

Learning Optimal Parameters in Decision-Theoretic Rough Sets

Joseph P. Herbert JingTao Yao

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

Abstract. A game-theoretic approach for learning optimal parameter values for probabilistic rough set regions is presented. The parameters can be used to define approximation regions in a probabilistic decision space. New values for loss functions are learned from a sequence of risk modifications derived from game-theoretic analysis of the relationship between two classification measures. Using game theory to maximize these measures results in a learning method to reformulate the loss functions. The decision-theoretic rough set model acquires initial values for these parameters through a combination of loss functions provided by the user. The new game-theoretic learning method modifies these loss functions according to an acceptable threshold.

1 Introduction

Rough set theory is a method of approximating uncertain information in data [9]. However, the traditional model is often too strict when including objects into the approximation regions and may require additional information [6], or require several approximations [1]. A key tenet in the introduction of probabilistic rough sets was to loosen the extreme membership requirements of equivalence classes to the object set found in conventional rough sets [12]. The decision-theoretic rough set (DTRS) model is a powerful solution to this problem of decreasing the boundary region through the use of conditional risk of a classification action [14]. However, it still remains that the loss functions used to calculate the region parameters are provided by the user [5, 15].

In general, probabilistic rough set approaches require that parameter values be provided from an external user, which are then used to relax the acceptance of uncertainty. The use of machine learning techniques to learn parameter values from the evaluation of a partial solution would be a beneficial addition. Game-theoretic learning is an approach that uses game theory to govern the optimization of one or more characteristics of a soft computing technique [3].

Preliminary studies of game-theoretic rough sets (GTRS) overcomes these challenges by interpreting classification ability as a decision problem within a game [5, 10]. Utilizing game theory, we can provide a sequence of risk modifications that find correct loss function values for an optimal classification ability. In this article, we provide a general approach for using a game-theoretic learning

method to govern the modification of loss functions in order to improve these measures. In addition, we will detail some side-effects that arise from using this approach repeatedly. A learning method is formulated and explained.

2 Decision-Theoretic and Game-Theoretic Rough Sets

2.1 Decision-Theoretic Rough Sets

A key contribution of the decision-theoretic rough sets is to considering costs associated when classifying an equivalence class into a particular region [7]. Let $\lambda_{\diamond P} = \lambda(a_{\diamond}|A)$ denote the loss incurred for taking action a_{\diamond} when an object is in A , and $\lambda_{\diamond N} = \lambda(a_{\diamond}|A^c)$ denote the loss incurred by taking the same action when the object belongs to A^c , where $\diamond = P, N$, or B and a_P, a_N , and a_B represent the three actions to classify an object into $POS(A)$, $NEG(A)$, and $BND(A)$ respectively. The expected loss $R(a_{\diamond}|[x])$ associated with taking the individual actions is expressed as:

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

where $P(a_{\diamond}|[x])$ be the conditional probability of an object x being in state a_{\diamond} given the description of $[x]$. The Bayesian decision procedure leads to minimum-risk decision rules (PN-BN):

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

The probability for classifying an element into positive region, α , and the probability for putting an element into boundary region, β , can be derived using loss functions [14],

$$\alpha = \frac{\lambda_{PN} - \lambda_{BN}}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}, \quad \beta = \frac{\lambda_{BN} - \lambda_{NN}}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}. \quad (2)$$

Various configurations of loss functions results in the DTRS model having the same functionality of other intermediate approaches [13]. In addition, two fundamental questions need to be answered regarding the use of loss functions in the DTRS model. First, given a set of initial values for all loss functions, how much may they change in order to achieve similar classification ability? Second, can a method be proposed to learn optimal values for all loss functions based only upon the analysis of the data? An extension using game theory was proposed [5, 10] to answer the first question, which is reviewed below.

Table 1. The strategy scenario of increasing approximation accuracy [5].

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

2.2 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 [8]. We proposed a game-theoretic rough set model [5]. In particular, a game is formulated in such a way that each player represents a classification measure, such as approximation accuracy (ϕ) and approximation precision (ψ). The actions these players choose are summarized in Table 1. They either increase or decrease the conditional risk by modifying the associated loss functions in (1). 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, $F = \{\mu_\phi, \mu_\psi\}$, showing payoff functions that measure the increase in accuracy and precision respectively [5].

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 $\mu_\psi : S_2 \mapsto P_2$ are used to derive the payoffs for ϕ and ψ containing:

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

$$P_2 = \{\psi_{2,1}, \psi_{2,2}, \psi_{2,3}\}, \quad (4)$$

reflecting payoffs from the results of the three actions, i.e., $\mu_\phi(a_j) = \phi_{1,j}$ [10]. 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 be seen in Table 2. 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_ψ 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 [2].

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

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

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

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 2. If one is interested in maximizing accuracy, all needed is to choose a suitable action that fits with his or her acceptable risk.

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 $R_P^* \geq (R_N - R_P)$ 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} [5]. Assuming that $\lambda_{PP} \leq \lambda_{BP} < \lambda_{NP}$ and $\lambda_{NN} \leq \lambda_{BN} < \lambda_{PN}$, we calculate the following,

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

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 $\langle \phi_{1,i}^*, \psi_{2,j}^* \rangle$. 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} [10].

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 measure.

3 Learning Optimal Parameters

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 [4]. 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 General Approach

We define a general approach for using game theory in conjunction with the DTRS model to aid in decision making [10]. There are five steps to be performed:

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.

To arrive at the stage where loss functions are modified to optimize a given measure, we must continually repeat Steps 4 and 5. Once result acquisition occurs (meaning that a suitable action has been chosen and recorded), we must repopulate the payoff table to reflect new payoffs that will arise if more actions are undertaken. The actions that each measure may undertake remain the same.

This repetition continues until the next action that should be performed exceeds the acceptable levels of tolerance a user may have. R_P^* must satisfy $R_P^* \geq (R_N - R_P)$ and $R_P^* \geq (R_B - R_P)$. That is, the new value of the respective loss function, denoted by *, must remain satisfied:

$$0 < \lambda_{PP}^* \leq \frac{\lambda_{BP} - \lambda_{PP}}{\lambda_{PP}} \quad , \quad 0 < \lambda_{PN}^* < \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{PN}}. \quad (6)$$

3.2 Repetitive Risk Modifications

Repeatedly modifying the conditional risk changes the sizes of the regions. Referring to Table 1, choosing action $\downarrow R_P$ (decreasing R_P) will increase the size of the positive region. 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.

Using New Zealand Stock Exchange data [11], Table 3 summarizes new region sizes (in total universe percentage) that result in changing loss functions given

Table 3. Region sizes with changes in conditional risk.

Strategy	Method	α	β	Region Size (% of universe)		
				<i>POS</i>	<i>NEG</i>	<i>BND</i>
$\downarrow R_P$	Decrease λ_{PP}	0.75	0.25	23%	37%	40%
		0.72	0.25	33%	37%	30%
		0.60	0.25	48%	37%	15%
$\uparrow R_N$	Increase λ_{NP}	0.75	0.25	23%	37%	40%
		0.75	0.20	23%	34%	43%
		0.75	0.10	23%	13%	64%
$\uparrow R_B$	Increase λ_{BP}	0.75	0.25	23%	37%	40%
		0.67	0.29	43%	37%	20%
		0.60	0.33	48%	43%	9%

a chosen strategy. As we decrease λ_{PP} , we see the parameter α decrease. This 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.

When increasing or decreasing a region size, the remaining two regions are decreased or increased 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. Essentially, a decreased risk, or cost, 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_{PP} is lowered, λ_{BP} and λ_{NP} will increase to some extent.

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.

3.3 The Parameter Learning Sequence

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

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

$$\lambda_{\diamond P}^* = \lambda_{\diamond P} \pm (\lambda_{\diamond P} \cdot P(A|[x]) \cdot (\mu(\Gamma) - \mu(a_i))), \quad (7)$$

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 1, 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:

$$\lambda_{\diamond N}^* = \lambda_{\diamond N} \pm (\lambda_{\diamond N} \cdot P(A^c|[x]) \cdot (\mu(\Gamma) - \mu(a_i))). \quad (8)$$

From Table 1, we could decrease either λ_{PP} or λ_{PN} . To accomplish this, we would solve either of the following two equations:

$$\lambda_{PP}^* = \lambda_{PP} + (\lambda_{PP} \cdot P(A|[x]) \cdot (\mu(\Gamma) - \mu(a_i))), \quad (9)$$

$$\lambda_{PN}^* = \lambda_{PN} + (\lambda_{PN} \cdot P(A^c|[x]) \cdot (\mu(\Gamma) - \mu(a_i))). \quad (10)$$

$P(A^c|[x])$ is used since it has a loss function that measures the cost of classifying an object into a set's complement. For example, let $\lambda_{PP} = 2$, $P(A^c|[x]) = 0.7$, $\Gamma = \phi$, $\mu(\phi) = 0.5$, and $\mu(a_1) = 0.55$. Using (9), we acquire a new loss function $\lambda_{PP}^* = 1.93$, or a decrease of 3.5%. This procedure can be repeated by repopulating the payoff table and choosing a suitable action to increase ϕ , given that the loss functions remain within the tolerance defined above.

4 Conclusions

We provide a parameter learning method using game theory that defines loss functions according to an optimal value of classification ability. Measures of classification ability, such as approximation accuracy and approximation precision, are given as players in a game - each with a goal of optimizing their values. Actions performed in this game consist of increasing or decreasing the size of the classification regions. This is achieved by modifying the values of 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.

The game-theoretic rough set model (GTRS), further studied in this article, overcomes some challenges presented by the decision-theoretic model. Loss functions used to calculate region parameters are no longer needed to be provided, as they can be derived from the data with game-theoretic analysis. Game theory shows great potential in solving decision problems such as those seen in parameter estimation. It allows for the observation of many possible outcomes of taking actions in improving classification ability. We believe that game theory should be used to analyze the effects of modifying the users' notions of risk, as it is fully capable of deriving a learning sequence in order to achieve a classification goal.

Many new avenues of research are opened up by the GTRS model. By interpreting fundamental notions of rough sets with game theory, additional insights into the competition and cooperation between measures, parameters, and risk can be observed. For example, the scope of the game does not necessarily have to be limited to approximation accuracy and precision. Other classification measures can be used to provide other insights into the performance of classification. New directions in the choice of players could be considered. As well, cooperative games could be formulated if we wish to achieve balance between classification measures.

References

1. Ciucci, D.: A unifying abstract approach for rough models. In: Proceedings of Rough Sets and Knowledge Technology (RSKT'08), LNAI. 5009 (2008) 371–378
2. Duntsch, I., Gediga, G.: Uncertainty measures of rough set prediction. *Artificial Intelligence* **106** (1998) 109–137
3. Herbert, J.P., Yao, J.T.: A game-theoretic approach to competitive learning in self-organizing maps. In: International Conference on Natural Computation (ICNC'05), LNCS. 3610 (2005) 129–138
4. Herbert, J.P., Yao, J.T.: Rough set model selection for practical decision making. In: Proceedings of Fuzzy Systems and Knowledge Discovery (FSKD'07). Vol. III (2007) 203–207
5. Herbert, J.P., Yao, J.T.: Game-theoretic risk analysis in decision-theoretic rough sets. In: Proceedings of Rough Sets and Knowledge Technology (RSKT'08), LNAI. 5009 (2008) 132–139
6. 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** (2007) 485–494
7. Lingras, P., Chen, M., Miao, D.: Rough multi-category decision theoretic framework. In: Proceedings of Rough Sets and Knowledge Technology (RSKT'08), LNAI. 5009 (2008) 676–683
8. Neumann, J.V., Morgenstern, O.: *Theory of Games and Economic Behavior*. Princeton University Press, Princeton (1944)
9. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. *Information Sciences* **177** (2007) 41–73
10. Yao, J.T., Herbert, J.P.: A game-theoretic perspective on rough set analysis. *Journal of Chongqing University of Posts and Telecommunications* **20** (2008) 291–298
11. Yao, J.T., Herbert, J.P.: Financial time-series analysis with rough sets. *Applied Soft Computing* (2009) (In Press) DOI: 10.1016/j.asoc.2009.01.003.
12. Yao, Y.Y.: Probabilistic approaches to rough sets. *Expert Systems* **20** (2003) 287–297
13. Yao, Y.Y.: Decision-theoretic rough set models. In: Proceedings of Rough Sets Knowledge Technology (RSKT'07), LNAI. 4481 (2007) 1–12
14. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. *International Journal of Man-machine Studies* **37** (1992) 793–809
15. Zhou, X., Li, H.: A multi-view decision model based on decision-theoretic rough set. In: Proceedings of Rough Sets and Knowledge Technology (RSKT'09), LNAI. 5589 (2009)