Extracting Rules for Black Jack Using Machine Learning and Fuzzy Systems

Karla R. Cardoso
Fed. Territorial Univ. of the Semi-Arid
Natural and Exact Sciences Institute
Email: raphaela.krc@gmail.com

Marcos E. Cintra, Márcio Basgalupp
Science and Technology Institute
Federal University of São Paulo
Email: mecintra@gmail.com, basgalupp@gmail.com

Abstract—**Black Jack is a card game played by two or more players who aim at reaching 21 points. It is one of the most played card games as it presents an advantage for the player against the Casino: the player decides what to do before the dealer and he/she may stand (stop getting cards) when reaching a sum close to and smaller than 21. In the literature, it is possible to find some techniques proposed to support the prediction of the decision to be taken in accordance to the player's hand. Such techniques focus on finding a way to minimize the casino's advantage and turn the odds in favour of the player. Such strategies currently used were defined in the 1960's by mathematicians based on probability with hundreds of hands. However, these strategies are complex, making it difficult to memorize all possible actions to be taken. In this sense, this study aims to analyse data sets containing information on Black Jack hands in order to obtain rules to favour the player. This is an initial work, as there is a lack of proposals in the literature, based on the hypothesis that we could obtain sets of rules that could be used to support the player's decisions using machine learning algorithms. We used two fuzzy rule-based algorithms, namely FuzzyDT and FuzzyFCA, as well as the classic C4.5, PART, and Ripper algorithms to extract rules. The rules obtained are described and the results discussed.**

I. Introduction

Black Jack (BJ), also known as *twenty-one*, is one of the most played casino game in the world [16]. It is played by one or more players and a dealer. The aim of the game is to beat the dealer by getting 21 points on the player's first two cards (called a BJ), without a dealer's BJ, reaching a final score higher than the dealer's without exceeding 21 or by letting the dealer draw cards until exceeding 21.

The existing strategies for BJ basically consist on a specific system for mentally counting cards in order to decide the next decision. Such strategies were defined in the 1960's by mathematicians using probability. In this work, we aimed at finding new strategies using machine learning techniques applied to large sets containing information on possible hands of BJ. The hypothesis of this work is that it is possible to find new BJ strategies in the form of rule sets that offer advantage to the player over the dealer using machine learning algorithms for classification, association rule, and fuzzy systems.

Classification is a relevant machine learning task, widely used for pattern recognition, data mining, and decision making, among others [7]. Decision trees (DT) are widely used for classification [15]. DT algorithms usually produce simple models with competitive classification rates which can be graphically represented. DTs are highly interpretable and can handle continuous and discrete attributes. C4.5 [15] is a popular DT based on entropy and information gain.

The fuzzy logic [22], [12] allows the natural treatment of uncertainty and imprecision present in data. Fuzzy systems use a reasoning mechanism which is able to represent the subjectivity of the human thought. Any system that includes at least one variable defined by linguistic terms which, in turn, are represented by fuzzy sets, according to the fuzzy set and fuzzy logic theories, is considered a fuzzy system. Rule-based fuzzy systems are formed by a knowledge base, which contains a fuzzy data base, a fuzzy rule base, and an inference mechanism. The fuzzy data base contains the definitions of the variables in terms of fuzzy sets. The fuzzy rule base contains a set of fuzzy rules representing the knowledge of the domain. The inference mechanism uses both, the fuzzy rule base and the fuzzy data base to classify an input example.

FuzzyFCA [8], for instance, is a fuzzy genetic system which forms a fuzzy rule base by selecting rules from a set of candidate rules. FuzzyFCA uses the Formal Concept Analysis to extract the rules that form the set of candidate rules. Fuzzy decision trees have also been proposed in the literature [18], [5], [14]. Fuzzy DTs combine the high interpretability of DTs with the capability of dealing with imprecision and uncertainty in data. FuzzyDT [5] is based on the classic C4.5 algorithm.

In order to extract rule sets to support players of BJ, we initially used the following algorithms: C4.5 [15], FuzzyFCA [8], FuzzyDT [5], PART [1], and Ripper [9]. These algorithms were empirically selected based on their characteristics of computational cost and ability to deal with large datasets, as well as for the fact that they produce interpretable models (rule sets). As these are initial experiments, not all obtained rule sets present the required characteristics to make them useful for real use. Nevertheless, due to the lack of proposals on the use of Machine Learning with BJ, our results are presented and discussed, as they can be the base for further studies.

The remaining of this paper is organized as follows: Section II presents the BJ game; Section II-C presents a bibliographical review of related proposals and BJ strategies. Section III describes our experiments, followed by the final considerations and future work in Section IV.

978-1-5090-6020-7/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: Universidade Federal de Vicosa. Downloaded on November 18,2022 at 13:44:03 UTC from IEEE Xplore. Restrictions apply.

II. The Black Jack Game

BJ [17], [4], also known as 21, is one of the most played games in American casinos [10]. In fact, BJ competes with Poker, Roulette, and Craps, which are considered standard gambling games. Among such games, BJ is the only one that presents an advantage to the player in relation to the dealer (casino) as it allows the player to count the cards during a hand, making it possible for the player to make plausible predictions on future deals. This way, the players can choose the most appropriate move according to the cards on the table. However, BJ is still quite neglected in the scientific game literature and offers a relatively unexplored area for mathematical and statistical analysis [2].

A. The rules of the game

A hand starts when bets are placed. Next, the dealer distributes the cards. Each player receives two cards facing down, while the dealer shows his/her first card and covers the second one. After card distribution, each player can **Hit** (get a new card), ***Stand*** (get no more cards), ***Double down*** (double the bet), or ***Split*** the cards into two sets (only when the two cards have the same value). When the player splits, he/she can ask for as many cards as he/she wants, as long as it does not exceed 21 points. The player usually hits until reaching a relatively good hand (close to 21 points) and then Stands.

Once all players have played, from left to right of the dealer, it is the dealer's turn. The most frequently used strategy on casinos proposes that the dealer should hit until reaching a sum of 17 to 21 points. If the dealer exceeds 21, all the players which had a sum of less than 21 points win.

Each card has a specific value, although when summing the cards, the numeric value or figure (Jack, Queen, King, Ace), does not consider the suit of the cards. Queens, Kings and Jacks value 10. The Ace's value can be defined as 1 or 10. This way, if the first two cards are Aces and the third card values 10, the sum is considered 21 (a BJ). However, if the sum of the cards is less than 21 and the next card is an ace, the following conditions are evaluated: if the sum of the cards, considering the Ace as 11, is less than 21, the Ace's value remains 11; if the sum of the cards, considering the Ace as 11, is higher than 21, the Ace's value is considered 1.

B. Strategies for counting the cards

Different counting strategies have been proposed. Next, we present some of the most well-known card counting strategies.

1) The Thorp strategy

The mathematician Edward Thorp [19] developed this strategy which consists in analysing the cards on the table, summing the cards of the players and the dealer, and proposing an action to each configuration. The Thorp strategy had much repercussion and several casinos forbid its use. Since then, the player has to memorize and perfect this quite complicated strategy, while disguising the fact that he/she is counting the cards. Table I presents the Thorp strategy.

TABLE I
THORP STRATEGY TABLE OF ACTIONS.

	Dealer									
Player	2	3	4	5	6	7	8	9	10	A
5..7	H	H	H	H	H	H	H	H	H	H
8	H	H	H	D	D	H	H	H	H	H
9	D	D	D	D	D	H	H	H	H	H
10	D	D	D	D	D	D	D	D	H	H
11	D	D	D	D	D	D	D	D	D	D
12	H	H	S	S	S	H	H	H	H	H
13	S	S	S	S	S	H	H	H	H	H
14	S	S	S	S	S	H	H	H	H	H
15	S	S	S	S	S	H	H	H	H	H
16	S	S	S	S	S	H	H	H	H	H
17+	S	S	S	S	S	S	S	S	S	S
A 2	H	H	D	D	D	H	H	H	H	H
A 3	H	H	D	D	D	H	H	H	H	H
A 4	H	H	D	D	D	H	H	H	H	H
A 5	H	H	D	D	D	H	H	H	H	H
A 6	D	D	D	D	D	H	H	H	H	H
A 7	S	D	D	D	D	S	S	H	H	S
A 8	S	S	S	S	S	S	S	S	S	S
A 9	S	S	S	S	S	S	S	S	S	S
2 2	P	P	P	P	P	P	H	H	H	H
3 3	P	P	P	P	P	P	H	H	H	H
4 4	H	H	H	P	D	H	H	H	H	H
5 5	D	D	D	D	D	D	D	D	H	H
6 6	P	P	P	P	P	P	H	H	H	H
7 7	P	P	P	P	P	P	P	H	S	H
8 8	P	P	P	P	P	P	P	P	P	P
9 9	P	P	P	P	P	P	P	P	S	S
10 10	S	S	S	S	S	S	S	S	S	S
A A	P	P	P	P	P	P	P	P	P	P

Table I presents, in each column, the value of the dealer's card, and, in each row, the sum of the players cards. Notice that "H" means the player should Hit (ask for a new card), "S" means Stand (no more cards), "D"means Double the bet, and "P" means the player should Split the cards. Table I first considers the sum of the player's card, and then, the union of an Ace with another card, and, finally, doubling the cards. Consider the following example: if the sum of the player's card is 16 and the card on the table is 7, then the player should Hit (ask for another card). However, if the card on the table is a 6, then the player should Stand.

2) The High-Low strategy

One of the most popular and easy strategies to count cards for BJ is the High-Low, or just Hi-Lo [21]. The reason for its popularity is that it is easy to learn and to implement. Each card of the deck receives a value: -1, +1, or 0 [23]. The counting strategy of Hi-Lo is an easy way to follow the high cards in the deck. This strategy is called an "equilibrated system". If all cards of the deck are counted, the sum is zero. The values of each card are presented next.

- Cards from 2 to 6 = +1
- Cards from 7 to 9 = 0
- Cards from 10 to Ace = -1

The idea behind Hi-Lo is that the bigger the sum of all cards, the bigger the odds of big cards in the deck ("small" cards were drawn). Notice that it is important to know if the deck has many big cards as big cards favour the players and the dealer equally. However, if it is known that there is a higher chance of a big card, it makes deciding on a difficult move easier: the smaller the sum, the higher the probability of bigger cards in the deck, thus, the player should be cautious. On the other hand, if the sum is negative and the player has to take a decision, the player can Hit, knowing that small cards are more probable. The more cards are distributed and counted, the more precise the Hi-Lo method is.

Authorized licensed use limited to: Universidade Federal de Vicosa. Downloaded on November 18,2022 at 13:44:03 UTC from IEEE Xplore. Restrictions apply.

3) The Uston SS strategy

The Uston SS counting was defined from the need of Ken Uston, one of the most well-known experts in BJ [20]. The Uston SS technique is more advanced than Hi-Lo and offers more chances of winning, but it is harder to learn. It works the same way as Hi-Lo, but it has 6 categories, instead of just 3. The values of the cards are:

- Cards 2, 3, 4, 6 = $+2$
- Cards 5 =+3
- Cards $7 = +1$
- Cards $8 = 0$
- Cards $9 = -1$
- Cards 10 to A $= -2$

The same rules of Hi-Lo apply to Uston SS. If during the game the value of the cards on the table is positive, there will be more chances of big cards and the player should risk Hitting. However, if the value is negative, the player should act with caution based on his/her bets.

C. BJ strategies in the scientific literature

There are many books describing the counting strategies previously presented in this paper. However, there are few proposals in the scientific literature for BJ. The reasons for that may include the fact that the computational representation of the game can be challenging. Also, for a strategy to be feasible and usable in practice, it must be presented in an interpretable way and include few rules. In fact, as the game is ruled by the probability distribution of the cards, such condensed and highly interpretable strategies are difficult to be proposed using the existing algorithms. Next, we present two proposals found in the literature.

In [11], the authors use three neural networks (one for Splitting, one for Doubling, and a last one to Hit) to evolve strategies. Initially, a set of 60 players were randomly generated across 10 tables, and 1,000 hands were played. In the sequel, modifications are carried out in the networks in order to improve the strategies. Finally, the best strategies are compared and it is pointed out that they evolved to other known strategies, which shows that the player can win a casino hand at an intermediary level.

In [10], the author presents the use of evolutionary algorithms based on the Thorp strategy [19]. Moreover, the Hi-Lo counting method is also considered. The strategy proposed based on the experiments carried out by the author shows an advantage of the players over the dealer.

III. Experiments

Initially, two datasets containing 10,000 and 1,000,000 BJ hands were randomly created. The datasets were defined in the ARFF format, in order to be used with WEKA's[13] algorithms and are available at UCI Machine Learning Repository. In order to define the entries of the datasets, we considered the following restrictions:

- The use of a single deck of cards;
- The entries represent hands between a single player and the dealer;
- It is a fact that the maximum number of cards a player or dealer can get not exceeding 21 is 11 cards, considering the drawn cards are all four ace cards, four 2 cards, and three 3 cards, totaling 21 points. Due to the low probability of a player getting the 11 smaller cards, and due to processing and memory restrictions, we empirically defined 8 as the maximum number of cards the player or dealer can have in a single hand for the 10,000 hand dataset. The larger dataset considers up to 10 cards for the player and for the dealer.

Once these restrictions were established, the entries were defined with the following format:

- CoP1, CoP2, CoP3, CoP4, ct5, CoP6, CoP7, CoP8, CoP9, CoP10, SoP, TCoP, CoD1, CoD2, CoD3, CoD4, CoD5, CoD6, CoD7, CoD8, CoD9, CoD10, SoD, TCoD, W;

Where:

- CoP1 to CoP10: value of the 1st to 10th player's cards;
- SoP: sum of all player's cards when he/she stands;
- TCoP: total number of cards of the player in the hand;
- CoD1 to CoD10: value of the 1st to 10th dealer's cards;
- SoD: sum of all dealer's cards at the end of the hand;
- TCoD: total number of cards of the dealer in the hand;
- W: winner. W is set to 0 (zero) if the dealer wins, 1 (one) if the player wins, and 2 (two), if there is a tie.

Consider the following example of the dataset with 10,000 entries (up to 8 cards for the player and dealer):

- 4, 6, 7, 3, 0, 0, 0, 0, 20, 8, 8, 2, 0, 0, 0, 0, 0, 18, 3, 1;

The values of the entry correspond to:

- 4, 6, 7, 3, 0, 0, 0, and 0 are the values of the first to the eighth card of the player;
- 20: sum of the values of all the cards of the player
- 4: number of cards the player received;
- 8, 8, 2, 0, 0, 0, 0, and 0 are the values of the first to the eighth card of the dealer;
- 18: sum of the values of all the cards of the dealer;
- 3: number of cards the dealer received;
- 1: the player won the hand.

As previoulsy stated, we extracted rule sets using the following algorithms: C4.5 [15], FuzzyFCA [6], FuzzyDT [5], PART [1], and Ripper [9]. All algorithms were used with default parameters. As the number of examples in a dataset directly influences the choice of the algorithms that can process them, the smaller dataset (with 10,000 hands) was used with C4.5, FuzzyDT, PART, and Ripper. The dataset with 1,000,000 hands was used with the C4.5 and FuzzyFCA. The experiments are described next.

A. Experiments with the dataset containing $10,000$ hands

Next, we present the experiments with C4.5, FuzzyDT, PART, and Ripper algorithms and the extracted rule sets.

1) Experiments using C4.5

For the C4.5 algorithm, three models were generated:

1) Using all attributes (21 attributes);
2) Using only 5 attributes: the sum of the cards of the player and dealer (SoP and SoD), the number of cards

Authorized licensed use limited to: Universidade Federal de Vicosa. Downloaded on November 18,2022 at 13:44:03 UTC from IEEE Xplore. Restrictions apply.

used by the player and the dealer ($TCoP$ and $TCoD$), and the winner;

3) Using all attributes, except the SoP and SoD.

In spite of the difference in the number of attributes, both datasets (containing 10,000 and 1,000,000 randomly generated BJ hands) produced the same model for the first two previously described setups. The model is shown in Figure 1.

[scale=0.48]arvoreTotal2.jpg

Fig. 1. Decision tree generated by C4.5 using information on 10,000 hands.

When considering only the branches that has the Player as the winner, the decision tree in Figure 1 can be seen as a set of 10 rules. Due to space limits, we show only the first three rules found in the DT. Notice that the number in brackets next to the winner is the total number of hands from the dataset that match that branch.

1) *If* SoP > 21 *then* Winner = Dealer
2) *If* SoP ≤ 21 *and* SoD > 21 *then* Winner = Player
3) *If* SoP ≤ 18 *and* $18 <$ SoD ≤ 21 *then* Winner = Player

Notice that these rules were simplified from the actual rules of the induced DT. The simplification makes it easier for the memorization of such rules and their real use. For instance, the blue coloured branches can be simplified into a single rule, in which the winner is the dealer. However, the induced tree presents obvious rules and only use the sum of the cards of the player and the sum of the cards of the dealer as attributes. Such rules are not relevant for players. In an attempt to extract more relevant rules, we induced the third DT, removing these two attributes (SoP and SoD). However, due to its size (809 branches/rules), the generated rule set is impossible to be memorized and used.

2) Experiments using PART

PART produced 25 rules. Due to space restrictions, we also show only the three first ones. Notice that the number in brackets at the end of the rule is the number of hands from the dataset that match that rule.

1) *If* SoD > 21 *then* Winner = Player (2019.0)
2) *If* SoP > 19 *and* SoD $<= 19$ *then* Winner = Player (1254.0)
3) *If* SoD $<= 16$ *and* SoP > 16 *then* Winner = Player (559.0)

The rules extracted by PART, as with C4.5, only include the sum of the cards of the player and of the dealer (SoP and SoD) as attributes. Again, in an attempt to extract relevant rules, we removed the SoP and SoD attributes and obtained a large set of 398 rules, thus, too large for use in practice. However, the rule set has two important characteristics: i) when compared to the rules obtained with C4.5, the rules extracted by PART are already simplified, presenting only one condition for each attribute, and ii) the rule set, as a whole, points to the rule, taken as common knowledge, that when reaching 17 points, the player should Stand.

3) Experiments using Ripper

Ripper extracted a set of 36 rules. Next, due to space restrictions, we show and analyze three of these rules.

1) *If* CoP1 $>= 10$ *and* CoP2 $>= 10$ *and* CoP3 $<= 0$ *and* CoD1 $>= 10$ *and* CoD2 $>= 10$ *and* CoD3 $<= 0$ *then* Winner is Tie
2) *If* CoP3 $<= 0$ *and* CoD2 $>= 8$ *and* CoD3 $>= 1$ *and* CoD4 $>= 6$ *then* Winner is Player
3) *If* CoP3 $<= 0$ *and* CoD1 $>= 8$ *and* CoD2 $>= 4$ *and* CoD3 $>= 8$ *then* Winner is Player

The first rule states that if the player gets two 10 cards or two aces, he should Stand. The same condition is described for the Dealer, resulting in a tie. The second rule describes a situation in which the player Stands with only two cards (card 3 = 0), while the dealer gets four cards. Unfortunately, the value of the first card of the Dealer is not presented. The player wins, probably because the Dealer got more than 21 points. The last rule is similar to the second one, but in this case the cards of the Player are known: he/she Stands with two cards, which can be an ace and a 10 or two aces (a BJ), as the total points of the Dealer is 20. Although the number of rules is not as large as the ones produced by C4.5 and PART, the number of conditions in the rules can be considered a problem for their practical use and each rule must be analysed by an expert in order to became really useful (or not). Also, another shortcoming with the obtained rules is the fact that not all rules present a condition for the first two cards of the player and the first card of the dealer, which are the basic configuration of a BJ hand.

4) Experiments using FuzzyDT

For the FuzzyDT and FuzzyFCA algorithms, the attributes were defined using 3, 5, and 7 triangular sets evenly distributed in the domains. Figure 2 and Figure 3 show the definition of the linguistic variable representing the sum of the cards of the player/dealer, and of the linguistic variable representing the value of each single card. The definitions of the linguistic variables were done empirically.

[width=.40]conjuntoSoma.pdf

Fig. 2. Definition of the sum of the cards in terms of fuzzy sets.

[width=.40]conjuntoCarta.pdf

Fig. 3. Definition of the value of the cards in terms of fuzzy sets.

The total number of rules obtained by FuzzyDT using 3 fuzzy sets was 319, for 5 fuzzy sets, 517 rules, and for 7 fuzzy sets, 451 rules. Due to the large number of rules in each rule set, only 3 rules generated by FuzzyDT using 3 fuzzy sets (low, medium, high) are shown next.

- *If* CoP1 = low *and* CoP2 = low *and* SoP = low *and* CoD1 = low *and* CoD2 = medium *and* CoD3 = low *and* SoD = low *and* TCoD = low *then* Winner = Player
- *If* CoP1 = low *and* CoP2 = low *and* SoP = low *and* CoD1 = low *and* CoD2 high CoD3 = low *and* SoD = low *and* TCoD = low *and then* Winner = Player
- *If* CoP1 = medium *and* CoP2 = medium *and* SoP = low *and* CoD1 = low *and* CoD3 = low *and* SoD = low *and* TCoD = low *then* Winner = Player

Notice that the rules generated by FuzzyDT use the values of the cards of the player and the values of the cards of the dealer, as well as the sum of the cards. However, the number of rules and the number of conditions in each rule pose a problem for their use as part of a BJ strategy.

B. Experiments with the dataset containing $1,000,000$ hands

For the larger dataset, the C4.5 and FuzzyFCA methods were used, due to their characteristics of being able to deal with large datasets. We also used the FuzzyDT and Apriori

Authorized licensed use limited to: Universidade Federal de Vicosa. Downloaded on November 18,2022 at 13:44:03 UTC from IEEE Xplore. Restrictions apply.

algorithms, but did not obtain good results. Thus, these experiments are not described. The results obtained with C4.5 and FuzzyFCA are shown next.

1) Experiments using C4.5

For the larger dataset, as with the smaller one, C4.5 was used to generate three models:

1) Using all attributes (21 attributes);
2) Using only 5 attributes: the sum of the cards of the player and dealer (SoP and SoD), the number of cards used by the player and the dealer ($TCoP$ and $TCoD$), and the winner;
3) Using all attributes, except SoP and SoD.

The two first models are identical to the models obtained with the smaller dataset and present only the sum of the cards of the player, of the dealer (SoP and SoD) and the Winner attributes. The last model produced a set with many hundreds of rules.

2) Experiments using FuzzyFCA

The FuzzyFCA algorithm was executed using a 10-fold cross-validation strategy. Thus, FuzzyFCA generated 10 sets of rules containing an average of 320 rules each.

We searched in all 10 rule sets for identical rules. Next we present the set of 6 rules which consider the Player as winner and which are present in all 10 rule sets. We found 9 rules in each the Dealer wins in all 10 rule sets, and 4 rules with ties in all 10 rule sets. Due to space reasons, these rules are not presented.

1) *If* CoP1 = *low and* CoP2 = *medium and* CoP4 = *low and* CoP5 = *low and* SoP = *low and* CoD2 = *high and* CoD3 = *low and* TCoP = *high and* SoD = *medium then* Winner = Player
2) *If* CoP3 = *medium and* TCoP = *medium and* CoD2 = *medium and* CoD3 = *high then* Winner = Player
3) *If* CoP2 = *high and* CoP4 = *low and* CoP5 = *low and* TCoP = *medium and* SoP = *low and* CoD2 = *low and* CoD5 = *low and* TCoD = *high then* Winner = Player
4) *If* CoP3 = *high and* CoP5 = *low and* TCoP = *medium and* CoD2 = *medium and* CoD4 = *low and* CoD5 = *low and* TCoD = *high and* SoD = *low then* Winner = Player
5) *If* CoP3 = *low and* CoP5 = *low and* CoD1 = *high and* CoD2 = *medium and* CoD4 = *low then* Winner = Player
6) *If* CoP2 = *medium and* TCoP = *medium and* SoP = *medium and* CoD2 = *low then* Winner = Player

FuzzyFCA, as with the other algorithms but Ripper, produced large sets of rules, most of them containing many conditions. Thus, the rule sets produced by FuzzyFCA also cannot be used in practice for the game of BJ. Focusing on these 6 rules, we tried to modify them using the knowledge of a player (an expert). However, the linguistic values used in the rules were perceived by the player as more difficult to process than integer values. Also, as with Ripper, another shortcoming with the obtained rules is the fact that not all rules present a condition for the first two cards of the player and the first card of the dealer, which are the basic configuration of a BJ hand.

C. Considerations on the rule sets obtained

The total number of rules obtained by each algorithm are shown next:

- C4.5: 398 rules;
- FuzzyFCA: $\approx$ 320 rules;
- FuzzyDT: 319 rules;
- PART: 398 rules;
- Ripper: 39 rules.

Two algorithms produced reduced rule sets, C4.5 and Ripper. However, the rules produced by C4.5 do no present any relevant information for the game of BJ. Ripper, on the other hand, produced rules with many conditions, which makes it difficult to use them in practice. When removing the SoD and SoP attributes, C4.5 produced a large set of rules which cannot be used in practice. The remaining algorithms, FuzzyDT, FuzzyFCA, and PART also produced large rule sets.

D. Evaluation of the obtained models

In order to evaluate the obtained models, we played 50 hands using a deck with 52 cards. Initially, the player and dealer received two cards, and then, the models were used to decide their next moves. Due to space limitations, we do not present the cards of each hand. For FuzzyFCA, we only used the set of 19 rules that were present in all 10 rule sets. Also, when two rules were possible for FuzzyFCA, we used only the one presenting the linguistic value with bigger membership degree.

As previously stated, only FuzzyFCA produced rules that use the values of the cards in their conditions. However, these rules were not enough to contemplate all possible combinations in the hands, thus, a dice was used to decide what to do when no rule covered the setup of the hand.

Table II shows the results of the hands played.

TABLE II
PERCENTAGE OF WINS FOR 50 HANDS PLAYED USING EXTRACTED MODELS.

	Player	Dealer	Tie
C4.5	42	46	12
FuzzyFCA	71	21	8
FuzzyDT	40	58	2
PART	67	24	9
Ripper	40	46	14

The values in Table II represent the percentage of wins for the Player and for the Dealer, as well as the percentage of ties. FuzzyFCA and PART obtained better results for the player. All other algorithms provided better results for the Dealer.

IV. Final Considerations and Future Work

BJ is one of a few card games played in casinos that gives the players some advantage over the dealer. In order to support players of BJ, some strategies for counting cards have been proposed. Such strategies are based on the studies of mathematicians who analysed large sets of examples of the game and applied probability to devise counting strategies.

In this paper, we present the experiments carried out using machine learning algorithms and rule-based fuzzy systems to extract rules from two sets of 1,000,000 and 10,000 BJ hands. Each entry of the datasets represent a complete hand containing the information on the values of the cards of the player and the dealer, as well as the winner. The dataset with 1,000,000 hand information was made available at UCI as a

Authorized licensed use limited to: Universidade Federal de Vicosa. Downloaded on November 18,2022 at 13:44:03 UTC from IEEE Xplore. Restrictions apply.

contribution of this study. We applied the classic C4.5, PART, and Ripper algorithms to extract rules from both datasets. We also applied FuzzyDT and FuzzyFCA, two rule-base fuzzy systems to extract rules.

All 4 decision trees induced by C4.5 only used the attributes containing the sum of the cards of the player and dealer. In an attempt to obtain relevant rules, the C4.5 algorithm was executed removing the SoP and SoD attributes, generating a large set of rules. Large rule sets were also obtained by FuzzyDT, FuzzyFCA, and PART. Ripper produced a smaller rule set, but with many conditions in them. For FuzzyFCA, a small rule set was selected contemplating only the rules found in all 10 rule sets, as the experiments were performed using a 10-fold cross-validation strategy.

Our next step is to work on the condensation of the rule sets in order to define table representations of strategies for each rule set, similar to the table defined by Thorp (presented in Table I), and compare the obtained models with Thorp. We also intend to work on the definition of the linguistic variables used by FuzzyFCA and FuzzyDT with two fuzzy sets. We also intend to extract rules using other machine learning and fuzzy algorithms. Another important aspect to be addressed in future work is to run more hands with the extracted models. We also intend to adopt the datasets to be able to use more than one deck of cards and more than one player in the game at the same time.

ACKNOWLEDGEMENTS

The authors would like to thank CAPES and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grant 2016/02870-0, for funding this research.

REFERENCES

Authorized licensed use limited to: Universidade Federal de Vicosa. Downloaded on November 18,2022 at 13:44:03 UTC from IEEE Xplore. Restrictions apply.