Towards a Time-Aware Hidden Markov Model for the Truco Game.

Joaquim Assunção[1], **Gustavo Bathu Paulus**[2]

[1]Universidade Federal de Santa Maria (UFSM)
97105-900– Santa Maria – RS – Brazil

[2]Instituto Federal do Rio Grande do Sul (IFRS)
98200-000–Ibirubá – RS – Brazil

joaquim@inf.ufsm.br, gustavo.paulus@ibiruba.ifrs

Abstract. *Similar to Poker, the game of Truco has challenges for Artificial intelligence. Considering a large number of game states, a scenario characterized by partial visibility, stochastic behavior, and score susceptible to bluff; this game offers a good set of rules to test and improve AI techniques. In this article, we describe the creation of a Hidden Markov Model (HMM) agent using temporal control. The model has an embedded vector that adjusts its probabilities for further game actions, consequently, improving the model playing performance. The evaluation is given with over 210,000 matches, serving as empirical proof of the idea.*

Keywords— *HMM, time-aware, games, Truco*

1. Introduction

Games are perfect experimental fields for the construction, testing, and consequent evolution of Artificial Intelligence (AI). They are well defined, easy to understand, and many solved challenges are often similar to problems encountered in the real world. Recognized AI contributions to game projects enabled the development of new intelligent techniques, which are effective in solving real-world problems.

According to [Niklaus et al. 2019], implicit information, combined with stochastic aspects, allows card games to simulate challenges presented in different decision scenarios with a partial view. According to [Rubin and Watson 2012], the random distribution of cards, and the partial vision of the game state, make it difficult to create the game trees, e.g., Minimax an AI approach well known for many solving many games such as Chess.

A few works used Case-Based Reasoning (CBR) [Richter and Weber 2013] as an AI framework to solve problems in the game of Truco [Paulus et al. 2019, Moral et al. 2020, Vargas et al. 2021], such approach relies on creating a database of cases for the game, latter exploring the data with different approaches. Specifically, [Paulus et al. 2019] did a major comparison between shallow learning approaches, [Vargas et al. 2021] built a database focusing on bluff and [Rossato et al. 2020], without using CBR, built a lightweight Markov model using the natural force of all possible hands.

This work focuses on improving the model described in [Rossato et al. 2020] by using an HMM instead of a Markov chain (see Section 3), and further enhancing the model (hybrid) with a time tracking parameter aiming to store bluffs and reuse it in a future turn (Section 4) similar to [Vargas et al. 2021], except the problem is solved within the model instead of creating a database for cases. The model weights and probabilities are based on a uniform distribution according to the strength of the hands, making it a direct probabilistic approach. Going further, to prove the idea, we made 210,000 matches testing the model and the combination of different values for its parameters (Section 5).

2. Background

Similar to Poker, Truco involves cognitive challenges related to the partial vision of the opponent's cards and the randomness of the card draw. However, unlike Poker, Truco has several decision-making stages due to the multiple game interactions found in the different hands played (see Table 1). In this case, Truco matches are divided into hands where two or more opponents interact.

Table 1. Bets and raises.

Bet	1st^{a} Raise	2nd^{a} Raise
Envido	Real Envido	Falta Envido
Envido	Falta Envido	-
Real Envido	Falta Envido	-
Falta Envido	-	-
Flor	Contra Flor	Contra Flor e Resto
Truco	Retruco	Vale 4

In particular, this work only explores matches between two players. Truco uses 40 cards from a Spanish deck, in each playing hand, the cards are shuffled and dealt, where each player is dealt three cards. To play a hand, each player receives three cards. The game is split into several hands, where it usually ends when a player reaches X points (can be 9 to 30, depending on the variation). As a turn-based game, players take turns to play cards and place bets in the hand dispute, where the winner of a turn starts the next turn of play in that hand. To win a hand, the player must win at least two of the three turns of the played hand. For each turn of one hand, different game scenarios are presented.

The main actions of this game in Truco can be divided into "playing cards" and "making bets". Bets are placed to increase the number of points won in the game. Bets can be made to increase winnings when the player has good cards or to win with bad cards by bluffing. There are two types of possible bets: the *envido* and the *truco* (same name of the game). For all intents, in a pure probabilistic Markov model, both modes can be modeled similarly, using all the possible states and the transition probabilities according to the difference in the states' (hands) strength.

3. HMM and our implementation

A Markovian model is a special type of stochastic process that can be applied to almost any system [Stewart 2009]. Such models can represent states of a system, with assigned transition probabilities for all the states. The system can only occupy one state at time and the probability of being in a current state only depends on the previous state. Formally, we can say that the process can be represented by a stochastic variable in time $\{x^{(t)}, t \in T\}$, being $x^{(t)}$ the occupied state in a given time t.

A Hidden Markov Models is a two part, state-dependent model, where the hidden part is the set of states describing the system, linked by transition probabilities, and the observable part is achieved through emission probabilities after the hidden state is chosen [Zucchini et al. 2017]. In this work, the states of a hidden part are the set of all possible hands of *envido* against all possible hands of *envido*. The emission, observable part, is an action of bet, accept a bet, deny a bet, or accept and raise a bet.

As the invisible part is represented by a set of N variables changing at T times, we used these time changes and recorded it in a separated vector. Then, we embedded these vectors in the hidden part of the HMM transition. Such a simple change is enough to record behaviors in time and emit an expected (i.e., better move) observable output.

4. Proposed Model

Our model is a direct extension of the model proposed in [Rossato et al. 2020]. This work uses the state space, of each mode of the game, mapped into Markov chains that hold a set of transition probabilities according to the strength of the cards. For instance, the *truco* mode has a set of 9880 possible states ($C(40,3) = \frac{40!}{(40-3)!*3!}$). Thus, it is represented by a Markov with 9880 states and transitions among these states. The second mode, *envido*, has a set of 560 possible states ($C(14,3) = \frac{(14+3-1)!}{(14-1)!*3!}$), thus, a Markov of 560 states.

Despite the apparent difficulty to map these states, it is a straightforward process once the cards and their values are mapped. Regarding the number of transitions, it is dramatically decreased each time a bot/player plays a card, thus the search space in the transition probability matrix is often small enough for real-time computation. The *envido* mode is especially interesting because most of the states are only different when playing the full game, or they are a collection of only bluffs play. Thus, for the propose of tests, we have 225 states considering any score lower than 20 as one state.

As in [Rossato et al. 2020], all the states have their transitions values uniformly distributed, stronger the hand more likely to play, except for the following cases, which are fixed.

- Any state with 33 points to any state with < 20 points: 0;
- Any state with X points to any state with X points: 0.5;
- Any state with < 20 points to any state with 33 points: 0;

Our model improves on the original by adding an emission to different actions. Thus, each of the three observable states has its own weight; *Raise* is the most difficult to happen, having the weight transition multiplied by itself. For instance, a hand with $\{1\clubsuit, 2\clubsuit, 10\heartsuit\}$ has 28% to accept an *envido* call (without knowing any card of the opponent. However, the probability to raise is only 8% (0.28*0.28). On the other hand, denying a bet happens every time the probability of acceptance is not fulfilled.

However, any agent modeled with such structures has a problem not approached in any know work [Paulus et al. 2019, Moral et al. 2020, Vargas et al. 2021, Rossato et al. 2020]; as the game rolls the personality of the player or the approach of the agent, can be learned. Such a task is easy for humans but not so for AI. In an attempt to improve on this regard, we used a temporal vector, attached to the model, which changes the future decision (in the same game) based on how much an opponent agent uses to bluff. This vector is described by τ in the model (Figure 1).

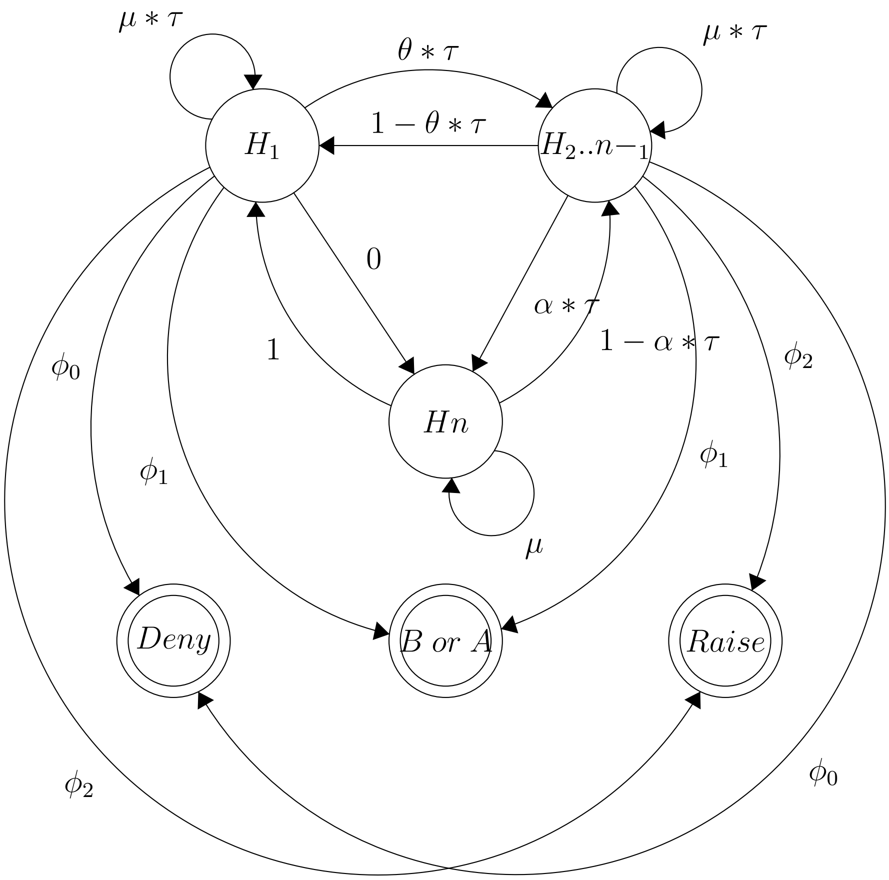

Figure 1. Illustration of the HMM for the Truco game

The model parameters are the following:

- H_x represents a local state. A unique hand of cards for the *Truco* or *Envido* mode. *Envido* has $n = 15$.
- H_n is the strongest game, for both modes, and H_1 is the weakest. Hence, the probabilities 1 and 0.
- μ is the average weight for any game, i.e., 50% chance to accept a call.
- σ, α and θ are vectors with the transition probabilities to the correspondent hand.
- $1 - \sigma$, $1 - \alpha$, and $1 - \theta$ are the complementary values for the vectors σ,α and θ.
- τ is the vector holding changes according to the opponents' bluff. It is set to 1 in a new game and changes in the game, altering the transition probabilities in future hands.
- ϕ are the emission probabilities. They are tied to the hidden transition probabilities. ϕ_1 receives the default emission probability. ϕ_2 receives $\phi_1^{\phi_1}$, ϕ_0 receives the remaining probability.

5. Performance and conclusion

The model was tested on over 210,000 matches against the model proposed in [Rossato et al. 2020]. Despite the original proposal, we also created two small variations of the original Markovian agent. The first, played fair (Table 2,"honest"), never betting with weak hands (≤ 22 points). The second, being the opposite, bluffing half of the time (Table 2, "Half time"). The temporal HMM already has a small advantage ($\approx 1\%$ win ratio) by using different probabilities to emit the states (bet > accept > raise). However, it played considerably better against the default bot, proposed in [Rossato et al. 2020] (win ratio of $\approx 56\%$).

Against the honest agent, we detect no meaningful ($> 1\%$) change. However, against the default agent, it is already advantageous to use the temporal bluff control. The 210,000 matches show clearly that the ratio of wins gets higher until around +2 per bluff (see Table 2). Regarding the hybrid agent, which uses fixed rules to bluff half time, the temporal parameter increased its win ratio once more, proving the parameter a successful idea. Playing a 10,000 matches against the Half Time bot without, using the τ parameter, served as an experiment control and showed us that to bluff is advantageous since the score was near equal, a small improvement over the original. On the other hand, the idea of decreasing the temporal parameter when the player makes an honest play seems to have little impact, which is open to further investigation.

Table 2. Simple Markov bot vs the temporal agent. Tests with the bluff temporal (τ) parameter. Each test was performed with 10,000 games.

$\tau+$	$\tau-$	Bluff	Temporal agent wins	$\tau+$	$\tau-$	Bluff	Temporal agent wins
0.2	0	Honest	5139	1.3	-0.2	Default	5586
0.2	0	Default	5213	1.3	-0.6	Default	5501
0.4	0	Default	5271	1.6	-0.2	Default	**5678**
0.6	0	Default	5451	1.6	-0.6	Default	5578
0.8	0	Default	5440	2	-0.2	Default	5679
1	0	Default	5463	2	-0.6	Default	5682
1.3	0	Default	5570	1.6	0	Half time	**5600**
1.6	0	Default	**5655**	1.6	-0.2	Half time	**5682**
2	0	Default	**5671**	2	-0.2	Half time	**5854**
2.5	0	Default	**5645**	2.5	-0.2	Half time	**5831**
				0	0	Half time	5008

After 210,000 matches, it is clear that the time constraint works as expected in our initial model design, especially when a player uses bluff too often. Due to the lack of implementation and compatibility with others, we could not test the full game, but only the *envido* mode, which can be tested directly using the achieved points. Nonetheless, the approach described in [Rossato et al. 2020] is the same for both *envido* and *truco* mode; thus, there is no reason for the performance to be too far from these results.

In the final regard, we highlight that the default temporal HMM behavior does not help with such situations. Equations such as Chapman–Kolmogorov can be useful to get the next step and the steady-state probabilities, but they cannot be directly used to boost an action according to the opponent's behavior. However, the real-time changing vector τ can be used seamlessly within the model, storing probability changes according to the opponents' actions. In this work, these probabilities were used to respond properly against a bluff, however, it is possible to use in other scenarios in the game or even in other applications.

6. Future works

One future work is integrating the existing agents to play together, further improving our knowledge about the performance of different techniques and implementations. Furthermore, most works only focus on testing agents playing other agents. Thus, it would be relevant to test these agents against human players of different skill levels.

Regarding training and competition, we aim at the integration with mobile apps so we could test against a large number of users and use the data to improve the model. Furthermore, with large amounts of real-world data, we could use our knowledge to model and test deep learning approaches such as Long short-term memory (LSTM) networks, deep Q-networks or even hybrid models.

References

Moral, R. C., Paulus, G. B., Assunção, J. V., and Silva, L. A. (2020). Investigating case learning techniques for agents to play the card game of truco. In *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 107–116. IEEE.

Niklaus, J., Alberti, M., Pondenkandath, V., Ingold, R., and Liwicki, M. (2019). In *Survey of Artificial Intelligence for Card Games and Its Application to the Swiss Game Jass*, pages 25–30.

Paulus, G. B., C. Assuncao, J. V., and Silva, L. A. L. (2019). Cases and clusters in reuse policies for decision-making in card games. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1361–1365.

Richter, M. and Weber, R. (2013). *Case-Based Reasoning: A Textbook*. Springer Berlin Heidelberg.

Rossato, L., Silva, L., and Assunção, J. (2020). A markovian model for the game of truco. In *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*.

Rubin, J. and Watson, I. (2012). Case-based strategies in computer poker. *AI communications*, 25(1):19–48.

Stewart, W. J. (2009). *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton University Press.

Vargas, D. P., Paulus, G. B., and Silva, L. A. (2021). Active learning and case-based reasoning for the deceptive play in the card game of truco. In *Brazilian Conference on Intelligent Systems*, pages 313–327. Springer.

Zucchini, W., MacDonald, I., and Langrock, R. (2017). *Hidden Markov Models for Time Series: An Introduction Using R, Second Edition*. Chapman & Hall/CRC Monographs on Statistics and Applied Probability. CRC Press.