J. P. Herbert, J.T. Yao, Game-Theoretic Risk Analysis in Decision-Theoretic Rough Sets, Proceedings of the 3rd International Conference on Rough Sets and Knowledge Technology (RSKT'08), May 17-19, 2008, Chengdu, China, LNAI 5009, pp132-139.

Game-Theoretic Risk Analysis in Decision-Theoretic Rough Sets

Joseph P. Herbert JingTao Yao

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: [herbertj,jtyao]@cs.uregina.ca

Abstract. Determining the correct threshold values for probabilistic rough set models has been a heated issue among the community. This article will formulate a game-theoretic approach to calculating these thresholds to ensure correct approximation region size. By finding equilibrium within payoff tables created from approximation measures and modified conditional risk strategies, we provide the user with tolerance levels for their loss functions. Using the tolerance values, new thresholds are calculated to provide correct classification regions. Better informed decisions can be made when utilizing these tolerance values.

1 Introduction

In rough sets [10], a set within the universe of discourse is approximated. Rough set regions are defined with these approximations. One of the goals of improving the classification ability of rough sets is to reduce the boundary region, thus, reducing the impact that this uncertainty has on decision making. The decision-theoretic rough set [16] and variable-precision rough set [17] models were proposed solutions to this problem of decreasing the boundary region.

The decision-theoretic rough set model (DTRS) [14] utilizes the Bayesian decision procedure to calculate rough set classification regions. Loss functions correspond to the risks involved in classifying an object into a particular classification region. This gives the user a scientific means for linking their risk tolerances with the probabilistic classification ability of rough sets [12].

The decision-theoretic model observes a lower and upper-bound threshold for region classification [13]. The thresholds α and β provide the probabilities for inclusion into the positive, negative, and boundary regions. The α and β thresholds are calculated through the analysis of loss function relationships, thus, a method of reducing the boundary region materializes from the modification of the loss functions. Utilizing game theory to analyze the relationships between classification ability and the modification of loss functions, we can provide the user with a means for changing their risk tolerances.

Classification ability of a rough set analysis system is a measurable characteristic [4]. In this article, we introduce a method for calculating loss tolerance using game theory to analyze the effects of modifying the classification risk. This also provides an effective means of determining how much a loss function can fluctuate in order to maintain effective classification ability.

2 Decision-Theoretic Rough Sets

The decision-theoretic approach is a robust extension of rough sets for two reasons. First, it calculates approximation parameters by obtaining easily understandable notions of risk or loss from the user [14, 15].

2.1 Loss Functions

Let $P(w_j|\mathbf{x})$ be the conditional probability of an object x being in state w_j given the object description $\mathbf{x}$. 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 to classify an object into $POS(A)$, $NEG(A)$, and $BND(A)$ respectively. Let $\lambda(a_\diamond|A)$ denote the loss incurred for taking action $a_\diamond$ when an object is in A, and let $\lambda(a_\diamond|A^c)$ denote the loss incurred by taking the same action when the object belongs to A^c. This can be given as loss functions $\lambda_{\diamond P} = \lambda(a_\diamond|A)$, $\lambda_{\diamond N} = \lambda(a_\diamond|A^c)$, and $\diamond = P$, N, or B. Through the combination of the set of loss functions, α, β, and γ parameters can be calculated to define the regions.

A crucial assumption when using this model is that the set of loss functions is provided by the user. This is a drawback, as it is still dependant upon user-provided information for calculating rough set region boundaries. In order to pass this obstacle, a method of calculating loss functions from the relationships found within the actual data must be found. Although this is beyond the scope of this article, we can provide a method for determining how much these loss functions can change, an equally important problem.

2.2 Conditional Risk

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

$$\begin{aligned}
R_P = R(a_P|[x]) &= \lambda_{PP}P(A|[x]) + \lambda_{PN}P(A^c|[x]),\\
R_N = R(a_N|[x]) &= \lambda_{NP}P(A|[x]) + \lambda_{NN}P(A^c|[x]),\\
R_B = R(a_B|[x]) &= \lambda_{BP}P(A|[x]) + \lambda_{BN}P(A^c|[x]), \qquad (1)
\end{aligned}$$

where $\lambda_{\diamond P} = \lambda(a_\diamond|A)$, $\lambda_{\diamond N} = \lambda(a_\diamond|A^c)$, and $\diamond = P$, N, or B. R_P, R_N, and R_B are the expected losses of classifying an object into the positive region, negative region, and boundary region respectively. The Bayesian decision procedure leads to the following minimum-risk decision rules (PN-BN):

$$\begin{array}{lll}
\text{(PN)} & \text{If } R_P \le R_N \text{ and } R_P \le R_B, & \text{decide POS}(A);\\
\text{(NN)} & \text{If } R_N \le R_P \text{ and } R_N \le R_B, & \text{decide NEG}(A);\\
\text{(BN)} & \text{If } R_B \le R_P \text{ and } R_B \le R_N, & \text{decide BND}(A);
\end{array}$$

These minimum-risk decision rules offer us a foundation in which to classify objects into approximation regions. They give us the ability to not only collect decision rules from data frequent in many rough set applications [6], but also the calculated risk that is involved when discovering (or acting upon) those rules.

3 A Game-Theoretic Calculation for Conditional Risk

We stated previously that the user could make use of a method of linking their notions of cost (risk) in taking a certain action and classification ability of the classification system. Game theory can be a powerful mathematical paradigm for analyzing these relationships and also provides methods for achieving optimal configurations for classification strategies. It could also provide a means for the user to change their beliefs regarding the types of decisions they can make [7]. They would not have to change the probabilities themselves, only their risk beliefs. This is beneficial as many users cannot intuitively describe their decision needs in terms of probabilities.

3.1 The Boundary Region and Conditional Risk

We wish to emphasize the relationship between the conditional risk, loss functions, and boundary region. Classification can be performed by following the minimum risk decision rules PN, NN, and BN or by using the α and β parameters to define region separation. We wish to make the boundary region smaller by modifying either method so that the positive region can be increased. To measure the changes made to the regions, we use two measures: approximation accuracy (ϕ) and approximation precision (ψ).

When increasing the size of the positive region, the size of the lower approximation is made larger. By recording the accuracy and precision measures, we can directly see the impact this has on classification ability. To increase the size of the lower approximation, measured by ϕ and ψ, we can observe the changes in the conditional risk found in Equation 1. That is, to increase the size of the lower approximation, we can reduce the risk associated with classifying an object into the positive region. This can be done by modifying the loss functions.

Furthermore, while doing this, we need to maintain the size of $|\overline{apr}(A)|$. Recalling rules (PN, NN, BN), we see that in order to increase the size of the lower approximation, we need decrease the expected loss R_P. This results in more objects being classified into the positive region since it is less "risky" to do so. An increase R_N and R_B may also have the desired effect. This is intuitive when considering that in order for more objects to be classified into $POS(A)$, we need to lower the risk involved in classifying an object into this region.

We see that in order to decrease the value of R_P, we need to decrease one or both of the loss functions λ_{PP} and λ_{PN} (Equation 1: R_P). Likewise, to increase

Table 1. The strategy scenario of increasing approximation accuracy.

Action (Strategy)	Goal	Method	Result
a_1 $(-R_P)$	Decrease R_P	Decrease λ_{PP} or λ_{PN}	Larger POS region
a_2 $(+R_N)$	Increase R_N	Increase λ_{NP} or λ_{NN}	Smaller NEG region
a_3 $(+R_B)$	Increase R_B	Increase λ_{BP} or λ_{BN}	Smaller BND region

R_N, we need to increase either λ_{NP} or λ_{NN}. Finally, to increase R_B, we need to increase λ_{BP} or λ_{BN}. This is summarized in Table 1.

We want to increase approximation precision when considering the second measure, ψ. For the deterministic case, in order to increase precision, we need to make $|\underline{apr}(A)|$ as large as possible. Again, recalling rules (PN, NN, BN), we see that in order to increase the size of the lower approximation, we need to decrease the expected loss R_P and to increase R_N and R_B. It has the same strategy set as the first player because we wish to increase the size of the lower approximation.

Of course, there may be some tradeoff between the measures ϕ and ψ. An increase in one will not have a similar increase in the other. This implies some form of conflict between these measures. We can now use game theory to dictate the increases/decreases in conditional risk for region classification and as a method for governing the changes needed for the loss functions.

3.2 Game-Theoretic Specification

Game theory [9] 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 [8, 11], networking [1], and machine learning [5].

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,

$$G = \{O, S, F\}, \tag{2}$$

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. With improved classification ability as the competition goal, each player can represent a certain measure such as accuracy (ϕ) and precision (ψ). With this in mind, a set of players is formulated as $O = \{\phi, \psi\}$. 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 measure is effectively competing with the other to win the "game". Here, the game is to improve classification ability. To compete, each 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 $S_i = \{a_1, \ldots, a_m\}$ 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 ψ would be to "acquire a maximal value for approximation precision as possible".

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 (ψ), 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 [13] suggested an alternative precision measure.

In general, the two measures ϕ and ψ 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.

3.3 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, $F = \{\mu_\phi, \mu_\psi\}$, 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 $\mu_\psi : S_2 \mapsto P_2$ are used to derive the payoffs for ϕ and ψ containing:

$$P_1 = \{\phi_{1,1}, \phi_{1,2}, \phi_{1,3}\}, \tag{3}$$

$$P_2 = \{\psi_{2,1}, \psi_{2,2}, \psi_{2,3}\}, \tag{4}$$

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 μ_ψ 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 [2].

Table 2. Payoff table for ϕ, ψ payoff calculation (deterministic).

		ψ		
		$-R_P$	$+R_N$	$+R_B$
ϕ	$-R_P$	$< \phi_{1,1}, \psi_{1,1} >$	$< \phi_{1,2}, \psi_{1,2} >$	$< \phi_{1,3}, \psi_{1,3} >$
	$+R_N$	$< \phi_{2,1}, \psi_{2,1} >$	$< \phi_{2,2}, \psi_{2,2} >$	$< \phi_{2,3}, \psi_{2,3} >$
	$+R_B$	$< \phi_{3,1}, \psi_{3,1} >$	$< \phi_{3,2}, \psi_{3,2} >$	$< \phi_{3,3}, \psi_{3,3} >$

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.

3.4 Payoff Tables and Equilibrium

To find optimal solutions for ϕ and ψ, we organize payoffs with the corresponding actions that are performed. A payoff table is shown in Table 2, and will be the focus of our attention.

The actions belonging to ϕ are shown row-wise whereas the strategy set belonging to ψ are column-wise. In Table 2, 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 ψ contains the same actions for the second player.

Each cell in the table has a payoff pair $< \phi_{1,i}, \psi_{2,j} >$. A total of 9 payoff pairs are calculated. For example, the payoff pair $< \phi_{3,1}, \psi_{3,1} >$ containing payoffs $\phi_{3,1}$ and $\psi_{3,1}$ 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 [3]. 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 $< \phi^*_{1,i}, \psi^*_{2,j} >$ is an equilibrium if for any action a_k, where $k \neq i, j$, $\phi^*_{1,i} \geq \phi_{1,k}$ and $\psi^*_{2,j} \geq \psi_{2,k}$. The $< \phi^*_{1,i}, \psi^*_{2,j} >$ 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.

3.5 Loss Tolerance Calculation

Observed from decision rules (PN, NN, BN), we can calculate how much the loss functions need to be modified to acquire a certain level of accuracy or precision. 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. However, the rules (NN) and (BN) are also sensitive to the modification of R_P, denoted R_P^*. R_P^* must satisfy $R_P^* \geq (R_P - R_N)$ and $R_P^* \geq (R_P - R_B)$. This results in upper limit of t_{PP}^{max} for λ_{PP} and lower limit of t_{PN}^{min} for λ_{PN}. Assuming that $\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}$, we calculate the following,

$$t_{PP}^{max} \leq \frac{\lambda_{BP} - \lambda_{PP}}{\lambda_{PP}} \quad , \qquad t_{PN}^{min} < \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{PN}}. \tag{5}$$

That is, t_{PP} is the *tolerance* that loss function λ_{PP} can have (t_{PP} 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 $< \phi_{1,i}^*, \psi_{2,j}^* >$. In brief, when selecting a strategy, i.e., $(+R_P)$, 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} for λ_{PP} and t_{PN} for λ_{PN}.

For example, let $\lambda_{PP} = \lambda_{NN} = 4$, $\lambda_{BP} = \lambda_{BN} = 6$, and $\lambda_{PN} = \lambda_{NP} = 8$. The inequality restrictions for the loss functions hold. We calculate that $t_{PP}^{max} = 0.5$ and $t_{PN}^{min} = 0.25^{\dagger}$. This means that we can increase the loss function λ_{PP} by 50% and increase the loss function λ_{PN} by 25%[†] and maintain the same classification ability. This new information was derived from the analysis of the conditional risk modifications made possible through the use of game theory.

4 Conclusions

We provide a preliminary study on using game theory for determining the relationships between loss function tolerance and conditional risk. By choosing measures of approximation accuracy and approximation precision as players in a game, with goals of maximizing their values, we set up a set of strategies that each can perform. We investigate the use of three strategies for the deterministic approximation case. The strategies involve decreasing or increasing the expected losses for classifying objects into rough set regions.

[†] An incorrect value of -0.125 for t_{PN}^{min} was written in the original publication. A decrease of 12.5% has been changed to an increase of 25%

Ultimately, taking an action within the strategy set involves modifying user-provided loss functions. We provide a method for indicating how much a loss function can be modified in order to provide optimal approximation accuracy and precision. This is very useful for the users as determining the amount of tolerance they should have when modifying loss functions is difficult.

By finding an equilibrium in the payoff tables, we may find the correct values for the loss functions, and thus, the optimal values of α and β parameters for determining the region boundaries. Based on this, we express the consequences of an increased or decreased expected loss of classification with the approximation accuracy and precision measures.

References

1. Bell, M.G.F.: The use of game theory to measure the vulnerability of stochastic networks. IEEE Transactions on Reliability **52** (2003) 63–68
2. Duntsch, I., Gediga, G.: Uncertainty measures of rough set prediction. Artificial Intelligence **106** (1998) 109–137
3. Fudenberg, D., Tirole, J.: Game Theory. The MIT Press (1991)
4. Gediga, G., Duntsch, I.: Rough approximation quality revisited. Artificial Intelligence **132** (2001) 219–234
5. Herbert, J., Yao, J.T.: A game-theoretic approach to competitive learning in self-organizing maps. In: International Conference on Natural Computation (ICNC'05), LNCS. Volume 1 of 3610. (2005) 129–138
6. Herbert, J., Yao, J.T.: Time-series data analysis with rough sets. In: Proceedings of Computational Intelligence in Economics and Finance (CIEF'05). (2005) 908–911
7. Herbert, J.P., Yao, J.T.: Rough set model selection for practical decision making. In: Proceeding of Fuzzy Systems and Knowledge Discovery (FSKD'07). III (2007) 203–207
8. Nash, J.: The bargaining problem. Econometrica **18** (1950) 155–162
9. Neumann, J.V., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
10. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences **11** (1982) 341–356
11. 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
12. Yao, J.T., Herbert, J.P.: Web-based support systems with rough set analysis. In: Proceedings of Rough Sets and Emerging Intelligent Systems Paradigms (RSEISP'07), LNAI. 4585 (2007) 360–370
13. Yao, Y.Y.: Probabilistic approaches to rough sets. Expert Systems **20** (2003) 287–297
14. Yao, Y.Y.: Decision-theoretic rough set models. In: Proceedings of Rough Sets Knowledge Technology (RSKT'07), LNAI. 4481 (2007) 1–12
15. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. International Journal of Man-machine Studies **37** (1992) 793–809
16. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In Ras, Z.W., Zemankova, M., Emrich, M.L., eds.: Methodologies for Intelligent Systems. Volume 5. North-Holland, New York (1990) 17–24
17. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences **46** (1993) 39–59