What's the Next Move? Learning Player Strategies in Zoom Poker Games

Murillo G. Carneiro and Gabriel A. de Lisboa
Faculdade de Computação, Universidade Federal de Uberlândia
Uberlândia, MG, 38400-902
Email: {mgcarneiro, gabriel_alves}@ufu.br

***Abstract*—In this article, we address the problem of modeling the actions of a human player in order to learn his strategies from his past game logs in Zoom Texas Hold'em poker variant. Although Texas Hold'em is a very popular game, Zoom is yet a very recent format of game in which, instead of playing in a specific table against a specific set of opponents, a player is placed in a large pool of players in which their opponents change every hand. Pros and cons of Zoom include respectively bigger effective time playing (and possibly getting money) and scarcity of data to get reads from the opponents. To deal with this problem, our model consists of a simple and generic set of features designed to fulfill each one of four proposed categories (hand quality, position insights, aggressiveness and current situation) in order to be able to capture a wide range of player strategies in each stage of the game. As a consequence of our modeling, we generate five data sets which were further evaluated by machine learning techniques. The results show that much of the player strategies were effectively learned, especially by non-linear techniques. Moreover, our data sets are available online as a test-bed for machine learning research in poker games.**

I. Introduction

Poker is a very popular family of card games played around the world in which players bet (or bluff) that their hand[1] is better than their opponents hand. Although poker gained attention only recently in Computer Science, it has been studied for many decades in areas like Mathematics and Economics, where was used, for example, by von Neumann and John Nash in the earliest investigations in game theory. The game is considered an interesting test-bed for Artificial Intelligence research as it comprises a set of challenging characteristics that include multiple agents, incomplete information (partially observable), stochastic states and sequential actions [1], [2].

Among the hundreds of poker variants, the game of Texas Hold'em is the most played and one of the most strategically complex [3]. In the literature, it is also the most investigated, especially the two-player version (Heads-up) due to its smaller complexity in comparison to the multi-player ones (Multi-way). Many of Heads-up works present contributions in terms of poker player agents which are related to the development of algorithms able to play poker against both computers or human players; the concepts and methods behind such agents can be associated with knowledge-based systems [4], [5], evolutionary algorithms and neural networks [6], simulation-based techniques like Monte Carlo [7] and near-equilibrium algorithms [8], just to name a few. Most recent Heads-up works have developed agents able to beat even professional human players [9], [10]. Poker agents have also been developed for the Multi-way version mostly inspired in concepts and methods from Heads-up agents [11], [12], although their performances seem not even close to the latters. Multi-way works have also attempted to characterize players styles, for example, to explain successful play [13]. Another topic largely investigated in the literature is opponent modeling. It consists of establishing a probability distribution of the opponents hand as well as modeling their actions in specific situations by considering their past actions in the game [14]–[16].

In this paper, we extend the poker literature by investigating a recent format of online game named Zoom. To the best of our knowledge, this is the first investigation concerning Zoom poker. In the traditional format, a player usually does many games against a specific set of players in a given table, which makes quite possible to exploit reading about his opponents. By the contrary, Zoom format consists of a large pool of players (often between 500 and 2000) in which the opponents of a player change every hand. Zoom poker has attracted a lot of players especially because they can play hands faster than in the traditional format. For example, when players fold in Zoom games they are taken to another table to start immediately a new hand, i.e., they do not wait until an eventual showdown to play again. However, the scarcity (or lack) of opponents data in Zoom represents a very complex problem for most opponent modeling methods as they usually require some amount of information from the past games of the opponents in order to give some insight.

To be specific, we address here the problem of modeling the actions of a human player in order to learn his strategies from his past game logs in the Zoom Texas Hold'em poker variant. Such a problem has a variety of applications which includes empirical validation of theoretical concepts in poker, better understanding of the player actions and comparative analysis of players strategies, just to name a few. To handle the problem, we model a simple and generic set of features able to capture a wide range of players strategies at each stage of the game. The efficiency of such a model is further evaluated by supervised machine learning techniques.

The remainder of the paper is organized as follows. Section II presents a relevant background about poker. Section III describes in detail our model to learn player strategies,

[1] In poker, the term "hand" denotes both the player hole (private) cards or a game. The reader can understand the meaning by the context.

including data, problem formulation, feature designing and preprocessing steps. Section IV presents the experimental results obtained by applying machine learning techniques on the data sets generated. Finally, Section V concludes the paper.

II. Background

In the following we briefly describe relevant concepts about the poker variant considered in this article and hand assessment methods.

A. No Limit Texas Hold'em

The No Limit Texas Hold'em is the most popular poker variant. At the beginning of every hand, called the **pre-flop**, all players are dealt two hole cards (private or pocket cards). The dealer player is assigned and marked with a dealer button. Before any betting takes place, the two players to the left of dealer are forced to post the small blind and big blind bets in order to assure that every hand have money involved. The small blind is typically a half of the big blind (the minimum bet). Fig. 1 shows an illustration about the initial configuration of the table. Note that the dealer position rotates clockwise from hand to hand.

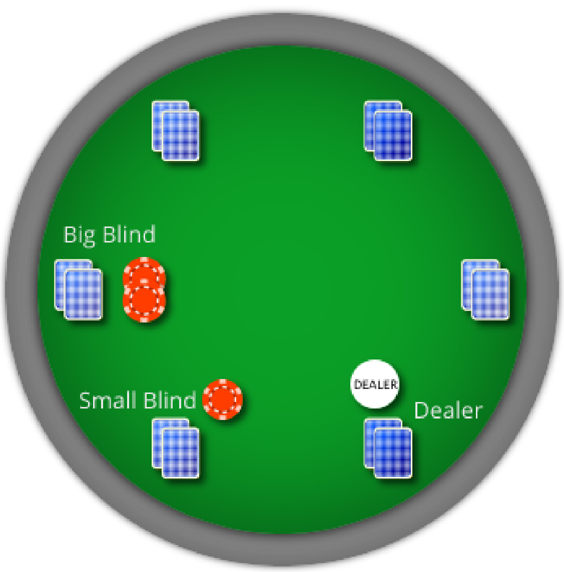


Fig. 1: The initial configuration of a hand in Texas Hold'em.

The first betting round occurs from the player to the left of the big blind. After that round, three community cards (shared or board cards) are dealt face up on the table, collectively called the **flop**, and the second betting round ensues. In the following, a fourth community card is dealt face up, called the **turn**, and the third round of betting occurs. On the **river**, a fifth community card is dealt face up and the final betting round occurs. Table I quickly describes each of the game stages. In each betting round, every player can choose one of the following actions:

- Fold: When the player gives up the hand and, consequently, every chip previously he puts into the pot.

TABLE I: Summary of the rounds in Texas Hold'em.

	1	Small blind and big blind
	2	Distribution of the hole cards
Pre-flop	3	First betting round
Post-flop	4	Flop → First three community cards
	5	Second betting round
	6	Turn → Fourth community card
	7	Third betting round
	8	River → Fifth three community card
	9	Fourth betting round
	10	Showdown → The remaining players show their hands

- Check: When the player decides not to bet, but keeps his cards and wait the next player action.
- Call: When the player matches the current highest bet in order to continue disputing the pot.
- Bet: When the player bets, although he could have checked.
- Raise: When the player bets higher than the current highest bet.

In No Limit Texas Hold'em there is no bet limit, therefore the value of the bet can go from the minimum bet up to the total amount of chips the player has. If the final betting round is completed with at least two players disputing the pot, then those players reveal their hole cards, called the **showdown**, and the player with the best hand wins the pot (it is divided equally if there is a tie).

A poker hand identifies the strength of a player in a game of poker. It denotes the best possible rank of all five card combinations including the player hole cards and the community cards. In such a combination it is possible to use both, one or none of the hole cards. Fig. 2 shows the possible hand ranks in the game (stronger ranks first).

B. Hand Assessment

Poker literature contains a set of tools often used to evaluate hands or game situations. Next we quickly introduce those that are employed in this paper.

The *Hand Strength* (HS) uses enumeration methods to estimate the probability of a player hand being the best at any time [14]. The *Hand Potential* uses enumeration methods to estimate both the positive and negative effects of the future community cards: the positive potential ($PPot$) is the probability of a hand becoming the best hand after those cards; and the negative potential ($NPot$) is the probability that a leading hand will lose at the end. The *Effective Hand Strength*[2] combines HS, $PPot$ and $NPot$ measures as follows:

$$EHS = HS^n(1 - NPot) + (1 - HS^n)PPot \ , \qquad (1)$$

where n means the number of opponents.

The *Pot Odds* (PO) determines the expected value of a hand when the player is faced with a bet. Be c the amount required to call the current bet/raise and p the amount currently in the pot, the PO is given by $\frac{c}{c+p}$.

[2]Many related works employ a more optimistic version of EHS in which $NPot = 0$ as they assume an aggressive player agent. We do not make this assumption here as our model aims to learn a wide range of players strategies.

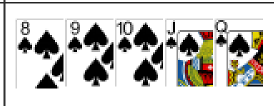
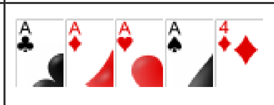
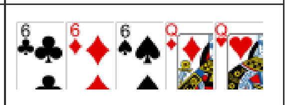
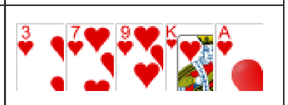
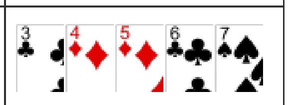
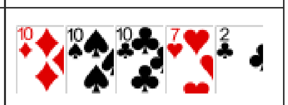
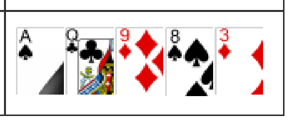
Name	Description	Example
Straight flush	Five cards in a sequence, all in the same suit.	
Four of a kind	All four cards of the same rank.	
Full house	Three of a kind with a pair.	
Flush	Any five cards of the same suit, but not in a sequence.	
Straight	Five cards in a sequence, but not of the same suit.	
Three of a kind	Three cards of the same rank.	
Two pair	Two different pairs.	
Pair	Two cards of the same rank.	
High Card	When you haven't made any of the hands above, the highest card plays.	

Fig. 2: Hand ranks in poker.

III. Model Description

This article aims to learn the strategies of a given player in Zoom poker games. In order to accomplish that, three steps are required: collect a great amount of games from the target player (Section III-A); formulate appropriately the problem of playing poker (Section III-B); and design a set of features able to analyze and detect the complex decision-making actions behind this problem (Sections III-C to III-F). For the sake of clarity, we refer to our target player as hero in this section.

A. Zoom Poker Data

The data adopted in this work consists of game logs from money games in an online casino. To be specific, it was collected by the hero himself from PokerStars[3] software. It contains thousands of game logs from games played against human players between the years 2015 and 2017 in Zoom six-handed tables of \$2 and \$5 (buy-in). Table II presents relevant information about the data. A point that attracts attention in the table is the high number of distinct opponents the hero played against considering the number of games. Indeed, it is high by the own characteristics of the Zoom. A cursory analysis of the data shows that the hero played in average eleven and two times against each opponent in the pre-flop and flop phases, respectively. Such an analysis emphasizes the great complexity of Zoom format, as many of the theories and methods proposed in literature for common poker games would not be effective given that opponent modeling strategies would have to deal with high scarcity of data. On the other hand, it also makes Zoom a very challenging problem for Artificial Intelligence research.

TABLE II: Metadata of the Zoom poker game logs.

Name	Game stats.	Hero stats.
#Games (also pre-flop phases)	182409	182409
#Flop phases	75312	26302
#Turn phases	48793	15656
#River phases	34665	10379
#Showdowns	24047	5663
#Distinct opponents	-	86557
Earnings	-	\$284,12
Expected Value (EV)	-	\$327,01

The game logs trace the history of actions made by the players during each one of the games. Fig. 3 shows an example about a poker hand: the first line registers an identification number for that hand; lines 2-3 inform modality, structure and format in which the poker game was played; lines 4-9 inform the position (seat 1 is the dealer), name and the amount of chips each player has, e.g., hero is in seat 6 and has \$4.49 in chips; the forced blinds are registered in lines 10-11 and then the hole cards are dealt to every player; the hero has “Ah Ad”, where h and d mean the heart and diamond suits (line 13); the first betting round then starts with players in seats 4 and 5 folding (lines 14-15); the hero raises, the player in seat 1 calls and other players folds (lines 16-19); the flop community cards then are dealt face up on the table “6h Th 3h”, the hero bets and the opponent folds, therefore, the hero collects the pot and does not show his hand (lines 21-25). As one can observe, collecting information from such a document is usually difficult because the file is in an unstructured format. In order to support our work, we have developed a range of C++ tools to collect and organize the data in structured format, such as shown by Fig. 3(b), which the meaning and relevance of each column (feature) is discussed later.

B. Problem Formulation

The problem addressed in this paper consists of modeling players strategies in the Zoom Texas Hold’em poker game. Despite we do not know any work in literature which deals with Zoom format, Texas Hold’em has been largely investigated. Many of such works consider the inherent division of the game to develop specific solutions for each round: pre-flop, flop, turn and river. Indeed, strategies tend to vary according to the level of information a player has. Therefore, an important advantage of such a formulation is the lower complexity of dealing with smaller problems.

[3] www.pokerstars.com

a) Poker Hand file

Hand #226
PokerStars Zoom Hand #149503216412: Hold'em No Limit ($0.01/$0.02) - 2016/02/27 11:34:37 ET
Table 'Halley' 6-max Seat #1 is the button
Seat 1: Yottab ($2.38 in chips)
Seat 2: TOM5T8 ($2.11 in chips)
Seat 3: vikt338 ($2.01 in chips)
Seat 4: PALAS15 ($2.20 in chips)
Seat 5: finn41 ($2.32 in chips)
Seat 6: **Hero** ($4.59 in chips)
TOM5T8: posts small blind $0.01
vikt338: posts big blind $0.02
*** HOLE CARDS ***
Dealt to **Hero** [Ah Ad]
PALAS15: folds
finn41: folds
Hero: raises $0.04 to $0.06
Yottab: calls $0.06
TOM5T8: folds
vikt338: folds
*** FLOP *** [6h Th 3h]
Hero: bets $0.10
Yottab: folds
Uncalled bet ($0.10) returned to **Hero**
Hero collected $0.14 from pot
Hero: doesn't show hand

b) Extracting pre-flop features

FC	SC	POS	SU	BR	CON	PA	TP	PO	BV	ACT
A	A	6	0	0	00010	1	1.5	0.4	1.0	3

Fig. 3: A game log of a poker hand: a) original unstructured format; and b) structured format. Note that the real names of the players have been changed.

On the other hand, that formulation can also have some disadvantages if each game round has too much (or too little) specific details to be modeled. In this case, the management of the smaller problems can become even more complicated than the bigger one. In order to take this into account, we propose an additional formulation which splits the problem in two sets instead of four. Despite each game round has its own characteristics, the differences between pre-flop and post-flop stages are very large. While the former presents a high number of players, no community cards and a small amount of money in dispute, the latter usually has few players and a much higher level of information about the game.

In summary, we formulate our poker problem in function of each game round (pre-flop, flop, turn and river) and in function of pre-flop and post-flop stages, which means the design of five smaller problems (data sets).

C. Feature Designing

In order to design an appropriate set of features, we define four major categories of information based on the poker literature [1], [3], [17] which encompasses distinct properties of the game to detect a wide range of player strategies. The categories are defined below:

1) **Hand Quality** provides a set of features able to assess the quality of a hand in function of different criteria.
2) **Position Insights** includes features which reflects some (dis)advantage related to the player position in the game. Such features play a key role in Texas Hold'em as who acts first also informs first his opponents [17].
3) **Aggressiveness** gathers a set of features which characterizes passive-aggressive players and also indicates promising bluff scenarios.
4) **Current Situation** provides a set of features which lists the more relevant events happened so far in the game: actions, money, etc.

Next we describe the set of features designed for pre-flop and post-flop rounds.

D. Pre-flop phase

The pre-flop phase is characterized by the low level of information, especially in Zoom poker where the amount of information about the opponents is scarse or even null. Another relevant point here is that our model should not be too specific about one or another kind of strategy, but simple and generic to be able to learn from players with very distinct strategies. Table III summarizes the set of features designed for the pre-flop round, which are discussed hereafter.

1) Hand Quality: The most relevant attributes in the pre-flop are FC, SC and SU as they describe the hero hand. By ignoring the suit of each card, we achieved a reduction from 1326 to 169 in the number of possible initial hands, without any loss of information (every suit has the same value in poker). The BR attribute indicates the presence of some high card and the CO describes the relation between both hole cards: cards in sequence (Connector, OneGapper and TwoGappers) or pairs (Pair) have value in pots against many players, while disconnected cards (Disconnect) not.

2) Position Insights: The POS feature shows the hero position in the table, which is a relevant information on this round. Many players use some strategies in which the position plays a major role in the decision to play a hand or not.

3) Aggressiveness: The PAPF feature provides general information about the passive-aggressive behavior of the opponents in the table by considering their actions.

4) Current Situation: Besides the Aggressiveness category, the PAPF feature also provides relevant information for this category as it maps the action of the opponents in the pre-flop. The TP and BV features, which indicate respectively the current pot size and the value of the last bet, gather fundamental information when the hero faces a bet but has a hand with great potential, and the PO feature denotes the expected value of a hand when the player is faced with a bet.

TABLE III: Summary of the features extracted for the pre-flop.

ID-Feat.Name	Description	Range
FC-FirstCard[4]	Value of the first hole card	{A,K,...,2}
SC-Sec.Card[4]	Value of the second hole card	{A,K,...,2}
SU-Suited	If the hole cards have the same suit	{0, 1}
BR-Broadway	If the hole cards contain any two broadway cards (A,K,Q,J or 10)	{0, 1}
CON-Connect	This class indicates how the hole cards are connected between them:	
	Connector: if the hole cards are in sequence (e.g., 78, QJ)	10000
	OneGapper: if the hole cards have a "hole" between them (e.g., 79, KJ)	01000
	TwoGappers: if the hole cards have two "holes" between them (e.g., 69, AJ)	00100
	Pair: if the hole cards have the same value (e.g., AA, 22)	00010
	Disconnect: if none of the above	00001
POS-Position	Position of the hero in table: SmallBlind(1), BigBlind(2), ..., Dealer(6)	{1,2,...,6}
PAPF-Prev.Act.PF	This class characterizes the action prior to the hero's action:	
	Unopened: if everyone have left the hand before the hero's turn	1
	Limper: if one player has paid the big blind and the others have left the hand	2
	Limpers: if two or more players have paid the big blind and the others have left the hand	3
	EpRaise: if any opponent has raised before the hero's turn	4
	EpRaiseAndCall: if any opponent has raised and one or more players have paid before the hero's turn	5
	LpRaise: if any opponent has raised after the hero's turn	6
	LpRaiseAndCall: if any opponent has raised after the hero's turn and one or more players have paid	7
	2Raise: if two or more players have raised and no player paid before the hero's turn	8
	2RaiseAndCall: if two or more players have raised and at least one player paid	9
TP-TotalPot	The amount of money in the pot (divided by the big blind value)	$\mathbb{R}_{>0}$
BV-BetVillain	The value of the villain's bet. If no one has raised, the first bet is the big blind	$\mathbb{R}_{>0}$
PO-PotOdds	The expected value of a hand when the player is faced with a bet	$\mathbb{R}_{>0}$
ACT-Act.Hero	The decision taken by the hero: Fold(1), Call(2), Raise(3) or Check(4)	{1,2,3,4}

The data set obtained from our data by extracting the features discussed in this section is named ZP-PreFlop.

E. Post-flop phases

The post-flop rounds have much more information available than the pre-flop one. Among the post-flop rounds such a difference is smaller. Therefore, our challenge here is to select another simple and generic set of appropriate features able to capture the particular characteristics of each round and also distinguish eventual strategies adopted among them. Table IV summarizes the set of features designed for the post-flop.

[4]For computer simulations, we transform {A,K,Q,J,T} to {14,13,12,11,10} as a preprocessing step (see Section III-F).

TABLE IV: Summary of the features extracted for the post-flop rounds.

ID-Feat.Name	Description	Range
EHS-Eff.Hand.Str.	The *Effective Hand Strength* combines hand strength and potential, Eq. (1)	[0,1]
POS-Position	Position of the hero in table: SmallBlind(1), BigBlind(2), ..., Dealer(6)	{1,2,...,6}
AG-AggressorPos	The player who made the last raise in the pre-flop	
	IPAgg: if hero plays before the aggressor	1
	HeroAgg: if hero is the aggressor	2
	OPAgg: if hero plays after the aggressor	3
IP-InPosition_Vs	Number of opponents who plays before the hero	{0,1,...,5}
OP-OutPosition_Vs	Number of opponents who plays after the hero	{0,1,...,5}
PRA-Prev.RoundAct.	The hero's action in the previous round: Check(1), Call(2), Bet(3) or Raise(4)	{1,2,3,4}
BSU-BoardSuit	If community cards have the same suit	
	Rainbow: if all cards have different suits	1
	TwoSuited: if two cards have same suit	2
	Monotone: if at least three cards have the same suit	3
BCA-BoardCards	If the board has community cards with the same value	
	NoPaired: if the board has no card with the same value	1
	Paired: if the board has two cards with the same value	2
	Triplet: if the board has three cards with the same value	3
BCON-BoardConnect	If the board has community cards in sequence	
	Connect: if the board has three cards in sequence (e.g., 567)	3
	SemiConnect: if the board has three cards in sequence with at most two "holes" (e.g., 753, J98)	2
	Disconnect: if none of the above	1
PA- Prev.Act	The action prior to the hero's action:	
	NoAction: if the hero is the first to play in that round	0
	Check: if every opponent before the hero played Check	1
	Bet: if any opponent has made a bet before the hero's turn	2
	BetAndCall: if any opponent has made a bet and at least one other has paid before the hero's turn	3
	BetAndRaise: if any opponent has made a bet and at least one other has raised before the hero's turn	4
RO-Round	The current round of the hand: flop(1), turn(2) or river(3)	{1,2,3}
TP-TotalPot	The amount of money in the pot (divided by the big blind value)	$\mathbb{R}_{>0}$
BV-BetVillain	The value of the villain's bet. If no one has raised, the first bet is the big blind	$\mathbb{R}_{\geq 0}$
PO-PotOdds	The expected value of a hand when the player is faced with a bet	$\mathbb{R}_{\geq 0}$
ACT-Act.Hero	The decision taken by the hero: Fold(1), Check(2), Call(3), Bet(4) or Raise(5)	{1,2,...,5}

1) Hand Quality: In the post-flop phases, we replace the card values by adopting the EHS feature, Eq. (1). Such a feature quantifies the strength of a hand compared to all other possible hands, taking into account also its potential to improve or deteriorate.

2) Position Insights: Besides the POS feature was already explained before, IP and OP features inform the number of opponents who acts before and after the hero, respectively. In addition, the hero usually is interested in the particular position of the aggressor (the player who made the last raise on the pre-flop). Thus, the AG feature indicates whether the aggressor is in position (acts later) against the hero or not.

3) Aggressiveness: The PRA feature describes the hero's action at the previous round, which is a fundamental information to do followed bluffs (Double/Triple Barrel Bluff). As some boards are more suitable to bluff than others, we have also modeled the BSU, BCA and BCON features to characterize the community cards, which can indicate the probability of the opponent having a hand of value.

4) Current Situation: The PA feature saves the action of the opponents in the current round, which make this feature also part of the Aggressiveness category. The RO feature indicates the current round, which is very useful in our second formulation (pre-flop and post-flop problems). TP, BV and PO features were already explained before.

By extracting the features designed above from our data, we obtained four data sets. For our first problem formulation, we have three data sets: ZP-Flop, ZP-Turn, ZP-River, which refers to each game round. For our second problem formulation, we obtain the ZP-PostFlop data set, which comprehends the whole post-flop stage.

F. Data Cleaning and Preprocessing

After extracting the features for each problem, we cleaned and preprocessed our data. Data cleaning here is related to the removal of ambiguous and duplicate objects. The former is a procedure that treats possible objects which have the same feature values but different classes and the latter a simple procedure to assure that the data sets do not have repeated objects. We defined the following criterion to eliminate ambiguous objects: keep the most frequent object and, in case of ties, that one which the class has less objects. Note that ambiguous objects can occur for several reasons, e.g., some wrong move, the change of the player strategy in the course of the games, etc. Table V shows the number of objects in each data set before and after cleaning the data. As expected, the ZP-PreFlop faces a big reduction ($\approx$ 79%) due to the low level of information in this round. By the contrary, post-flop data sets have a too small reduction ($\approx$ 1%) .

Many of our preprocessing steps are already represented in Tabs. III and IV. Indeed, the range of values in the table has been transformed from categorical to numerical values. In POS feature, for example, the small-blind position is converted to 1, the big-blind to 2, and so on. Another example is CON feature (Table III), in which we apply a one-hot encoding over the range $\{1,2,\ldots,5\}$ in order to obtain the range presented

TABLE V: Number of objects after the data cleaning.

Data set	Original	Cleaned
#ZP-Preflop	182409	37685
#ZP-Flop	26302	25983
#ZP-Turn	15656	15649
#ZP-River	10379	10342
#ZP-Postflop	52337	51980

in the table. The last step of our preprocessing, which is not presented in the tables, consists of applying a MinMax normalization over every feature (excepting the class) to assure that them lies on similar ranges, i.e., [0,1].

IV. Experiments

In this section, we conduct computer simulations to evaluate the data sets generated by our model. Section IV-A quickly describes the data, simulations and techniques; Section IV-B presents the results obtained on the data sets covering the pre-flop, flop, turn and river rounds and also the whole post-flop, i.e., the flop, turn and river rounds together.

A. Experimental setup

In the following we describe the conduction of the simulations, the machine learning techniques and the process of the parameter selection. A brief summary about the poker data sets is shown by Table VI, which presents the number of instances, features, classes and class distribution for each data set. One can see in the table that the classes are very imbalanced. For example, the Fold class occurs in more than 75% of the data in the ZP-PreFlop, while the Check class comprises around a half of the data in the ZP-PostFlop. Please refer to Section III-F for a description about the preprocessing of the data.

TABLE VI: Brief description of the poker data sets in terms of the number of data items (*#Obj.*), number of attributes (*#Attr.*), number of classes (*#Classes*) and class distribution.

Name	#Obj.	#Attr.	#Classes [Class Distribution]
ZP-PreFlop	37685	11	4 [75.7%, 12.7%, 9.1%, 2.5%]
ZP-Flop	25983	14	5 [9.2%, 48.4%, 8.2%, 32.6%, 1.6%]
ZP-Turn	15649	14	5 [8.4%, 49.6%, 11.0%, 29.5%, 1.5%]
ZP-River	10342	14	5 [11.7%, 52.9%, 8.9%, 24.8%, 1.7%]
ZP-PostFlop	51980	15	5 [9.4%, 49.6%, 9.2%, 30.2% 1.6%]

Each simulation is conducted by using a k-fold stratified cross-validation process, which splits the data set in k disjoint sets. In each run, k-1 sets are used as training data and 1 set is used as the test data, resulting in a total of k executions. In our study, the predictive performance of the methods is averaged over a repeated stratified cross-validation that averages five runs of 10-fold stratified cross-validation, taking the folds randomly each time.

For each of the data sets, we run the following supervised learning techniques: CART decision tree (DT), Random Forest (RF), k-Nearest Neighbors (kNN), Weighted k-Nearest Neighbors (WkNN), Naive Bayes (NB), Multilayer Perceptron

(MLP), Logistic Regression (LR) and Support Vector Machine (SVM). The parameters of each technique are selected through the grid search method by doing a 3-fold stratified cross-validation on each training partition (nested cross-validation). By doing this, we assure an unbiased learning as the test data are completely outside of the learning process. Following we list the parameters of each technique:

- DT has two parameters, the minimum number of samples required to split an internal node $m_{split} \in \{2, 5, 10, (1 * n)/100\}$ and the minimum number of samples required to be at a leaf node $m_{leaf} \in \{2, 5, 10, (1 * n)/100\}$, with n denoting the number of training data items;
- RF has one parameter, the number of trees in the forest $t \in \{2^1, 2^2, \ldots, 2^{10}\}$;
- kNN and WkNN have one parameter, the number of nearest neighbors $k \in \{1, 2, \ldots, 30\}$. The Euclidean has been defined as the distance metric for both methods;
- NB has no parameter. The likelihood of the features is assumed to be Gaussian;
- LR has two parameters, the norm used in the penalization $p \in \{l1, l2\}$ and the regularization strength $C \in \{2^2, 2^4, \ldots, 2^{14}\}$;
- MLP has two parameters, the initial learning rate $\alpha \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$ and the number of neurons in the hidden layer $n_h \in \{10, 20, 50, 100, 500, 1000\}$. ReLU has been defined as the activation function and the number of epochs has been fixed to 500;
- SVM has two parameters, the kernel coefficient $\gamma \in \{2^4, 2^3, \ldots, 2^{-10}\}$ and the penalty parameter $C \in \{2^{12}, 2^{11}, \ldots, 2^{-2}\}$. The radial basis function has been fixed as the kernel function and the stopping criterion has been defined as the Karush-Kuhn-Tucker violation to be less than 10^{-3}.

B. Results

Table VII shows the predictive performance of the eight techniques on the five poker data sets modeled in this paper. The results are provided in terms of weighted-F_1 in order to account for class imbalance. The first poker data set analyzed comprehends the pre-flop round, in which the player has to take an action after dealt two hole cards. The techniques achieved their overall best results in this data set, which shows that our simple and generic set of features is able to cover the player strategies, although it is not fixed to any specific strategy known in literature. RF registered the best results in the ZP-PreFlop data set, followed by SVM, MLP and DT.

On the post-flop data sets, in which three (flop), four (turn) and five (river) community cards are dealt face up on the table, the player has more information to take an action but the uncertainty about his opponents hands and future community cards make the decision-making much more complex. Table VII shows that on ZP-Flop, MLP achieves the best predictive performance; on the ZP-Turn, SVM does (with MLP close); and on the ZP-River, RF and SVM do (with MLP close again). Despite the results on the post-flop are not similar to the pre-flop, they are very attractive given the complexity of these rounds, which is even higher in the Zoom format. Another interesting point is that the worse results of every technique occurred when predicting the player actions on the turn stage, while the river presented the best post-flop results for the overall techniques. The reasons about why modeling player strategy on turn is harder than other stages may be related to some risk the player eventually takes by expecting some specific community card(s).

The fifth poker data set in Table VII is the ZP-PostFlop, which gathers the data from the flop, turn and river rounds together. Such an approach has the advantage of training a unique learning model which is able to predict the player actions along the whole post-flop. One can see in the table that the overall performance of the techniques in the ZP-PostFlop data set is consistent in average with their performance in each particular data set. Moreover, some techniques perform very well on this data set, e.g., MLP, which achieves the best result. We believe these results are possible because our modeling of the player strategies contains relevant information about every stage of the post-flop game, supporting the methods in the prediction of player actions through the analysis of the relations among the features.

Now we move on to analyze statistically the results presented in Table VII. The Friedman test has been selected as it permits the comparison among multiple techniques over multiple data sets [18]. Firstly, the average ranks related to the predictive performance of each technique are calculated (see "Avg.Rk." in the table). The Friedman test is then applied to determine if there is any statistically significant difference among the algorithms. The null-hypothesis states that all the algorithms are equivalent, therefore, their ranks should be equal. Under the significance level α at 0.1, the null-hypothesis is rejected, which means that at least one of the methods differs from the rest. Following we proceed with the Nemenyi post-hoc test considering again a significance level α at 0.1. The test indicates that the predictive performance of SVM and MLP outperform LR and NB, this latter also outperformed by RF. The result of the Nemenyi test is shown by Fig. 4.

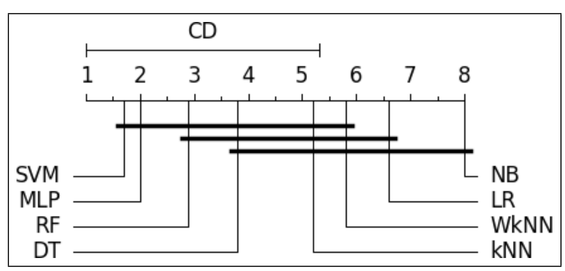


Fig. 4: The statistical significance diagram, which presents the critical difference found by the Nemenyi post-hoc test.

The statistical tests attest SVM, MLP and RF as the better techniques on our Zoom data sets. However, it is also important to observe DT results. Despite slightly worse than those techniques, DT achieved competitive performance in every data set with much lower time and space complexity. On the other hand, LR and NB obtained the worse results.

TABLE VII: Predictive performance of the supervised learning techniques on the modeled Zoom poker data sets. The results are provided in terms of weigthed-F_1 in order to account for class imbalance.

Algs.	**ZP-PreFlop**	**ZP-Flop**	**ZP-Turn**	**ZP-River**	**ZP-PostFlop**	Avg.Rk
DT	92.0 $\pm$ 0.4	77.1 $\pm$ 0.7	74.0 $\pm$ 1.0	80.3 $\pm$ 1.3	76.5 $\pm$ 0.5	3.8
RF	**92.8 $\pm$ 0.3**	76.6 $\pm$ 0.8	73.9 $\pm$ 0.8	**80.9 $\pm$ 1.1**	76.7 $\pm$ 0.5	2.9
kNN	90.3 $\pm$ 0.4	77.2 $\pm$ 0.7	71.2 $\pm$ 1.1	74.0 $\pm$ 1.4	75.0 $\pm$ 0.6	5.2
WkNN	90.0 $\pm$ 0.3	76.5 $\pm$ 0.8	71.7 $\pm$ 0.9	74.9 $\pm$ 1.4	74.8 $\pm$ 0.5	5.8
NB	79.7 $\pm$ 0.6	71.0 $\pm$ 0.9	66.0 $\pm$ 1.1	71.3 $\pm$ 1.6	69.1 $\pm$ 0.5	8.0
LR	82.2 $\pm$ 0.4	72.3 $\pm$ 0.8	69.4 $\pm$ 1.0	75.0 $\pm$ 1.1	69.9 $\pm$ 0.4	6.6
MLP	92.3 $\pm$ 0.6	**78.5 $\pm$ 0.7**	74.5 $\pm$ 1.2	80.8 $\pm$ 1.5	**78.0 $\pm$ 0.6**	2.0
SVM	92.5 $\pm$ 0.4	77.4 $\pm$ 0.7	**74.6 $\pm$ 0.9**	**80.9 $\pm$ 1.2**	77.6 $\pm$ 0.5	1.7

In common, both are the unique linear techniques evaluated in this study. By the results, it seems fair enough to say that non-linear techniques have some advantage over linear ones, possibly the non-linear relation among the features. However, such a statement requires further investigations. Finally, in order to provide a challenging test-bed for machine learning research, our modeled data sets are available online at www.facom.ufu.br/~murillo/zpdata.html .

V. Conclusion

In this article, we addressed the problem of modeling the actions of a human poker player in order to learn his strategies from his past game logs in the game of Zoom No Limit Texas Hold'em. Zoom is a new format of game where, instead of playing in a specific table against a specific set of opponents, a player is placed in a large pool of players in which their opponents change every hand. To handle the problem and the scarcity of reads from the opponents, we modeled a simple and generic set of features able to capture a wide range of players strategies considering each one of the four rounds of the game and also the pre-flop and post-flop stages. As a consequence of our modeling, we generated five data sets which were evaluated by eight machine learning techniques. The experimental results showed that much of the player strategies was effectively learned, especially by MLP and SVM techniques. They also revealed that the overall performances of non-linear techniques were better than the linear ones. In addition, despite players seem more selective with their hole cards in Zoom format, the possibly smaller number of opponents in the post-flop rounds hardly can be exploited by opponent modeling methods. To deal with such a problem, our modeling covers distinct properties of a hand (hand quality, position insights, aggressiveness and current situation) in order to map the player actions through the analysis of the relations among the features.

Further works can be divided into three major topics: feature design, in which we intend to analyze the salience of our current features and to include others relevant features in our model; comparative analysis of players strategies between poker hands played in both Zoom and regular format; and the study of other machine learning methods (e.g., deep learning) in order to improve the overall performance on the problem.

Acknowledgment

The authors thank the support given by the Minas Gerais State Research Foundation - FAPEMIG.

References

[1] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron, "The challenge of poker," *Artificial Intelligence*, vol. 134, no. 1-2, pp. 201–240, 2002.

[2] J. Rubin and I. Watson, "Computer poker: A review," *Artificial intelligence*, vol. 175, no. 5-6, pp. 958–987, 2011.

[3] D. Sklansky and M. Malmuth, *Hold'em poker for advanced players*. Two Plus Two Publishing LLC, 1999.

[4] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron, "Approximating game-theoretic optimal strategies for full-scale poker," in *International Joint Conference on Artificial Intelligence*, 2003, pp. 661–668.

[5] J. Rubin and I. Watson, "Case-based strategies in computer poker," *AI communications*, vol. 25, no. 1, pp. 19–48, 2012.

[6] X. Li and R. Miikkulainen, "Evolving adaptive poker players for effective opponent exploitation," in *AAAI-17 Workshop on Computer Poker and Imperfect Information Games*, 2017, pp. 353–360.

[7] G. Van den Broeck, K. Driessens, and J. Ramon, "Monte-carlo tree search in poker using expected reward distributions," in *Asian Conference on Machine Learning*. Springer, 2009, pp. 367–381.

[8] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione, "Regret minimization in games with incomplete information," in *Advances in neural information processing systems*, 2008, pp. 1729–1736.

[9] N. Brown, S. Ganzfried, and T. Sandholm, "Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit texas hold'em agent," in *International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 7–15.

[10] M. Moravčík, M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling, "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017.

[11] N. A. Risk and D. Szafron, "Using counterfactual regret minimization to create competitive multiplayer poker agents," in *International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 159–166.

[12] L. F. Teófilo, L. P. Reis, and H. L. Cardoso, "A profitable online no-limit poker playing agent," in *International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, 2014, pp. 286–293.

[13] U. Johansson, C. Sonstrod, and L. Niklasson, "Explaining winning poker–a data mining approach," in *International Conference on Machine Learning and Applications*, 2006, pp. 129–134.

[14] D. Billings, D. Papp, J. Schaeffer, and D. Szafron, "Opponent modeling in poker," in *AAAI Conference on Artificial Intelligence*, 1998, pp. 493–500.

[15] A. Davidson, D. Billings, J. Schaeffer, and D. Szafron, "Improved opponent modeling in poker," in *International Conference on Artificial Intelligence*, 2000, pp. 1467–1473.

[16] M. Johanson and M. Bowling, "Data biased robust counter strategies," in *Artificial Intelligence and Statistics*, 2009, pp. 264–271.

[17] D. Sklansky, *The theory of poker*. Two Plus Two Publishing LLC, 1999.

[18] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.