DOI: 10.48011/asba.v2i1.1732

Aplicação do Aprendizado por Reforço no Blackjack: Estudo de Caso de Estimação de Parâmetros

Heitor Magno Rodrigues Junior *
André Luiz Carvalho Ottoni **

* *Programa de Pós-Graduação em Engenharia Elétrica (PPEE), Universidade Federal de Juiz de Fora (UFJF), MG, (e-mail: heitormrjunior@gmail.com).*
** *Centro de Ciências Exatas e Tecnológicas (CETEC), Universidade Federal do Recôncavo da Bahia (UFRB), BA, (e-mail: andre.ottoni@ufrb.edu.br)*

Abstract: This work applies the Reinforcement Learning technique in the domain of the Blackjack card game in order to estimate the parameters of the learning rate (α) and discount factor (γ), in order to maximize the performance of the algorithm in the card game. 64 combinations of parameters are tested and the definition of the best combination is obtained through the statistical technique of Analysis of Variance (ANOVA) and the use of the Scott-Knott method (SK). The estimated combination was compared with adopted parameters from other works and had the best performance, with an average number of wins and draws greater than the number of losses.

Resumo: Este trabalho aplica a técnica de Aprendizado por Reforço (AR) no domínio do jogo de cartas *Blackjack* com o intuito de estimar os parâmetros de taxa de aprendizado (α) e fator de desconto (γ), de modo a maximizar o desempenho do algoritmo no jogo de cartas. São testadas 64 combinações de parâmetros e a definição da melhor combinação é obtida através da técnica estatística de Análise de Variância (ANOVA) e do uso do método Scott-Knott (SK). A combinação dos parâmetros estimada foi comparada com parâmetros adotados em outros trabalhos da literatura e obteve o melhor desempenho, com um número médio de vitórias e empates maior do que o número de derrotas.

Keywords: Reinforcement Learning; Blackjack; Parameter Estimation; Machine Learning; AI in Games;

Palavras-chaves: Aprendizado por Reforço; Blackjack; Estimação de Parâmetros; Aprendizado de Máquina; IA em Jogos; .

1. INTRODUÇÃO

O Aprendizado de Máquina, em inglês, *Machine Learning (ML)*, é um campo de estudo de grande relevância (Russell and Norving, 2013; Silva et al., 2016; Hutter et al., 2019). De fato, as aplicações do ML são diversas, como por exemplo em identificação de sistemas (Nepomuceno, 2019), mineração de dados (da Silva et al., 2017), otimização (Ottoni et al., 2018), previsão de séries temporais (Silva et al., 2016), reconhecimento de padrões (Silva et al., 2016) e tomada de decisão (Russell and Norving, 2013). De acordo com Russell and Norving (2013), o ML pode ser classificado em três áreas: Aprendizado Supervisionado, Aprendizado Não-Supervisionado e Aprendizado por Reforço.

O Aprendizado por Reforço (AR) é fundamentado nos Processos de Decisão de Markov (Watkins and Dayan, 1992; Russell and Norving, 2013; Sutton and Barto, 2018). Além disso, no AR um agente aprende interagindo com um ambiente. Basicamente, o aprendizado ocorre a partir da repetição dos seguintes passos: (i) o agente observa as condições do ambiente (estado); (ii) seleciona/executa uma ação, e; (iii) recebe um reforço (Sutton and Barto, 2018).

Na literatura, o AR possui aplicações em distintos domínios, como na Robótica (Kober et al., 2013), Sistemas Multiagentes (Da Silva and Reali Costa, 2019) e Otimização Combinatória (Ottoni et al., 2020). Outro campo de aplicação do AR concentra-se na área de Inteligência Artificial para Jogos (Mnih et al., 2015; Silver et al., 2018; Ramirez et al., 2019). Certamente, a capacidade de um agente aprender com recompensas e penalidades agindo em um ambiente é um fator que contribui para viabilizar a utilização em *games*. No entanto, a literatura ainda carece de pesquisas que avaliem os efeitos da definição dos parâmetros do AR em aplicações em jogos, mais especificamente para domínios de jogos de cartas, como *Blackjack* (Pérez-Uribe and Sanchez, 1998; Kakvi, 2009; Gan et al., 2019).

De fato, um dos principais aspectos do ML e também do AR é a estimação de parâmetros, como taxa de aprendi-

zado, fator de desconto, $\epsilon - greedy$ e função de reforço (Schweighofer and Doya, 2003; Even-Dar and Mansour, 2003; Barsce et al., 2017; Ottoni et al., 2018; Liessner et al., 2019; Hutter et al., 2019). Nesse aspecto, o desafio é propor métodos para a otimização e recomendação de parâmetros, de modo a otimizar o desempenho do aprendizado (Hutter et al., 2019). Uma das linhas adotadas consiste na utilização de métodos estatísticos para ajuste dos parâmetros do AR, como proposto em Ottoni et al. (2020).

Dessa forma, o objetivo deste trabalho é aplicar o AR e estimar parâmetros para o domínio do *Blackjack*. Para isso, foi adotado um importante algoritmo de AR, o *Q-learning* (Watkins and Dayan, 1992). Também foi proposto um modelo de AR para o *Blackjack*, baseado em estados, ações e recompensas, com algoritmo denominado RL-Blackjack. Além disso, foram aplicados os métodos de Análise Variância (ANOVA) (Montgomery, 2017) e Scott-Knott (Scott and Knott, 1974) para recomendação dos melhores parâmetros do AR para para o *Blackjack*.

Este trabalho está organizado em seções. A Seção 2 apresenta aspectos fundamentais do Aprendizado do Reforço e do jogo de cartas *Blackjack*. Detalhes do desenvolvimento do sistema de AR proposto são descritos na Seção 3. As Seções 4 e 5, por sua vez, apresentam os experimentos realizados e resultados, respectivamente. Finalmente, a Seção 6 destaca as conclusões deste trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Aprendizado por Reforço

O Aprendizado por Reforço (AR) é fundamentado nos Processos de Decisão de Markov (Russell and Norving, 2013; Sutton and Barto, 2018). Nesse aspecto, o AR é estruturado em: estados, ações, reforços e transições de estados. O objetivo é aprender uma política π que maximiza as recompensas pela tomada de decisão (ações) nas situações do ambiente (estados).

Um importante algoritmo de AR é o *Q-learning* (Watkins and Dayan, 1992; Sutton and Barto, 2018). O *Q-learning* utiliza de uma matriz de aprendizado Q para armazenar o conhecimento aprendido ao logo do tempo. Para cada execução de uma ação em um determinado estado, a matriz Q é atualizada, conforme Equação (1) (Watkins and Dayan, 1992):

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma Q_{max_a}(s',a) - Q(s,a)] \quad (1)$$

em que, a é a ação executada no instante t; s é o estado observado no instante t; s' é novo estado em $t+1$; $Q(s,a)$ é o valor armazenado na matriz Q para o par (s, a); r é a recompensa imediata no instante t; $Q_{max_a}(s',a)$ é maior valor armazenado na matriz Q referente a linha do novo estado s'; α e γ são os parâmetros, taxa de aprendizado e fator de desconto, respectivamente. O Algoritmo 1 apresenta a sequência de execução do *Q-learning*.

No Algoritmo 1, a política $\epsilon - greedy$ (parâmetro ϵ) é responsável por controlar a aleatoriedade na seleção de ações. Outros parâmetros do *Q-learning* são a taxa de aprendizado (α) e o fator de desconto (γ). A taxa de aprendizado ($0 < \alpha \leq 1$) regula a velocidade em que as novas informações sobrepõem-se sobre o aprendizado já armazenado na matriz Q. Já o fator de desconto tem o papel de controlar a influência das recompensas futuras: se $\gamma = 0$, o reforço imediato tem grande influência; se $0 < \gamma \leq 1$, as recompensas futuras são descontadas; se $\gamma = 1$, as recompensas não são descontadas. Nesse aspecto, um desafio do AR é configurar os parâmetros de aprendizado de forma a maximizar o desempenho no domínio em estudo, pois ϵ, α e γ podem assumir diferentes combinações de valores (Schweighofer and Doya, 2003; Even-Dar and Mansour, 2003; Barsce et al., 2017; Ottoni et al., 2018).

```
1.  Para cada (s,a) inicialize Q(s,a)=0;
2.  Observe  o estado s;
3.  Repita até o critério de parada ser satisfeito
4.      Selecione a ação a usando a política e-greedy;
5.      Execute a ação a;
6.      Receba a recompensa imediata r;
7.      Observe o novo estado s';
8.      Atualize o item Q(s,a) de acordo com a Eq. (1);
9.      s = s';
10. Fim Repita
```

Algoritmo 1: Q-learning.

2.2 Regras Básicas do Blackjack

O jogo de cartas *Blackjack* é jogado por um ou mais jogadores, de forma independente, contra o *dealer*, que é o responsável pela distribuição das cartas e organização do jogo. A cada rodada, o propósito do jogador é formar uma combinação de cartas em sua mão que vence a combinação de cartas do *dealer* (Snyder, 2013).

Uma rodada se inicia após os jogadores fazerem apostas. Depois disso, cada jogador recebe duas cartas com faces viradas para cima, enquanto o *dealer* também recebe duas cartas, mas apenas uma delas é revelada aos jogadores. O objetivo de cada jogador é obter cartas em sua mão cuja soma seja o mais próximo possível a 21, não podendo exceder esse valor. O valor das cartas é igual ao respectivo número, exceto para as cartas Valete (J), Dama (Q) e Rei (K), que valem 10, e a carta Ás (A), que pode valer 1 ou 11. Caso a soma das cartas de um jogador ultrapasse 21, ele perde a rodada, assim como a aposta realizada. Caso as duas cartas iniciais do jogador somem 21, ocorrendo um *Blackjack*, o jogador vence a rodada e recebe 2,5 vezes o valor da aposta inicial.

Enquanto a soma das cartas não ultrapassa 21, um jogador pode requisitar ao *dealer* que seja acrescentada mais uma carta aleatória do baralho à sua mão. Essa ação é conhecida como comprar (ou *hit*). Caso o jogador esteja satisfeito com as cartas que possui, ele pode optar por não receber mais cartas e aguardar o final da rodada, o que é também conhecido como parar (ou *stand*).

Depois de todos os jogadores terem feito suas jogadas é a vez do *dealer* jogar conforme uma regra fixa pré-estabelecida, que pode variar de acordo com o local em que o jogo ocorre, mas geralmente consiste em comprar mais cartas quando a soma de sua mão é 16 ou menos, e parar quando essa soma é 17 ou mais. Caso a soma da mão do *dealer* ultrapasse 21, todos os jogadores que optaram por

parar vencem a rodada e recebem o dobro da aposta inicial. Caso a soma da mão do *dealer* não ultrapasse 21, vencem a rodada e recebem o dobro da aposta inicial apenas os jogadores que possuem mãos com somas mais altas do que o *dealer*. Caso o valor da mão do jogador seja igual à mão do *dealer*, tem-se um empate e o jogador recebe sua aposta de volta. Caso contrário, o jogador perde a rodada, assim como a aposta inicial.

3. SISTEMA DE APRENDIZADO POR REFORÇO

3.1 Definição de Estados e Ações

Diante da dinâmica do jogo apresentada na Seção 2.2, foram definidos os estados e ações a serem considerados para aplicação do AR. Foram estabelecidas, portanto, duas possíveis ações para a máquina: comprar ou parar. A definição dos possíveis estados foi baseada nas possíveis situações de jogo que o agente poderia estar sujeito. Para simplificar a análise, o conhecimento inicial de uma das cartas do *dealer* foi desconsiderado, o que fez com que essa variável não interferisse no aprendizado da máquina.

A definição dos estados é feita com base na soma das cartas da mão do agente, que pode ir de 2, caso o agente receba dois Ases, até 21. No entanto, a estratégia de considerar a carta Ás valendo 1 é utilizada somente quando o limite 21 é ultrapassado caso essa carta fosse considerada com valor igual a 11. Por exemplo, caso um jogador tenha uma mão com Ás e 5, a soma considerada é de $11+5=16$; caso ele opte por comprar mais uma carta e receba um 7, então ele deve considerar a carta Ás com valor igual a 1 para que o limite não seja estabelecido e sua mão valha 13 $(1+5+7)$. A partir disso, tem-se que a soma das cartas da mão do agente pode ir de 4, caso o jogador receba inicialmente duas cartas 2, até 21.

Portanto, no modelo de AR proposto para o *Blackjack*, foram considerados 18 possíveis estados, baseados na soma das cartas do agente, e 2 ações realizáveis, que são comprar ou parar.

3.2 Recompensas

De acordo com o estado em que o agente se encontra e a ação executada são recebidas recompensas baseadas no estado futuro do agente, ou seja, para uma possível situação de jogo, o jogador recebe reforços (positivos ou negativos) dependendo da ação tomada, com base no resultado dessa ação. Por exemplo, caso o agente opte por comprar mais uma carta e a soma de sua mão não tenha ultrapassado 21, ele recebe um reforço positivo, indicando que, para aquele estado em que ele se encontrava, pode ser interessante optar por comprar mais uma carta. Da mesma forma, caso a soma tenha ultrapassado o limite, a recompensa é negativa, indicando o oposto.

Considerando a variável *sum* como a soma das cartas do agente, a definição das recompensas (R) foi feita com base em quatro situações de jogo, explicadas a seguir:

- $R = 1000$: caso a compra de uma carta tenha resultado em $sum \leq 21$;
- $R = 1000$: caso o agente ganhe ou a rodada termine empatada, indicando que a escolha de parar no último estado pode ter sido acertada;
- $R = -1000 \times |21 - sum|$: caso a compra de uma carta tenha resultado em $sum > 21$;
- $R = -1000 \times |21 - sum|$: caso o agente perca a rodada, indicando que a escolha de parar no último estado pode não ter sido acertada.

Como o empate é considerado boa jogada, a recompensa desse resultado é a mesma de quando o agente ganha. Nas penalidades, a utilização do fator $|21 - sum|$ foi feita com o intuito de penalizar quando o agente optasse por parar caso a soma de suas cartas fosse pequena ou caso uma compra tivesse acarretado uma grande extrapolação do limite. Por exemplo, caso o agente tenha optado por parar com $sum = 15$, o valor da recompensa será $R = -6000$, e essa penalidade seria aumentada caso $sum < 15$ e o agente tivesse optado por parar ou a opção por comprar mais uma carta resultasse em $sum > 27$.

3.3 Algoritmo RL-Blackjack

O funcionamento do Algoritmo proposto (RL-Blackjack) dá-se conforme representado através do esquema da Figura 1. A partir do estado s em que o agente se encontra, definido pela soma das cartas de sua mão, uma ação a é tomada conforme as características inerentes ao AR. Caso seja tomada a decisão de comprar mais uma carta, é verificado se a soma da mão do agente não ultrapassou 21. Se sim, o agente perde, a matriz de aprendizado $Q(s,a)$ é atualizada com recompensa negativa. Caso contrário, a recompensa é positiva e o estado s é alterado. Esse processo ocorre até que a soma das cartas ultrapasse 21 ou o agente decida parar.

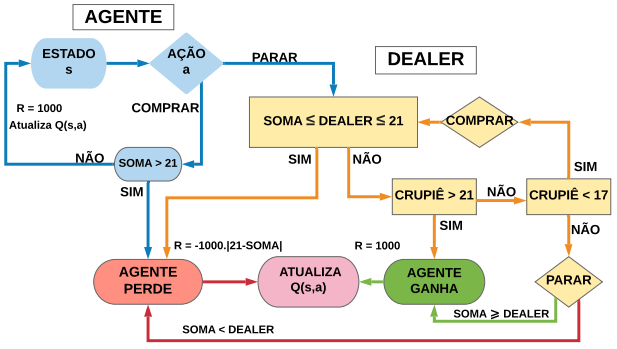

Figura 1. Dinâmica do algoritmo RL-Blackjack.

Quando o agente opta por parar, é a vez do *dealer* fazer sua jogada. Agora, a qualquer momento, caso a soma de suas cartas seja superior à soma das cartas do agente e inferior a 21, o agente perde a rodada e a matriz de aprendizado $Q(s,a)$ é atualizada com recompensa negativa. Se a soma das cartas do *dealer* superar 21, o agente ganha e é recompensado de forma positiva. Caso nenhuma das condições acima esteja acontecendo, o *dealer* opta por comprar mais uma carta enquanto a soma de sua mão não ultrapassar 16; ou parar, caso a soma de suas cartas esteja entre 17 e 21.

Quando o *dealer* para, o algoritmo RL-Blackjack verifica qual das duas mãos atingiu o valor mais próximo a 21. Caso

o valor das mãos seja o mesmo, a rodada termina em empate. Ao final da rodada, a matriz de aprendizado $Q(s,a)$ é atualizada conforme o resultado obtido. Lembrando que em caso de empate, a matriz $Q(s,a)$ é atualizada como se o agente tivesse ganhado.

4. PLANEJAMENTO DOS EXPERIMENTOS

4.1 Descrição dos Experimentos

Os experimentos foram conduzidos de modo a possibilitar a análise da influência dos parâmetros α e γ nos resultados do AR para o *Blackjack*. Para isso, as possíveis combinações de parâmetros foram determinadas com base em (Ottoni et al., 2018). Foram definidos oito possíveis valores para α, oito para γ e ϵ foi mantido constante, totalizando 64 combinações de parâmetros, como mostrado nas Equações (2) a (4).

$$\alpha = \{0{,}01; 0{,}15; 0{,}30; 0{,}45; 0{,}60; 0{,}75; 0{,}90; 0{,}99\} \quad (2)$$

$$\gamma = \{0{,}01; 0{,}15; 0{,}30; 0{,}45; 0{,}60; 0{,}75; 0{,}90; 0{,}99\} \quad (3)$$

$$\epsilon = 0{,}01 \quad (4)$$

Além disso, cada combinação foi simulada em 10 épocas com 50 episódios cada. Ou seja, para cada combinação, foram jogadas 500 rodadas, com a matriz de aprendizado $Q(s,a)$ sendo zerada a cada 50 partidas. Isso implica que, para uma dada combinação de parâmetros, o agente aprendia o jogo através do AR nas primeiras partidas. Em seguida, os resultados eram salvos e todo o aprendizado era apagado para dar início à segunda época. Dessa forma, ao final da avaliação de desempenho de uma dada combinação de α e γ, eram armazenados os resultados obtidos nas 10 épocas, cada uma com 50 episódios para aprendizado.

4.2 Metodologia para Estimação dos Parâmetros

Nesta seção, são apresentadas as análises adotadas para a estimação dos parâmetros α e γ para aplicação do AR no *Blackjack*. Para isso, foram adotados a técnica de Análise de Variância (ANOVA) (Montgomery, 2017) e, em seguida, o método de Scott-Knott (Scott and Knott, 1974; Jelihovschi et al., 2014). Essa metodologia de estimação de parâmetros do AR foi proposta em (Ottoni et al., 2020). O *software* R (R Core Team, 2020) em conjunto com os pacotes *Experimental Designs* {ExpDes.pt} (Ferreira et al., 2013) e *The ScottKnott Clustering Algorithm* {ScottKnott} (Jelihovschi et al., 2014) foram adotados na análise.

A primeira etapa foi estruturar o modelo fatorial, conforme apresentado na Equação (5):

$$y_{ijk} = \mu + \zeta_i + \eta_j + (\zeta\eta)_{ij} + \xi_{ijk}, \quad (5)$$

em que, i é o índice referente aos níveis da taxa de aprendizado ($i = 1, \ldots, 8$); j é o índice referente aos níveis do fator de desconto ($j = 1, \ldots, 8$); k é o índice referente a observação y_{ijk}; μ é o efeito geral da média; ζ_i é o efeito do fator α; η_j é o efeito do fator γ; $(\zeta\eta)_{ij}$ é o efeito da interação entre os fatores; ξ_{ijk} é o componente do erro.

Em seguida, foi aplicada a técnica de ANOVA a partir do modelo fatorial (Eq. (5)). A ANOVA busca verificar se existe diferença estatística significativa entre as médias populacionais para cada fator (α e γ) e também na interação entre os mesmos, conforme hipóteses:

$$\begin{cases} H_0: & \text{as médias são iguais,} \\ H_1: & \text{pelo menos uma das médias é diferente.} \end{cases}$$

Dessa forma, adotando um nível de significância de 5%, se a hipótese inicial (H_0) é aceita ($p{-}valor > 0{,}05$), as médias dos tratamentos para o fator são iguais. Caso contrário, se H_0 é rejeitada e a hipótese alternativa (H_1) é aceita ($p - valor < 0{,}05$), pelo menos uma média é diferente das demais.

Na sequência da metodologia experimental, caso o teste de ANOVA confirme que existe diferença significativa entre os tratamentos, é aplicado o método de Scott-Knott (SK). O SK é um algoritmo de agrupamento de dados hierárquico (Jelihovschi et al., 2014). Esse método é usado para particionar tratamentos em grupos distintos quanto o existe diferença significativa pela ANOVA. Nesse sentido, o SK divide os tratamentos em grupos de modo a maximizar a soma de quadrados (B_0). Por exemplo, inicialmente são definidos dois grupos com k_1 e k_2 tratamentos em cada um. Em seguida, são calculadas as somas totais de ambos os grupos (T_1 e T_2) e a soma de quadrados (B_0), conforme Equações de (6) a (8):

$$T_1 = \sum_{i=1}^{k_1} y_{(i)}, \quad (6)$$

$$T_2 = \sum_{i=k_1+1}^{k_1+k_2} y_{(i)}, \quad (7)$$

$$B_0 = \frac{T_1^2}{k_1} + \frac{T_2^2}{k_2} - \frac{(T_1+T_2)^2}{k_1+k_2}, \quad (8)$$

em que, y_i é a média do tratamento i; e $k = (k_1 + k_2)$ é o total de tratamentos. Na sequência, são testadas as hipóteses de grupos homogêneos (H_{sk0}) contra grupos heterogêneos (H_{sk1}), adotando as estatísticas de máxima verossimilhança e qui-quadrado (Jelihovschi et al., 2014; Ottoni et al., 2020). Se a hipótese H_{sk0} é rejeitada e H_{sk1} é aceita, então os grupos são separados. O procedimento de particionamento continua (cálculos das Eqs. 6 a 8 e testes de hipóteses) para os novos grupos formados até H_{sk0} ser aceita, indicando não ser mais necessário a divisão dos tratamentos.

5. RESULTADOS

5.1 Análise Descritiva

A princípio, observando os resultados obtidos por época, apenas 18 das 64 combinações de parâmetros apresentaram o número médio de vitórias ou empates maior ou igual que a média de derrotas (Média Vit/Emp $\geq$ 25). A Tabela 1

apresenta os resultados obtidos para essas 18 combinações diante da média de resultados das 10 épocas (50 episódios por época).

Tabela 1. Média de resultados das 18 combinações com mais vitórias (Vit) e empates (Emp).

α	γ	Média Vit/Emp
0,90	0,15	27,8
0,30	0,15	27,7
0,30	0,30	26,7
0,75	0,45	26,6
0,90	0,45	26,6
0,99	0,45	26,3
0,99	0,60	26,2
0,30	0,60	26,1
0,99	0,15	26,0
0,30	0,45	25,8
0,15	0,75	25,6
0,60	0,01	25,6
0,45	0,30	25,3
0,60	0,99	25,3
0,99	0,75	25,2
0,30	0,90	25,1
0,60	0,90	25,0
0,75	0,90	25,0

Apesar da Tabela 1 ser uma ferramenta útil na análise do desempenho dos parâmetros no jogo de *Blackjack*, não existe garantia que uma combinação é estatisticamente diferente das demais. A partir disso, faz-se necessário o uso de estratégias de estimação de parâmetros mais avançadas, como adoção da técnica de Análise de Variância (ANOVA) seguida do método de Scott-Knott, apresentados na Seção 4.3, cujos resultados são exibidos na próxima seção.

5.2 Estimação de Parâmetros

Os resultados da ANOVA indicaram para aceitar H_O para efeito de interação entre os fatores ($p - valor = 0{,}433$). No entanto, rejeitar H_O e aceitar H_1 para os efeitos principais dos dois parâmetros: α ($p - valor = 0{,}008$) e γ ($p - valor = 0{,}017$). Também vale destacar que as medidas de adequação para o modelo ANOVA foram observadas e satisfeitas: normalidade dos resíduos, homogeneidade das variâncias e independência (Montgomery, 2017).

Em seguida, após o teste de ANOVA confirmar que existe diferença significativa entre os tratamentos nos resultados do *Blackjack* ao adotar parâmetros distintos de α e γ, o método de Scott-Knott (SK) foi adotado. O SK foi utilizado para dividir em grupos os níveis dos parâmetros (α e γ). Conforme Tabelas 2 e 3, para cada um dos fatores, o método SK definiu dois grupos: A e B para α e, C e D para γ.

Os grupos A e C para os parâmetros α e γ, respectivamente, alcançaram os maiores resultados médios nas partidas do *Blackjack*.

Finalmente, para definir um único valor para cada parâmetro, foi adotado o critério de desempate de maior média nos grupos A (α) e C (γ). Dessa forma, os parâmetros estimados foram: $\alpha = 0{,}30$ e $\gamma = 0{,}15$.

Tabela 2. Resultados de agrupamentos pelo método de Scott-Knott para o parâmetro α.

α	**Médias**	**Grupos**
0,30	25,5125	A
0,99	24,8750	A
0,90	24,6875	A
0,60	24,4875	A
0,75	24,1500	B
0,45	23,6500	B
0,15	23,6250	B
0,01	23,5125	B

Tabela 3. Resultados de agrupamentos pelo método de Scott-Knott para o parâmetro γ.

γ	**Médias**	**Grupos**
0,15	25,2125	C
0,45	25,1625	C
0,60	24,4250	D
0,30	24,3750	D
0,01	24,2875	D
0,90	23,9500	D
0,75	23,9250	D
0,99	23,1625	D

5.3 Comparação com outros Trabalhos

Com o intuito de comprovar a eficácia dos parâmetros estimados $\alpha = 0{,}30$ e $\gamma = 0{,}15$, foram reproduzidas mais 10 épocas de 100 episódios cada. O resultado desses jogos foi comparado com desempenhos oriundos de outras combinações de parâmetros, estabelecidas na literatura, tanto no âmbito do *Blackjack* (Pérez-Uribe and Sanchez, 1998; Kakvi, 2009; Gan et al., 2019), quanto em outros tipos de aplicações (Celiberto Jr et al., 2012; Ottoni et al., 2018). A Tabela 4 mostra os valores dos parâmetros usados para comparação, assim como a aplicação para a qual foram estimados.

Tabela 4. Parâmetros de outros trabalhos adotados na comparação dos resultados.

Trabalho	Domínio	α	γ
Uribe1998	*Blackjack*	0,01	0,90
Kakvi2009	*Blackjack*	0,60	0,75
Gan2019	*Blackjack*	0,5475	0,9792
CelibertoJr2012	Futebol de Robôs	0,125	0,90
Ottoni2018	Caixeiro Viajante	0,7273	0,1539

O resultado obtido diante das 100 rodadas jogadas a cada época por cada combinação de parâmetros é mostrado na Tabela 5, onde pode ser visto que o agente tem o melhor desempenho quando os parâmetros utilizados são $\alpha = 0{,}30$ e $\gamma = 0{,}15$, vencendo ou empatando uma média de 52,7 rodadas a cada época. Para os demais parâmetros, a ordem de classificação de desempenho de média de vitórias ou empates por combinação ficou: 2º - Ottoni2018 (51,2), 2º - Kavi2009 (50,6), 3º - Gan2019 (49,7), 4º - Uribe1998 (48,7) e 5º - CelibertoJr2012 (48,6).

Além disso, vale destacar que a combinação estimada ($\alpha = 0{,}30$ e $\gamma = 0{,}15$), foi a única que ultrapassou a marca de 60 vitórias ou empates em uma época (Época 4 - Tabela 5), entre os parâmetros analisados. Nesse sentindo, reforçando a relevância da adoção da metodologia para a estimação de α e γ na aplicação do AR no *Blackjack*.

Tabela 5. Comparação dos resultados das combinações de parâmetros a cada época.

α	γ	Época	Vit/Emp	Média por Época
0,30	0,15	1	51	52,7
		2	46	
		3	51	
		4	61	
		5	54	
		6	47	
		7	58	
		8	54	
		9	57	
		10	48	
0,7273	0,1539	1	54	51,2
		2	47	
		3	50	
		4	47	
		5	57	
		6	53	
		7	48	
		8	47	
		9	58	
		10	51	
0,60	0,75	1	41	50,6
		2	54	
		3	52	
		4	56	
		5	46	
		6	51	
		7	52	
		8	53	
		9	48	
		10	53	
0,5475	0,9792	1	41	49,7
		2	50	
		3	51	
		4	49	
		5	57	
		6	48	
		7	57	
		8	44	
		9	47	
		10	53	
0,01	0,90	1	46	48,7
		2	43	
		3	54	
		4	47	
		5	51	
		6	46	
		7	50	
		8	49	
		9	58	
		10	43	
0,125	0,90	1	44	48,6
		2	51	
		3	47	
		4	47	
		5	51	
		6	44	
		7	49	
		8	45	
		9	57	
		10	51	

6. CONCLUSÃO

O objetivo deste trabalho foi aplicar o AR no domínio do jogo de cartas *Blackjack* com o intuito de estimar os parâmetros de taxa de aprendizado e fator de desconto γ, de modo a maximizar o desempenho do algoritmo no jogo de cartas. Para isso, foi desenvolvido um sistema de AR estruturado em ações, estados, recompensas dentro do jogo e proposto o algoritmo RL-Blackjack.

Adotando a técnica de ANOVA e o método Scott-Knott, os resultados da metodologia de estimação de parâmetros apontaram para a recomendação de $\alpha = 0{,}30$ e $\gamma = 0{,}15$ para o jogo de *Blackjack*. Para comprovação da eficácia da estimação, a combinação de α e γ sintonizada foi comparada com parâmetros adotados em outros trabalhos da literatura. Nesse sentido, os parâmetros ajustados obtiveram o melhor desempenho. Vale destacar também que a combinação estimada foi a única (entre os parâmetros analisados) a alcançar a marca de 61 vitórias ou empates em uma época com 100 jogos.

Em trabalhos futuros, espera-se adotar outros algoritmos de AR para a o domínio do *Blackjack*, como por exemplo o SARSA (Sutton and Barto, 2018). Além disso, também é esperado analisar a influência de outros parâmetros no desempenho do AR no *Blackjack*, como a estrutura da função de reforço e a definição do parâmetro ϵ da política $\epsilon - greedy$.

AGRADECIMENTOS

Os autores agradecem o apoio para realização deste trabalho do CNPq, UFJF, UFRB, UFSJ (Edital 001/2019/Reitoria) e Coordenação de Aperfeiçoamento Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

Barsce, J.C., Palombarini, J.A., and Martínez, E.C. (2017). Towards autonomous reinforcement learning: Automatic setting of hyper-parameters using bayesian optimization. In *2017 XLIII Latin American Computer Conference (CLEI)*, 1–9.

Celiberto Jr, L.A., Matsuura, J.P., De Màntaras, R.L., and Bianchi, R.A. (2012). Reinforcement learning with case-based heuristics for robocup soccer keepaway. In *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, 7–13. IEEE.

Da Silva, F. and Reali Costa, A. (2019). A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64, 645–703.

da Silva, L.A., Peres, S.M., and Boscarioli, C. (2017). *Introdução à mineração de dados: com aplicações em R*. Elsevier Brasil.

Even-Dar, E. and Mansour, Y. (2003). Learning Rates for Q-learning. *Journal of Machine Learning Research*, 5, 1–25.

Ferreira, E.B., Cavalcanti, P.P., Nogueira, D.A., and Ferreira, M.E.B. (2013). Package 'expdes. pt'.

Gan, X., Guo, H., and Li, Z. (2019). A new multi-agent reinforcement learning method based on evolving dynamic correlation matrix. *IEEE Access*, 7, 162127–162138.

Hutter, F., Kotthoff, L., and Vanschoren, J. (eds.) (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer. In press, available at http://automl.org/book.

Jelihovschi, E.G., Faria, J.C., and Allaman, I.B. (2014). Scottknott: a package for performing the scott-knott clustering algorithm in r. *TEMA (São Carlos)*, 15(1), 3–17.

Kakvi, S. (2009). Reinforcement learning for blackjack. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5709 LNCS, 300–301.

Kober, J., Bagnell, J.A., and Peters, J. (2013). Reinforcement Learning in Robotics: A Survey. *International Journal of Robotics Research*, July.

Liessner, R., Schmitt, J., Dietermann, A., and Bäker, B. (2019). Hyperparameter optimization for deep reinforcement learning in vehicle energy management. In *ICAART 2019 - 11th International Conference on Agents and Artificial Intelligence*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

Montgomery, D.C. (2017). *Design and analysis of experiments*. New York: John Wiley & Sons., 9th edition.

Nepomuceno, E.G. (2019). A novel method for structure selection of the recurrent random neural network using multiobjective optimisation. *Applied Soft Computing*, 76, 607–614.

Ottoni, A.L.C., Nepomuceno, E.G., and de Oliveira, M.S. (2018). A response surface model approach to parameter estimation of reinforcement learning for the travelling salesman problem. *Journal of Control, Automation and Electrical Systems*, 29(3), 350–359.

Ottoni, A.L.C., Nepomuceno, E.G., de Oliveira, M.S., and de Oliveira, D.C.R. (2020). Tuning of reinforcement learning parameters applied to sop using the scott–knott method. *Soft Computing*, 24, 4441–4453.

Pérez-Uribe, A. and Sanchez, E. (1998). Blackjack as a test bed for learning strategies in neural networks. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence*, volume 3, 2022–2027. IEEE.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Ramirez, A., Reinman, S., and Norouzi, N. (2019). Pokerbot: Hand strength reinforcement learning. In *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 1–6.

Russell, S.J. and Norving, P. (2013). *Artificial Intelligence*. Campus, 3st ed.

Schweighofer, N. and Doya, K. (2003). Meta-learning in reinforcement learning. *Neural Networks*, 16(1), 5–9.

Scott, A.J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 507–512.

Silva, I.N., Spatti, D.H., and Flauzino, R.A. (2016). *Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas: fundamentos teóricos e aspectos práticos*. ArtLiber.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419), 1140–1144.

Snyder, A. (2013). *Big Book of Blackjack*. Cardoza Publishing.

Sutton, R. and Barto, A. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2nd edition.

Watkins, C.J. and Dayan, P. (1992). Technical note Q-learning. *Machine Learning*, 8(3), 279–292.