https://doi.org/10.48009/3_iis_2019_117-127

Issues in Information Systems
Volume 20, Issue 3, pp. 117-127, 2019

USING DATA MINING TECHNIQUES FOR GAME RULE EXTRACTION

Chen-Huei Chou, College of Charleston, chouc@cofc.edu

ABSTRACT

Tic-Tac-Toe game is an easy to play two-player game played on a three by three grid. "Three in a row" is the rule to win a game. After watching a number of games, human players may be able to figure out the way to find the winner of finished games without knowing the rules. Data mining techniques have been applied to solve various problems. Machine classifiers can be used to deal with classification issues such as credit evaluation. Association rule mining is used to find significant relationships among items such as Market Basket Analysis. This study is to propose the use of both classification, specifically decision trees, and association rule techniques to find the rules of Tic-Tac-Toe games. Comparisons on results of the two different data mining techniques are made. The experimental results show that decision tree ID3 and association rule miner Predictive Apriori can identify 100% of game rules while decision tree generates more duplicating rules and it is harder to extract the rules. Although high accuracy rate was reached, Functional Tree utilizing logistic regression for nodes extracted no usable game rules.

Keywords: Tic-Tac-Toe Game, Game Rule, Data Mining, Classification, Association Rule

INTRODUCTION

Tic-Tac-Toe game is a popular two-player game played on a three by three grid. The player who can place the own marks three-in-a-row wins the game. Human beings may be able to figure out this rule by watching the games being played. Once the rule is obtained, it is not an issue to determine the winner of a game. How about machine learners? To judge the winner of the game requires the marks on the grid and follows the rule to find out the winner, either X or O.

Through a supervised learning process, a machine learning algorithm can build a classification model (classifier) from a training dataset. Once the classifier is built, the machine classifier can predict the outcome. Judging the winner of a Tic-Tac-Toe game is an example of classification. Classification techniques have been successfully applied to assess risks of prostate cancer patients (Churilov et al. 2005), profile Web usage in the workplace (Anandarajan 2002), solve problems such as credit evaluation (Sinha and May 2005), forecast financial performance (Walczak 2001), generate document taxonomies (Spangler et al. 2003), classify open source software development projects (Vlas and Robinson 2012), classify web queries (Beitzel et al. 2007), and social media opinion mining (Kang et al. 2018). Among the well-known classifiers, decision tree classifiers can generate a human interpretable tree. In this study, the first objective is to examine whether the decision tree classifiers can successfully generate valid rules for judging the winner of a Tic-Tac-Toe game.

Association rule miners can identify frequent patterns from a given dataset. This method has also been successfully applied to a number of areas including breast cancer detection (Karabatak and Ince 2009), concept design of convergent products (Lee et al. 2012), electronic catalog marketing and sales management (Liao and Chen 2004), intrusion detection to protect computer systems (Tajbakhsh et al. 2009), and forecasting changes in Stock Price Index (Na and Sohn 2011). The second objective of the study is to examine whether the association rule miners are able to find frequent associated game variables as rules to determine the winner. Specifically, the study attempts to answer the following research questions:

- *What are the human interpretable rules generated from the decision tree algorithms?*
- *What are the rules mined from association rule analysis?*
- *Can both decision tree classifier and association rule miner identify all valid rules judging the winner of a Tic-Tac-Toe game?*

The rest of the paper is organized as follows. Relevant background is discussed in the next section. Experimental methods are then described. Findings of experiments are reported. Finally, conclusions are drawn and future directions are highlighted.

LITERATURE REVIEW

Tic-Tac-Toe Game

Zaslavsky (1982) discovered that the popular two-player Tic-Tac-Toe game is similar to games played in ancient Egypt. The Tic-Tac-Toe, also called noughts and crosses, is originally designed as a paper and pencil game. The two players take turns marking down their own Xs or Os in a three by three grid. A player who successfully places three marks in a row, a column, or a diagonal wins the game. Thus, the rule to win the game is "three-in-a-row." Figure 1 shows a game won by X due to three X in the diagonal row. There are 255,168 possible games in total. Among the available games, 958 terminal configurations are reached when a winner is found (Schaeffer 2002). The terminal configuration could be a game with a determined winner without using all nine cells such as the game in Figure 1 or with all cells marked.

X	O	X
O	X	O
	O	X

Figure 1. A Tic-Tac-Toe won by X

Classification

Classification is one of the well-known data mining techniques. It is a supervised learning process to build a model for future predictions (classifications). The model, generated from training data, is to associate characteristics of data observations with a desired category. The observations in the training data consist of a set of characteristics (input variables) and a given category (output variable). Tic-Tac-Toe game is a typical binary classification problem. In classifying Tic-Tac-Toe games, there are nine positional input variables and a given winner output variable. Each input variable can carry an "X", an "O", or nothing (blank). The binary winner output variable can be either "X" or "O". The training dataset is a set of games $\{g_1, g_2, \ldots, g_j\}$ with their output category $\{c_1, c_2\}$. A machine learning algorithm, a classifier, is used to find the model $c_i = f(g_j)$ from correct pairs of $<g_j, c_i>$ consistently, where game g_j is represented by nine positional variables. Once the classifier is built, a correct prediction is made when a game g_j can be assigned to the correct winner category c_i, either "X" or "O".

Decision tree, one of the machine classifiers, can generate human-interpretable decision-making rules. Following the decisions through branches from the top of the tree, a final classification decision can be reached in the leaf node. Popular decision tree machine learning methods include ID3 (Quinlan 1986), C4.5 (Quinlan 1993), Best First Tree (Friedman et al. 2000; Shi 2007), and Functional Tree (Gama 2004). ID3 (Iterative Dichotomiser 3) algorithm is used to generate a decision tree from a given dataset in a top-down manner. Entropy of each attribute is calculated. The attribute with the minimum entropy is assigned as the root. Iterations continue from this root node to further split into child nodes using the unused attribute with minimum entropy. C4.5 is a successor of ID3. Instead of using entropy, this algorithm uses normalized information gain. It handles both continuous and discrete attributes. It also allows missing values in attributes. After the creation of the decision tree, pruning tree is an option in C4.5. Best-First Tree algorithm builds the same tree as a standard depth-first traversal trees, while the best split happens on those maximally reducing the Gini index. Using constructive induction, the functional tree algorithm integrates a univariate decision tree and a linear function to build a tree. It takes multivariate tests on nodes to derive predictions using linear functions.

Association Rule

Association analysis can be used to mine frequent patterns, associations, and correlations from a given data set. Association rule mining is one of the data mining techniques used to find significant relationships among items in a data set, without the associated output category. Such technique is widely used to find associated items in shopping carts called Market Basket Analysis (Agrawal et al. 1993).

Association rule miner is to find frequent patterns (item sets) from a dataset *D*. The dataset *D* contains transactions *T* where each transaction is an item set of $I = \{i_1, i_2, \ldots, i_m\}$, that is $T \subseteq I$. Following a two-step approach, the association rule algorithm first determines the large item set and then generates rules based on given thresholds. Support and confidence are commonly used measures. Support is to evaluate the statistical importance of the item set *X* in the dataset *D*, denoted as *Sup(X,D)*. It shows the rate of item set *X* that appears in the dataset *D*. High support of *X* also indicates its higher importance. The item set *X* is considered frequent if *Sup(X,D)* is greater than a given threshold. In the rule $X \rightarrow Y$ generated from the miner, confidence measure (denoted as $Conf(X \rightarrow Y)$) is to evaluate the level of association between item sets *X* and *Y*, where $X \subseteq I$ and $Y \subseteq I$. It represents the rate of transactions *T* that contains *X* in the dataset *D* also contains *Y*. Therefore, it is estimated as follows: $Conf(X \rightarrow Y) = Sup(X \cup Y, D) / Sup(X, D)$. Apriori (Agrawal 1993) and Predictive Apriori (Scheffer 2001) are two well-known association rule algorithms used in practical applications.

EVALUATION METHODS

Data Collection and Data Preparation

In a Tic-Tac-Toe game, two players take turns placing their own mark on a three by three grid. The nine cells in the grid can be described by nine positions. They are upper-left (UL), upper-middle (UM), upper-right (UR), middle-left (ML), middle-middle (MM), middle-right (MR), lower-left (LL), lower-middle (LM), and lower-right (LR). Assuming player X goes first, Schaeffer (2002) identified that 958 terminal configurations are reached when a winner is found. These 958 games are used by machine classifiers and association rule miners in this study. Table 1 lists 10 games (observations) out of the 958 games. For each game, 10 variables are used to describe the game details. The variables consist of nine positional variables and a Winner variable. Notation "x" is used for positional variables if the player X marks on the cells. Similarly, notation "o" is used if the player O marks on the cells. When a cell is left blank without any mark, "b" is applied. For the Winner variable, it could be "x" if the player X gets three marks in a row. "o" is stored if the play O is the winner.

Out of the 958 terminal games, player X is the winner of 626 games and player O is the winner of 332 games.

Table 1. Sample observations of Tic-Tac Toe games

UL	UM	UR	ML	MM	MR	LL	LM	LR	Winner
x	o	x	o	x	o	o	x	x	x
x	o	x	o	x	o	b	b	x	x
x	o	x	o	x	b	x	o	b	x
x	o	x	o	x	b	x	b	o	x
x	o	x	o	x	b	o	b	x	x
x	o	x	x	o	x	o	o	b	o
x	o	x	x	o	x	b	o	o	o
x	o	x	x	o	o	b	o	x	o
x	o	x	x	o	b	o	o	x	o
x	o	x	x	o	b	b	o	b	o
x	o	x	x	o	x	o	o	b	o

Experimental Design

Four decision tree machine learning methods: ID3 (Quinlan 1986), C4.5 (Quinlan 1993), Best First Tree (Friedman et al. 2000; Shi 2007), Functional Tree (Gama 2004) and two association rule miners: Apriori (Agrawal 1993) and Predictive Apriori (Scheffer 2001) were used for evaluations using Weka (Witten and Frank 2005) tool. Two experiments were conducted in order to answer the research questions. The main purpose of the first experiment is to explore the valid game rules from built decision trees. By traversing the decision nodes from the root to the leaf, if the combination of decisions on splits along with the correct winner classification is made, such combination is treated as a valid game rule extracted from the decision tree. The main purpose of the second experiment is to apply association rule algorithms to find frequent combination of positional marks along with the winner. If such combination of positional marks and the winner matches a valid game rule, it is considered a correct extraction. The two experiments are conducted using Weka tool. It is an open source data mining tool providing a good collection of well-known and modern machine learning algorithms. The abovementioned decision tree and association rule algorithms are available in Weka and have been tested by many published works for different data mining tasks.

Design of Experiment 1

The first experiment is designed to examine Tic-Tac-Toe game rules through decision tree machine classifiers. ID3, C4.5, Best First, and Functional Tree decision trees are used to learn the 958 terminal games with given winners. Once the decision trees are created, rule extraction is proceeded by visiting the tree nodes from the root to the leaves, using breadth first approach. The decisions made by decision trees are treated invalid if 1) incorrect winner is identified; 2) same player's marks do not form three-in-a-row.

Design of Experiment 2

In the second experiment, the Apriori and Predictive Apriori association rule miners take the 958 terminal games and examine the highly associated attributes. The source of attributes includes nine positional Tic-Tac-Toe game cells and a winner attribute. A valid rule is extracted when three positional marked cells form a row and a correct winner is associated. A rule mined is considered invalid if 1) winner is missing; 2) more than three positional cells are used; 3) less than three positional cells are used; 4) three positional cells do not form a row; 5) incorrect winner is identified.

Experimental Environment

Weka's default parameters were kept for the decision tree algorithms. The J4.8 decision tree method was Weka's implementation of C4.5. All experiments were conducted using a personal computer equipped with Intel Core i5 3.3GB CPU, 16GB RAM, and Microsoft Windows 10 64bit edition.

EXPERIMENTAL RESULTS

Tic-Tac-Toe Rules

To win a Tic-Tac-Toe game, the player, either X or O, needs to place own marks three-in-a-row in a horizontal row, a vertical column, or a diagonal. Figure 2 shows the eight possible rules to win a game. This same set of rules applies to both players. Thus, a total of 16 rules can be used to judge a game being played. To simplify the rules, a notation Rnp is used for each rule, where n is the nth rule and p is the player. For example, the rule R1o can demonstrate the winner O who placed marks in the upper-left, upper-middle, and upper-right cells. Similarly, player X can win a game by placing marks in the upper-right, middle-middle, and lower-left cells, judged by the rule R7x.

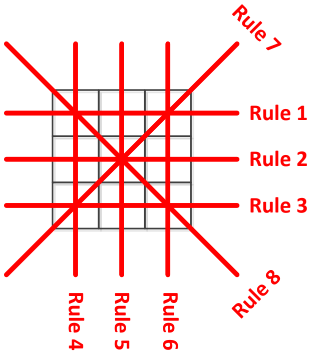
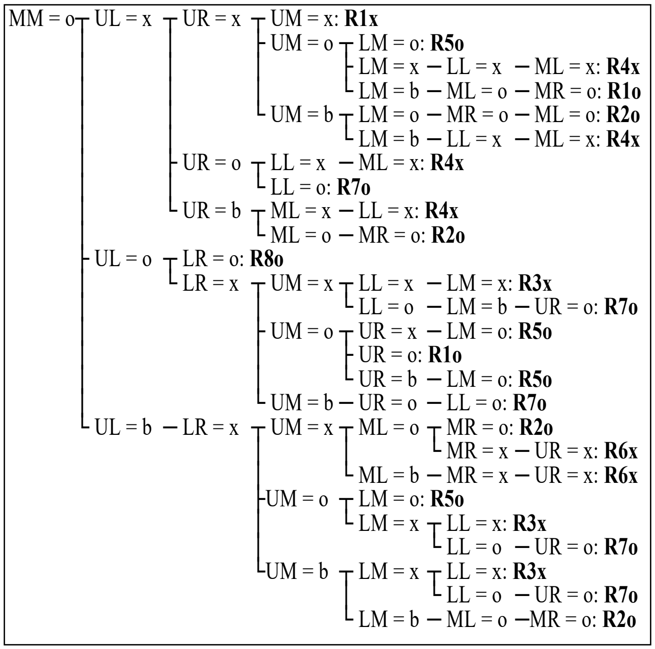


Figure 2. Rules of a Tic-Tac-Toe game

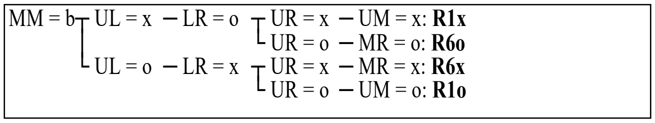
Decisions Made through ID3 Tree

Section 4.2, 4.3, 4.4, and 4.5 report the results of experiment 1 using ID3, C4.5, Best First, and Functional Tree decision trees, respectively. When using the 958 finished games to train the ID3 decision tree classifier and applying the whole set of games as test set for classification evaluation, the ID3 trained classifier reached 100% accuracy rate. The generated decision tree contained 251 leaves, meaning 251 decisions can be made. Although all games can be correctly classified to the correct winner, not all decisions were rule related. After cleansing the unrelated decisions based on the 16 rules, a total of 56 decisions were left. As a result, ID3 decision tree generated 22.31% of useable decisions. Figure 3 shows 16 decisions in the branch of ID3 decision tree when the middle cell is marked “o”. Four decisions were found in the branch when the central cell is left blank (Figure 4). Another 16 decisions listed in Figure 5 were the ones in the branch of ID3 tree when player X took the middle-middle cell. Following a designated path, a winner can be found. Based on the 16 rules identified earlier, the corresponding game rule suitable for judging the winner of a path was listed after the decision. For instance, player X is the winner when placing the mark in the middle-middle, lower-right, and upper-left cells. The corresponding rule is R8x. The designated path in Figure 5 is MM =x – LR= x – UL = x: R8x.



```
MM = o┬ UL = x ┬ UR = x ┬ UM = x: R1x
      │        │        ├ UM = o ┬ LM = o: R5o
      │        │        │        ├ LM = x ─ LL = x ─ ML = x: R4x
      │        │        │        └ LM = b ─ ML = o ─ MR = o: R1o
      │        │        └ UM = b ┬ LM = o ─ MR = o ─ ML = o: R2o
      │        │                 └ LM = b ─ LL = x ─ ML = x: R4x
      │        ├ UR = o ┬ LL = x ─ ML = x: R4x
      │        │        └ LL = o: R7o
      │        └ UR = b ┬ ML = x ─ LL = x: R4x
      │                 └ ML = o ─ MR = o: R2o
      ├ UL = o ┬ LR = o: R8o
      │        └ LR = x ┬ UM = x ┬ LL = x ─ LM = x: R3x
      │                 │        └ LL = o ─ LM = b ─ UR = o: R7o
      │                 ├ UM = o ┬ UR = x ─ LM = o: R5o
      │                 │        ├ UR = o: R1o
      │                 │        └ UR = b ─ LM = o: R5o
      │                 └ UM = b ─ UR = o ─ LL = o: R7o
      └ UL = b ─ LR = x ┬ UM = x ┬ ML = o ┬ MR = o: R2o
                        │        │        └ MR = x ─ UR = x: R6x
                        │        └ ML = b ─ MR = x ─ UR = x: R6x
                        ├ UM = o ┬ LM = o: R5o
                        │        └ LM = x ┬ LL = x: R3x
                        │                 └ LL = o ─ UR = o: R7o
                        └ UM = b ┬ LM = x ┬ LL = x: R3x
                                 │        └ LL = o ─ UR = o: R7o
                                 └ LM = b ─ ML = o ─MR = o: R2o
```

Figure 3. A branch of ID3 decision tree when the middle cell is marked by player O



```
MM = b┬ UL = x ─ LR = o ┬ UR = x ─ UM = x: R1x
      │                 └ UR = o ─ MR = o: R6o
      └ UL = o ─ LR = x ┬ UR = x ─ MR = x: R6x
                        └ UR = o ─ UM = o: R1o
```

Figure 4. A branch of ID3 decision tree when the middle cell is not marked

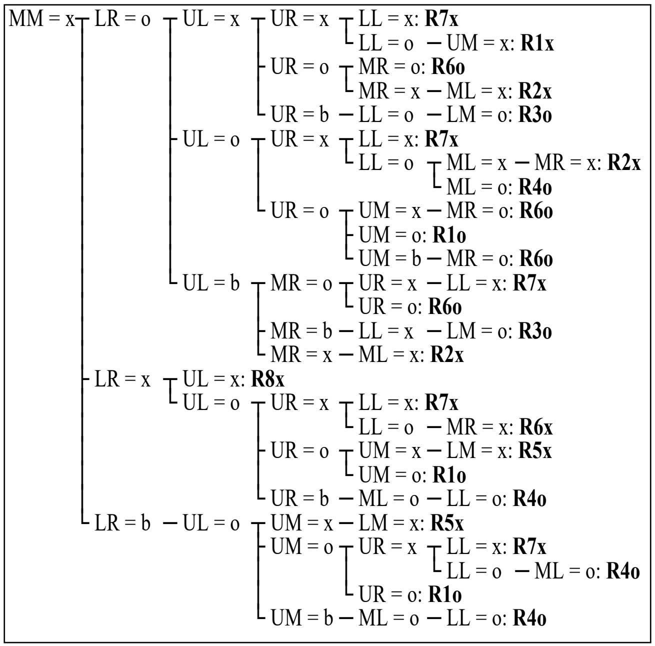


Figure 5. A branch of ID3 decision tree when the middle cell is marked by player X

Although paths in the three branches look different, some determined the same winner using the same rule. Summarizing the number of rules from Figures 3-5, the frequencies of identified rules were listed in Table 2. ID3 decision tree had 100% coverage of the 16 rules although redundant decision paths were made.

Table 2. Frequency of rules generated from decision trees

Rule	ID3	C4.5	BF	FT
R1o	6	3	1	0
R2o	4	1	1	0
R3o	2	2	3	0
R4o	4	1	1	0
R5o	4	2	1	0
R6o	5	3	1	0
R7o	5	1	0	0
R8o	1	1	0	0
R1x	3	2	2	0
R2x	3	1	0	0
R3x	3	1	1	0
R4x	4	2	0	0
R5x	2	0	0	0
R6x	4	0	1	0
R7x	5	2	0	0
R8x	1	1	0	0

Decisions Made through C4.5 Tree

Similar to the experiment for ID3 tree, the same 958 games were used to build C4.5 classifier. Using the same training set as test set for evaluation, the C4.5 decision tree reached 93.74% accuracy rate with 95 decisions potentially made. Following similar cleansing procedure, Figures 6-8 show the three branches of the C4.5 tree. Finally, 23 rule-related decisions were kept. The C4.5 decision tree generated 24.21% useable decisions. Some same rules were found in the branches and the frequencies were reported in Table 2. Rule R5x and R6x were not covered in the generated decision tree. The coverage of valid Tic-Tac-Toe rules was 14/16=87.5%.

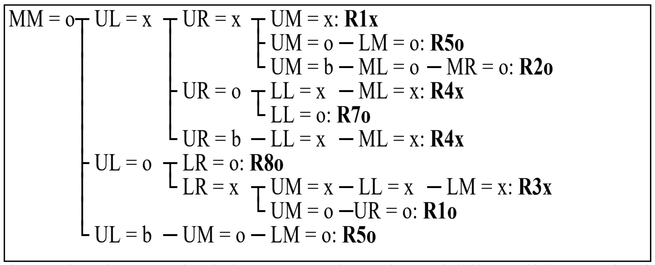


Figure 6. A branch of C4.5 decision tree when the middle cell is marked by player O

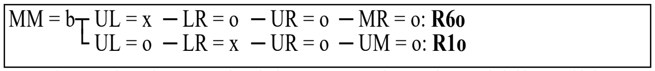

MM = b
UL = x – LR = o – UR = o – MR = o: **R6o**
UL = o – LR = x – UR = o – UM = o: **R1o**

Figure 7. A branch of C4.5 decision tree when the middle cell is not marked

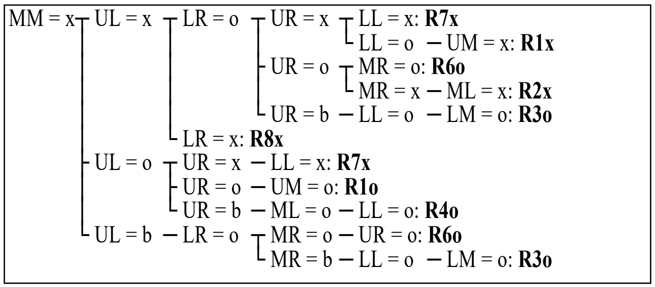


Figure 8. A branch of C4.5 decision tree when the middle cell is marked by player X

Decisions Made through Best First Tree

Following the same approach building classifiers as the ID3 and C4.5 to determine the winner of Tic-Tac-Toe games, Best First Tree is used to build a classifier for comparison. Using the whole 958 games for building the classifier and use the same set of observations for evaluation, the classifier can correctly classify 91.91% of games. Also, the Best First decision tree generated 47 decisions. After cleansing, 12 rule-based decision were left. Figures 9-10 show two branches of decisions established by the algorithm. However, only nine unique rules were created. It covered 9/16=56.25% of Tic-Tac-Toe rules.

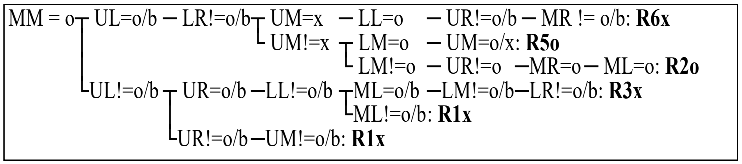


Figure 9. Best First decision tree when the middle cell is marked by player O

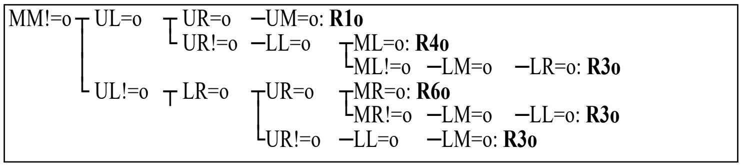

MM!=o
UL=o
UR=o –UM=o: **R1o**
UR!=o –LL=o
ML=o: **R4o**
ML!=o –LM=o –LR=o: **R3o**
UL!=o
LR=o
UR=o
MR=o: **R6o**
MR!=o –LM=o –LL=o: **R3o**
UR!=o –LL=o –LM=o: **R3o**

Figure 10. Best First decision tree when the middle cell is not marked by player O

Decisions Made through Functional Tree

Different from other algorithms branching the inner nodes, the functional tree algorithm applies logistic regression functions at the inner nodes. This classifier reaches 98.21% accuracy rate judging the winners from the 958 games. Only five decisions were made from the tree. Each node uses a function based on the value of nine grid position to determine the path. No easily readable rule is generated from this tree. As a result, it has 0% coverage of Tic-Tac-Toe rules.

Decisions Made through Apriori Association Rule

The results of experiment 2 using Apriori and Predictive Apriori association rule miners were reported in section 4.4 and 4.5, respectively. In a Tic-Tac-Toe game, the item set *I* = {UL, UM, UR, ML, MM, MR, LL, LM, LR, Winner}. Using the 958 terminal games served as transactions, the Apriori algorithm is to find the frequent item sets and generate rules based on given thresholds. In this experiment, the minimum support was set to 0.05 and confidence was set to 0.7, suggested by Na and Sohn (2011) using association rule to forecast changes in Stock Price Index. Table 3 shows the valid rules generated. First column indicated the rank of rules. Second column shows the association rule. For example, UL=x MM=x LR=x → Winner=x means UL=x, MM=x, LR=x, and Winner=x are four frequent items appearing in games together. The confidence is computed as follows.

$$Conf(UL = x, MM = x, LR = x \to Winner = x) = \frac{Sup(UL = x, MM = x, LR = x \cup Winner = x, D)}{Sup(UL = x, MM = x, LR = x, D)}$$

The result was listed in the third column and the confidence of UL=x MM=x LR=x → Winner=x was 1. The last column indicates the corresponding valid Tic-Tac-Toe game rule.

Table 3. Game rules generated by Apriori algorithm

Association Rule Rank	Association Rule	Confidence	Game Rule
1	UL=x MM=x LR=x → Winner=x	1	R8x
2	UR=x MM=x LL=x → Winner=x	1	R7x
3	UL=x UM=x UR=x → Winner=x	1	R1x
4	UL=x ML=x LL=x → Winner=x	1	R4x
5	UM=x MM=x LM=x → Winner=x	1	R5x
6	UR=x MR=x LR=x → Winner=x	1	R6x
7	ML=x MM=x MR=x → Winner=x	1	R2x
8	LL=x LM=x LR=x → Winner=x	1	R3x
9	UL=o MM=o LR=o → Winner=o	1	R8o
11	UR=o MM=o LL=o → Winner=o	1	R7o
95	UL=o LR=o Winner=o → MM=o	0.81	R8o
96	UR=o LL=o Winner=o → MM=o	0.81	R7o
252	MM=o LR=o Winner=o → UL=o	0.71	R8o
253	UL=o MM=o Winner=o → LR=o	0.71	R8o
254	MM=o LL=o Winner=o → UR=o	0.71	R7o
255	UR=o MM=o Winner=o → LL=o	0.71	R7o

All eight rules related to a "X" winner were identified from the top eight rules. However, the diagonal rules associated with an "O" winner were found from the 9th and 11th association rules. The equivalent two rules were found again at the 95th, 96th, 252nd, 253rd, 254th, 255th rules. The rules $X \to Y$ were considered the same when the $X \cup Y$ formed the same item set. As a result, three horizontal rules (R1o, R2o, R3o) and three vertical rules (R4o, R5o, R6o) associated with a winner "O" were missing. The coverage of the valid Tic-Tac-Toe game rules generated from the Apriori algorithm was 10/16=62.5%.

Decisions Made through Predictive Apriori Association Rule

The Predictive Apriori algorithm is to combine the support and confidence measures into a predictive accuracy measure. Similar to the Apriori algorithm, 958 games were used to mine association rules. Table 4 shows the rules found by the algorithm. First column lists the rank of the generated association rule (second column). Predictive accuracy is reported in the third column. The related game rule is listed in the last column. All 16 Tic-Tac-Toe game rules were found in the top 24 association rules, identified by the Predictive Apriori algorithm. Especially, all eight rules associated with the winner "X" were listed as the top eight association rules. The coverage of the valid Tic-Tac-Toe game rules generated from the Predictive Apriori algorithm was 100%.

Table 4. Game rules generated by Predictive Apriori algorithm

Association Rule Rank	Association Rule	Accuracy	Game Rule
1	UL=x MM=x LR=x → Winner=x	0.99499	R8x
2	UR=x MM=x LL=x → Winner=x	0.99499	R7x
3	UL=x UM=x UR=x → Winner=x	0.99498	R1x
4	UL=x ML=x LL=x → Winner=x	0.99498	R4x
5	UM=x MM=x LM=x → Winner=x	0.99498	R5x
6	UR=x MR=x LR=x → Winner=x	0.99498	R6x
7	ML=x MM=x MR=x → Winner=x	0.99498	R2x
8	LL=x LM=x LR=x → Winner=x	0.99498	R3x
9	UL=o MM=o LR=o → Winner=o	0.99495	R8o
11	UR=o MM=o LL=o → Winner=o	0.99495	R7o
19	UL=o UM=o UR=o → Winner=o	0.99488	R7o
20	UL=o ML=o LL=o → Winner=o	0.99488	R4o
21	UM=o MM=o LM=o → Winner=o	0.99488	R5o
22	UR=o MR=o LR=o → Winner=o	0.99488	R6o
23	ML=o MM=o MR=o → Winner=o	0.99488	R2o
24	LL=o LM=o LR=o → Winner=o	0.99488	R3o

CONCLUSIONS AND FUTURE DIRECTIONS

Two experiments were conducted in order to compare the Tic-Tac-Toe rules generated from decision tree classifiers and association rule miners. Although some redundant decisions related to the same valid rule were identified, the coverage of valid game rules generated by ID3 decision tree was 100%. In addition, using the training set for evaluation, the ID3 algorithm reached 100% accuracy rate in classifying Tic-Tac-Toe games. Using the same evaluation method, C4.5 decision tree reached 93.74% accuracy rate and generated a much smaller tree. The tree had 87.5% coverage of valid game rules. The worst decision tree result was produced by Best First tree. 91.91% of learned games can be correctly classified. This decision tree is the smallest among the three. However, only 56.25% of game rules can be extracted from the tree. While functional tree reaches 98.12% accuracy rate, it produced no easily readable rules due to the fact that logistic regression was used for nodes. It ended up with 0% coverage of the rules.

When using the same 958 terminal games for association miners, Apriori algorithm identified association rules which covered all game rules associated with a "X" winner. However, it only generated two valid game rules associated with an "O" winner. As a result, it had only 62.5% coverage of valid game rules. Another association rule algorithm Predictive Apriori identified all valid game rules from its top 24 outcomes.

In comparing the performance of rule generation suggested by decision tree and association rule algorithms, the best algorithm from both types had the same perfect coverage. However, association rule Predictive Apriori generated a cleaner rule set which is easier to comprehend. Especially, all eight rules associated with the winner "X" were listed as the top eight association rules.

The study contributes to both research and practice. It contributes to research by comparing the effectiveness of machine classifiers and association rule miners. The applicability of the two different rule generation techniques would suggest their pros and cons in implementing a decision support system which generates human-interpretable rules. Also, it may empirically benefit many different business applications, where managers need to interpret the result suggested and thus the found rules are needed to be coded into business processes. For example, loan managers will have the ability to tell rejected applicants what went wrong. The loan application program can also incorporate the found rules for future application processing and predictions.

Future studies can apply all 255,168 possible games for further comparison between decision tree classifiers and association rule miners. The outcome of these games may lead to a "X" winner, an "O" winner, and a draw game. The complexity of outcome and game size may further influence the performance of machine learners. It would be worth to examine other decision tree and association rule algorithms not covered in this study. Other games such as Connect Four (four in a row game) and Gomoku (five in a row game) are future candidates to compare the rules identified by both types of algorithms.

REFERENCES

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Acm sigmod record,* 22(2), 207-216, ACM.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37-66.

Anandarajan, M. (2002). Profiling Web usage in the workplace: A behavior-based artificial intelligence approach. *Journal of Management Information Systems*, *19*(1), 243-266.

Beitzel, S. M., Jensen, E. C., Lewis, D. D., Chowdhury, A., & Frieder, O. (2007). Automatic classification of web queries using very large unlabeled query logs. *ACM Transactions on Information Systems (TOIS)*, *25*(2), 9.

Churilov, L., Bagirov, A., Schwartz, D., Smith, K., & Dally, M. (2005). Data mining with combined use of optimization techniques and self-organizing maps for improving risk grouping rules: application to prostate cancer patients. *Journal of Management Information Systems*, *21*(4), 85-100.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337-407.

Gama, J. (2004). Functional trees. *Machine Learning*, *55*(3), 219-250.

Kang, M., Ahn, J., & Lee, K. (2018). Opinion mining using ensemble text hidden Markov models for text classification. *Expert Systems with Applications*, 94, 218-227.

Karabatak, M., & Ince, M. C. (2009). An expert system for detection of breast cancer based on association rules and neural network. *Expert systems with Applications*, *36*(2), 3465-3469.

Lee, C., Song, B., & Park, Y. (2012). Design of convergent product concepts based on functionality: An association rule mining and decision tree approach. *Expert Systems with Applications*, *39*(10), 9534-9542.

Liao, S. H., & Chen, Y. J. (2004). Mining customer knowledge for electronic catalog marketing. *Expert Systems with Applications*, *27*(4), 521-532.

Na, S. H., & Sohn, S. Y. (2011). Forecasting changes in Korea composite stock price index (KOSPI) using association rules. *Expert Systems with Applications*, *38*(7), 9046-9049.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.

Quinlan, J. R. (1993). Program for machine learning. C4.5., Morgan Kaufmann Publishers, San Mateo, CA.

S. Schaeffer (2002) Tic-Tac-Toe (Naughts and Crosses, Cheese and Crackers, etc.), Mathematical Recreations. Available: http://www.mathrec.org/old/2002jan/solutions.html

Scheffer, T. (2001, September). Finding association rules that trade support optimally against confidence. In *European conference on principles of data mining and knowledge discovery* (pp. 424-435). Springer, Berlin, Heidelberg.

Shi, H. (2007). Best-first decision tree learning. Hamilton, NZ.

Sinha, A. P., & May, J. H. (2004). Evaluating and tuning predictive data mining models using receiver operating characteristic curves. *Journal of Management Information Systems*, *21*(3), 249-280.

Spangler, S., Kreulen, J. T., & Lessler, J. (2003). Generating and browsing multiple taxonomies over a document collection. *Journal of Management Information Systems*, *19*(4), 191-212.

Tajbakhsh, A., Rahmati, M., & Mirzaei, A. (2009). Intrusion detection using fuzzy association rules. *Applied Soft Computing*, *9*(2), 462-469.

Vlas, R. E., & Robinson, W. N. (2012). Two rule-based natural language strategies for requirements discovery and classification in open source software development projects. *Journal of Management Information Systems*, *28*(4), 11-38.

Walczak, S. (2001). An empirical analysis of data requirements for financial forecasting with neural networks. *Journal of management information systems*, *17*(4), 203-222.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Zaslavsky, C. (1982). *Tic Tac Toe: And Other Three-In-A Row Games from Ancient Egypt to the Modern Computer*. Crowell.