shows the progression of learning procedure in determine suitable values for loss functions. A goal of $\phi \geq 0.85$ is set for the game to attempt to reach. Initial values of $\alpha = 1.0$ and $\beta = 0.0$ are shown. During the first investigation of the payoff table, we find there is negligible difference for the measures. The system therefore chooses to increase $\lambda_B P$ as initiation. The system continues to increase $\lambda_B P$ for the next two iterations. Approximation accuracy has increased from 0.06 to 0.20, with precision increasing from 0.06 to 0.19 likewise.

The game theory payoff table then tells us that we should decrease $\lambda_P N$ next, increasing accuracy and precision to $(0.34, 0.32)$. Once the system finds an increase to $\lambda_B N$ to be beneficial (iteration 6), we begin to see the parameter β increase.

The final configuration of the system, once reaching above our original goal of $\phi \geq 0.85$, is the following: $\alpha = 0.608$, $\beta = 0.421$, $\lambda_{PP} = 1.0$, $\lambda_{PN} = 3.2$, $\lambda_{BP} = 1.728$, $\lambda_{BN} = 2.07$, $\lambda_{NP} = 3.1$, $\lambda_{NN} = 1.0$, $\phi = 0.89$, and $= 0.47$.

We saw the largest increases in approximation accuracy by modifying λ_{BN} or λ_{BP}, or the loss

Figure 1. Learning loss functions vs Iteration, Measure.



incurred by incorrectly classifying an object into the boundary region when it in fact belongs in either the negative or positive region. This is because a change in these costs influence both α and β values, potentially allowing objects to enter positive and negative regions simultaneously. About one-third of the strategies had negative payoffs from their execution (if they were chosen). In this particular application, this can be thought of as a re-adjustment strategy.

5. Conclusion

We have proposed a formulation of the Game-theoretic Rough Set (GTRS) model. This model provides a means to decrease the size of the boundary region through competition between both the set of region parameters and a set of classification approximation measures. To do this, a game is formulated as a set of players either competing against or cooperating with each other, a set of actions that each player can execute, and definitions of suitable payoff functions for evaluating the outcomes of these actions.

In the first approach, each player is represented as a region parameter, which defines upper and lower probability values. These values determine the minimum conditional probability that an object must have in order to be included in the positive, negative, and boundary regions. To decrease the size of the boundary region, objects within must be moved to either the positive or negative regions. This constitutes a decision problem regarding which parameter to adjust. This decision problem is solved through the evaluation of payoff tables populated with payoff functions. The parameters may work against each other in order to increase their size, or form a coalition to help achieve some balance between the increases.

The second approach has each player represented as a classification approximation measure to maximize. We discussed the use of the classification approximation measures of approximation accuracy and approximation precision, although other measures may be equally applicable. Actions performed in this approach again consist of increasing or decreasing the size of the classification regions. This is achieved by modifying the values of decision-theoretic rough set loss functions within an acceptable range. We formulate this process of acquiring new loss functions gradually through a learning process. Generally speaking, an increase in risk for one classification action will result in a decrease in risk for other classification actions. We provide a parameter learning method using game theory that defines loss functions according to an optimal value of classification ability.

New avenues of research are opened up by the formation of the GTRS model. The interpretation of major challenges facing probabilistic rough sets with game theory offers us new insights into the competition and cooperation between measures, parameters, and conditional risk associated with classification.

References

[1] Bell, M. G. F.: The use of game theory to measure the vulnerability of stochastic networks, *IEEE Transactions on Reliability*, **52**(1), 2003, 63–68.

[2] Blaszczynski, J., Greco, S., Slowinski, R., Szelg, M.: Monotonic variable consistency rough set approaches, *International journal of approximate reasoning*, **50**(7), 2009, 979–999.

[3] Ciucci, D.: A Unifying Abstract Approach for Rough Models, *Proceedings of Rough Sets and Knowledge Technology (RSKT'08)*, LNAI 5009, 2008, 371–378.

[4] Duntsch, I., Gediga, G.: Uncertainty measures of rough set prediction, *Artificial Intelligence*, **106**(1), 1998, 109–137.

[5] Fischer, J., Wright, R. N.: An application of game-theoretic techniques to cryptography, *Discrete Mathematics and Theoretical Computer Science*, **13**, 1993, 99–118.

[6] Fudenberg, D., Tirole, J.: *Game Theory*, The MIT Press, 1991.

[7] Greco, S., Matarazzo, B., Slowinski, R.: Parameterized rough set model using rough membership and Bayesian confirmation measures, *International Journal of Approximate Reasoning*, **49**(2), 2008, 285–300.

[8] Herbert, J. P., Yao, J. T.: A Game-Theoretic Approach to Competitive Learning in Self-Organizing Maps, *International Conference on Natural Computation (ICNC'05)*, LNCS 3610, 2005, 129–138.

[9] Herbert, J. P., Yao, J. T.: Rough Set Model Selection for Practical Decision Making, *Proceedings of Fuzzy Systems and Knowledge Discovery (FSKD'07)*, Vol. III, 2007, 203–207.

[10] Herbert, J. P., Yao, J. T.: Game-Theoretic Risk Analysis in Decision-Theoretic Rough Sets, *Proceedings of Rough Sets and Knowledge Technology (RSKT'08)*, LNAI 5009, 2008, 132–139.

[11] Herbert, J. P., Yao, J. T.: Criteria for choosing a rough set model, *Computers and Mathematics with Applications*, **57**(6), 2009, 908–918.

[12] Herbert, J. P., Yao, J. T.: Learning Optimal Parameters in Decision-Theoretic Rough Sets, *Proceedings of Rough Sets and Knowledge Technology (RSKT'09)*, LNAI 5589, 2009, 610–617.

[13] Katzberg, J. D., Ziarko, W.: Variable precision rough sets with asymmetric bounds, *Rough Sets, Fuzzy Sets and Knowledge Discovery* (W. Ziarko, Ed.), Springer, London, 1994, 167–177.

[14] Kotlowski, W., Dembczynski, K., Greco, S., Slowinski, R.: Stochastic dominance-based rough set model for ordinal classification, *Information Sciences*, **178**(21), 2008, 4019–4037.

[15] Li, T. R., Ruan, D., Geert, W., Song, J., Xu, Y.: A rough sets based characteristic relation approach for dynamic attribute generalization in data mining, *Knowledge-Based Systems*, **20**(5), 2007, 485–494.

[16] Li, Y., Zhang, C., Swanb, J.: Rough set based model in information retrieval and filtering, *Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis*, 1999, 398–403.

[17] Nash, J.: The bargaining problem, *Econometrica*, **18**(2), 1950, 155–162.

[18] Neumann, J. V., Morgenstern, O.: *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, 1944.

[19] Pawlak, Z.: Rough sets, *International Journal of Computer and Information Sciences*, **11**, 1982, 341–356.

[20] Roth, A.: The evolution of the labor market for medical interns and residents: a case study in game theory, *Political Economy*, **92**, 1984, 991–1016.

[21] Slezak, D., Ziarko, W.: The investigation of the Bayesian rough set model, *International Journal of Approximate Reasoning*, **40**(1-2), 2005, 81–91.

[22] Taylor, S.: *Modelling Financial Time Series*, John Wiley & Sons Ltd, 1986.

[23] Tsumoto, S.: Accuracy and coverage in rough set rule induction, *Proceedings of RSCTC'02*, LNCS 2475, 2002, 373–380.

[24] Yao, J. T., Herbert, J. P.: Web-based Support Systems with Rough Set Analysis, *Proceedings of Rough Sets and Emerging Intelligent Systems Paradigms (RSEISP'07)*, LNAI 4585, 2007, 360–370.

[25] Yao, J. T., Herbert, J. P.: A Game-Theoretic Perspective on Rough Set Analysis, *Journal of Chongqing University of Posts and Telecommunications*, **20**(3), 2008, 291–298.

[26] Yao, J. T., Herbert, J. P.: Financial time-series analysis with rough sets, *Applied Soft Computing*, **9**(3), 2009, 1000–1007.

[27] Yao, J. T., Zhang, M.: Feature Selection with Adjustable Criteria, *Proceedings of RSFDGrC'05*, LNAI 3641, 2005, 204–213.

[28] Yao, Y. Y.: Probabilistic approaches to rough sets, *Expert Systems*, **20**(5), 2003, 287–297.

[29] Yao, Y. Y.: Decision-Theoretic Rough Set Models, *Proceedings of Rough Sets Knowledge Technology (RSKT'07)*, LNAI 4481, 2007, 1–12.

[30] Yao, Y. Y.: Probabilistic rough set approximations, *International Journal of Approximate Reasoning*, **49**(2), 2008, 255–271.

[31] Yao, Y. Y.: Two Semantic Issues in a Probabilistic Rough Set Model, *Fundamenta Informaticae*, **108**(3-4), 2011, 249-265.

[32] Yao, Y. Y., Wong, S. K. M., Lingras, P.: A decision-theoretic rough set model, in: *Methodologies for Intelligent Systems* (Z. W. Ras, M. Zemankova, M. L. Emrich, Eds.), vol. 5, North-Holland, New York, 1990, 17–24.

[33] Zhou, X., Li, H.: A Multi-view Decision Model Based on Decision-theoretic Rough Set, *Proceedings of Rough Sets and Knowledge Technology (RSKT'09)*, LNAI 5589, 2009, 650–657.

[34] Ziarko, W.: Variable precision rough set model, *Journal of Computer and System Sciences*, **46**, 1993, 39–59.

[35] Ziarko, W.: Acquisition of heirarchy-structured probabilitic decision tables and rules from data, *Expert Systems*, **20**, 2003, 305–310.

[36] Ziarko, W., Fei, X.: VPRSM approach to WEB searching, *Proceedings of RSCTC'02*, LNCS 2475, 2002, 514–521.