Jonas Cardoso Carvalho Santana

Aprendizado de Máquina e o Dilema dos Prisioneiros

Brasil

2019

Jonas Cardoso Carvalho Santana

Aprendizado de Máquina e o Dilema dos Prisioneiros

Universidade de Brasília – UnB

Faculdade de Economia, Administração, Contabilidade

e Gestão de Políticas Públicas

Graduação

Orientador: Daniel Oliveira Cajueiro

Brasil

2019

Resumo

Dado o grande crescimento da área de aprendizado de máquina, buscamos verificar o desempenho da aplicação de seus métodos na confecção de estratégias frente o caso do Dilema dos Prisioneiros iterado. Também buscamos verificar se há correlação entre esse e o desempenho no Processo de Moran, utilizado para análise intertemporal de populações.

Palavras-chave: dilema dos prisioneiros. aprendizado de máquina. processo de Moran

Abstract

Given the great growth of the machine learning area, we seek to verify the performance of the application of its methods in strategy making in the case of the iterated Prisoners' Dilemma. We also sought to verify if there is a correlation between this and the performance in the Moran Process, used for intertemporal analysis of populations.

Keywords: prisoner’s dilema. machine learning. Moran process

Lista de Ilustrações

Sumário

Introdução

O campo da aprendizagem de máquina surgiu da busca por programas que pudessem reconhecer padrões de forma autônoma. Seja identificar o conteúdo de uma imagem, transcrever sons de forma simultânea, prever o futuro de séries temporais, são inúmeras as aplicações de seus métodos. Podemos aplicar esse instrumental no escopo da teoria dos jogos, na qual enfrentamos o problema de decisão ótima do agente. Ao aplicar esses métodos em situações de incerteza, na qual não se sabe como seu oponente se comportará, podemos extrair conjuntos de estratégias que após serem treinadas realizando esses jogos inúmeras vezes aprendem como se comportar de maneira a obter melhor retorno.

Buscamos estudar o comportamento dessa abordagem quando defrontada com o caso especial do Dilema dos Prisioneiros. Uma situação na qual a busca pelo retorno individual ótimo gera ineficiência de Pareto. Esperamos compreender se a busca por maiores recompensas, dado o cenário de múltiplas iterações, resulta numa situação de cooperação ou não. Damos um passo a mais e analisamos se essas estratégias formadas quando colocadas em um processo de evolução populacional acabam por dominar as outras.

Inicialmente definimos o Dilema dos Prisioneiros e suas características do caso singular, iterado finito e iterado infinito. Em seguida simulamos o caso infinito com 17 estratégias diferentes, sendo 7 baseadas em aprendizado de máquina. Repetimos 50000 vezes jogos com 200 rodadas, porém emitimos dos jogadores essa informação. Verificamos os resultados com resultados já encontrados por outros autores. E, por fim, simulamos 100 vezes o processo evolucionário de uma população composta por essas estratégias e comparamos o desempenho obtido nessa com o desempenho obtido no dilema dos prisioneiros iterado infinito.

Parte I

Referencial Teórico

1 O Dilema dos Prisioneiros

Somos confrontados com problemas de decisão recorrentemente. Escolher a roupa a ser utilizada no dia seguinte, decidir qual será sua próxima refeição, alocar como gastará o seu salário, liberar ou não o comércio com outro país são alguns exemplos. Buscamos por meio da teoria dos jogos entender como essas decisões se dão. Em seguida, procuramos definir uma maneira ótima de se comportar frente a essas decisões. Porém, seria possível o ótimo individual não resultar no ótimo social?

Primeiramente definiremos as características do jogo em questão. Em seguida, analisaremos o caso iterado desse jogo e suas particularidades. Por fim, veremos os estudos já realizados quanto a esse problema, suas conclusões e qual será o ponto de partida para a análise realizada nesse trabalho.

1.1 Definição do problema de decisão

Suponha que dois criminosos foram presos em flagrante e estão sendo questionados separadamente na delegacia. A polícia suspeita que eles realizaram outro crime anteriormente, porém não possuem provas que os incriminem. Para incentivá-los a confessar esse crime, o policial dá a cada detento a opção de delatar ou não seu companheiro sujeito aos seguintes resultados que compõem o conjunto X:

- Ambos delatam: ambos cumprem pena de quatro anos
- Um delata e o outro não: o delator não cumpre pena e o delatado cumpre cinco anos
- Ambos não delatam (cooperam entre si): ambos cumprem pena de dois anos

O instrumental da teoria dos jogos (TADELIS, 2013) nos permite analisar esse problema de decisão enfrentado por ambos os criminosos. Denominamos os problemas de decisão como jogos e seus participantes como jogadores. No caso acima descrito, temos:

- Conjunto dos jogadores participando do jogo:
    - $N = \{1, 2\}$

- Conjunto de estratégias para cada jogador pertencente a N:
  - $S_i = \{\text{Delatar (D), Cooperar (C)}\}\ \forall\ i \in N$
- Relações de preferências responsáveis por alocar os resultados pertencentes a X do mais desejado ao menos desejado. Portanto, no caso do dilema dos prisioneiros, quanto menor for o tempo da pena de determinado resultado, mais preferível ele será. Logo, temos preferências completas, transitivas e, portanto, racionais.
- Seja $u_i(s_1, s_2) : X \rightarrow \mathbb{R}$ a função retorno do jogador i dado que o jogador 1 escolheu a estratégia $s_1 \in S_1$ e o jogador 2 escolheu a estratégia $s_2 \in S_2$. Essa função é tal que dado dois resultados $x_1, x_2 \in X$, o retorno de x1 somente será maior que o de x_2 se x_1 for preferível a x_2. Definimos essas funções por:

$$u_1(D, D) = u_2(D, D) = 1$$

$$u_1(D, C) = u_2(C, D) = 5$$

$$u_1(C, D) = u_2(D, C) = 0$$

$$u_1(C, C) = u_2(C, C) = 3$$

Nesse caso temos um jogo na forma normal $G = \{S_1, S_2; u_1, u_2\}$. Ou seja, ambos os jogadores jogam simultaneamente e a combinação das estratégias escolhidas gera determina o retorno de cada um. Todo jogo normal pode ser representado na forma de uma matriz denominada matriz retorno. Para o dilema dos prisioneiros descrito, temos a seguinte matriz retorno:

Figura 1- Matriz Retorno Dilema dos Prisioneiros

		Prisioneiro 2	
		Não Delatar	Delatar
Prisioneiro 1	Não Delatar	3, 3	0, 5
	Delatar	5, 0	1, 1

Fonte: Autor

Observamos que para ambos os prisioneiros a estratégia “Não Delatar” leva a resultados melhores que “Delatar” independente da estratégia escolhida pelo outro. Definimos essa estratégia como estritamente dominada. Portanto, um jogador com preferências racionais nunca escolherá uma estratégia estritamente dominada, pois sempre haverá outra estratégia que resulta em um retorno maior.

Como ambos jogadores são racionais, ambos optarão por “Delatar” e o resultado será o retorno (1, 1). Observamos que esse resultado é estritamente pior que se ambos não delatassem. Temos nesse caso um resultado Pareto ineficiente, há como melhorar o retorno de um jogador sem piorar o do outro. Podemos também analisar ambos os jogadores como um grupo com preferências racionais e cujo ótimo seria ambos delatarem. Temos que a racionalidade individual não necessariamente leva à racionalidade do conjunto.

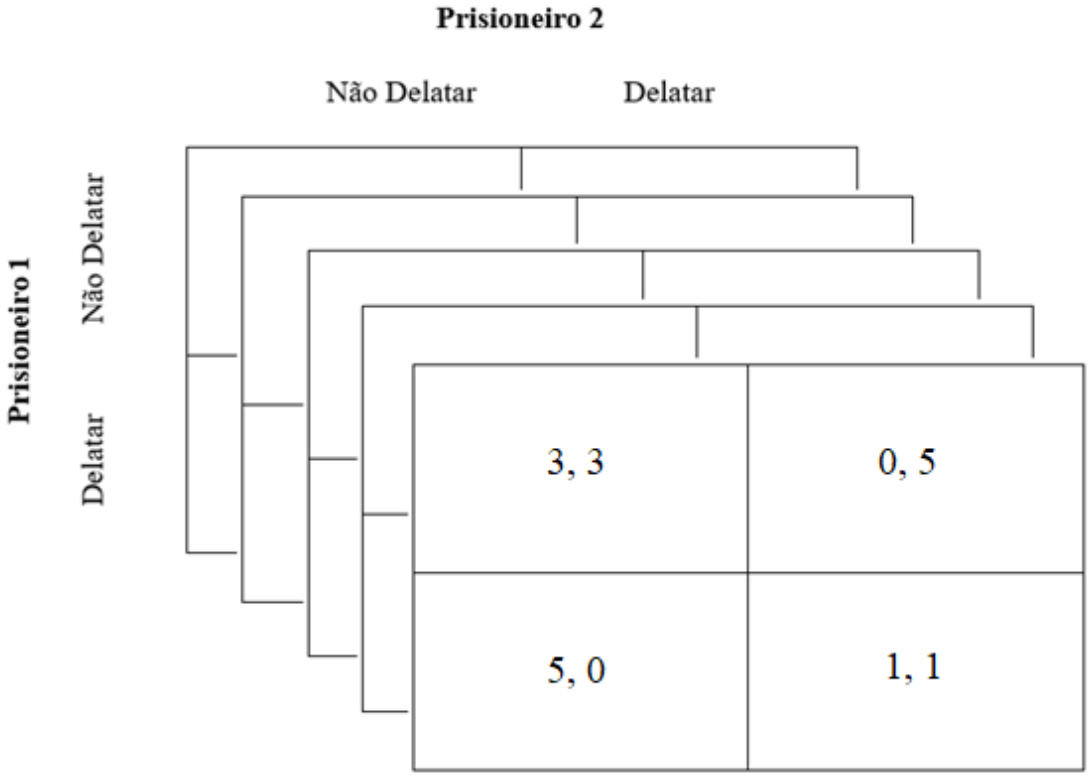
1.2 O Dilema dos Prisioneiros Iterado

Agora analisaremos esse mesmo jogo, porém repetido em sequência um número finito ou não de vezes. Veremos como a memória das estratégias previamente selecionadas influencia a escolha da estratégia atual. No primeiro caso, podemos observar por indução reversa que ambos os jogadores delatarão em todas as rodadas. Isso se deve à na última rodada não haver incentivos para cooperação, dado que ela não deixará informação para rodadas futuras. Portanto, o jogador escolherá o ótimo do caso singular. Dado que ambos os jogadores sabem que o outro não cooperará na última rodada, logo ambos não cooperarão na penúltima e assim sucessivamente até a primeira rodada.

Entretanto, uma observação comum em experimentos do dilema dos prisioneiros iterado finito é que os jogadores nem sempre escolhem o ótimo do caso singular, existe um grau de cooperação. Uma possível explicação para tal seria que os jogadores possuem informação incompleta sobre os outros jogadores (KREPS, MILGROM, *et al.*, 1982). O jogador conferiria probabilidade α de o oponente não ser racional e 1-α de ele o ser, levando-o a considerar seu retorno esperado para o jogo. Nesse caso, o equilíbrio toma a forma de um equilíbrio sequencial. Esse requer que a estratégia escolhida pelo jogador em qualquer ponto do jogo seja parte de uma estratégia ótima daquele ponto em diante

dadas as hipóteses do jogador de como o jogo se desenrolará devido à atualização da probabilidade de racionalidade atribuída ao outro jogador.

Figura 2 - Matriz Retorno Dilema dos Prisioneiros Iterado



Fonte: Autor

No caso do dilema dos prisioneiros iterado infinitamente podemos considerar dois cenários, com e sem desconto intertemporal. Em ambos os casos os jogadores buscam maximizar seu retorno esperado total, entretanto no segundo caso é necessário trazer os retornos a valor presente. Suponha o jogo descrito inicialmente iterado infinitamente, no qual o jogador 1 possui taxa de desconto intertemporal ε e o jogador 2 utilizará a estratégia *Grim Reaper* ou *Grudger*. Ou seja, ele irá cooperar sempre a não ser que o jogador 1 não coopere. A partir desse momento, o jogador 2 nunca mais cooperará. Para definir a estratégia ótima do jogador 1, compararemos a decisão de cooperar sempre com a de não cooperar em dado momento:

- Retorno esperado de cooperar sempre:
  - $\sum_{i=1}^{\infty} 3 * \varepsilon^{i-1} = \frac{3}{1-\varepsilon}$

- Retorno esperado de não cooperar sempre:
  - $5 + \sum_{j=2}^{\infty} 1 * \varepsilon^{j-1} = 5 + \frac{1*\varepsilon}{1-\varepsilon}$

Comparando os dois resultados, observamos que o jogador 1 não cooperará se ε > 50%, cooperará se ε < 50% e se ε = 50% ele ficará indiferente entre as duas opções. Temos que a depender dos parâmetros a cooperação se torna a estratégia dominante.

1.3 O Experimento de Axelrod

Em 1980, o professor da Universidade de Michigan Robert Axelrod realizou dois experimentos nos quais pediu a diversos pesquisadores que lhe enviassem programas em FORTRAN e Basic contendo suas estratégias para que competissem entre si no jogo do dilema dos prisioneiros iterado. O primeiro (AXELROD, 1980a) contou com 14 estratégias, todas jogaram entre si, cada jogo teve 200 rodadas e, portanto, configurou-se o caso finito pois cada jogador tinha conhecimento quanto ao término. O segundo (AXELROD, 1980b) contou com 63 estratégias, todas jogaram entre si 5 vezes e o tamanho de cada partida foi definido aleatoriamente. Definiu-se que o jogo poderia terminar após cada rodada com probabilidade 0.00346, essa probabilidade leva o valor esperado do tamanho a ser 200. Da distribuição composta por esses valores, retirou-se de forma aleatória 5, os quais seriam os respectivos tamanhos de cada partida para todos os confrontos.

Em ambos os experimentos a estratégia vencedora foi a “Tit-for-Tat”, a qual consiste em começar cooperando e em seguida sempre jogar a mesma estratégia que o outro jogador escolheu na rodada anterior. Nesse trabalho, observaremos o confronto entre estratégias pré-definidas e aquelas resultantes de processos de aprendizagem de máquina. Para a generalização do jogo estudado, nos referiremos aos prisioneiros 1 e 2 como jogador e oponente.

2. Aprendizagem de Máquina

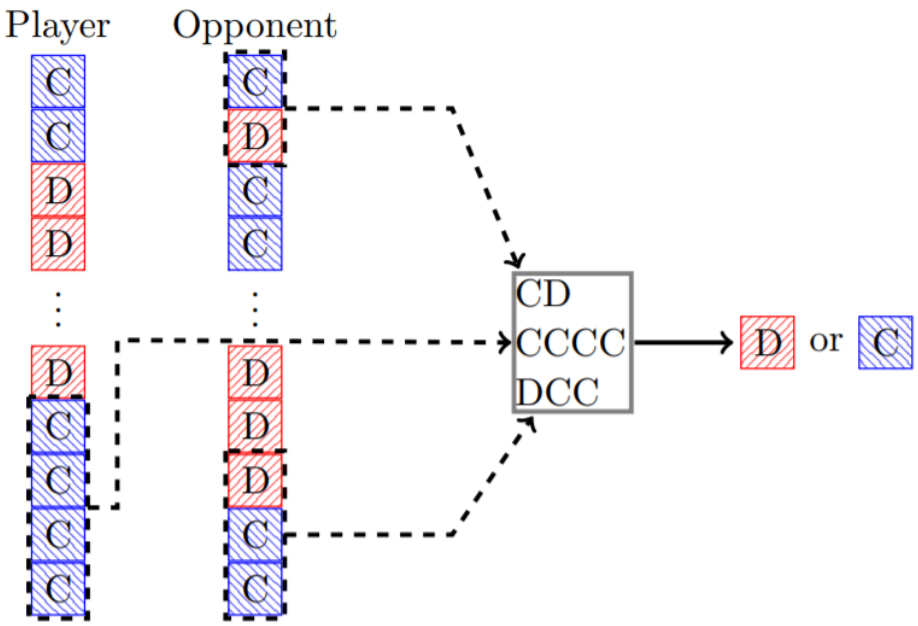
Podemos utilizar diferentes abordagens de aprendizagem de máquina para criar estratégias que joguem o dilema dos prisioneiros iterado. De maneira geral, em um processo de treinamento, o jogador a cada rodada receberá o retorno quanto à qualidade da jogada escolhida e, com base nele, atualizará seu processo decisório para as próximas jogadas. Após múltiplas rodadas, espera-se que o jogador tenha aprendido uma maneira eficiente de jogar. Tem-se que essas estratégias além de terem melhor desempenho que as estabelecidas em ambos os artigos de Axelrod (HARPER, KNIGHT, *et al.*, 2017) podem inclusive resultar em combinações de estratégias pré-definidas. Nesse trabalho verificaremos se essa melhor performance resulta em maior dominância populacional por meio de um processo de Moran (MORAN, 1958).

As estratégias apresentadas foram treinadas utilizando-se dois métodos: algoritmo evolutivo e otimização por enxame de partículas (KENNEDY e EBERHART, 1995). O primeiro consiste em gerar uma população aleatória, em seguida avalia-se a aptidão de cada indivíduo e seleciona-se os melhores. Esses irão se reproduzir misturando suas características com chance de sofrerem mutações e repete-se o processo até que reste um tipo dominante. O segundo inicializa uma população aleatória em que cada indivíduo possui um estado e uma velocidade inicial. Similarmente à otimização simplex, calcula os erros referentes a cada indivíduo, entretanto, possui memória e leva em conta o resultado total do grupo para executar sua ação. Sua próxima ação é definida pela velocidade que o leva com pesos diferentes em direção à melhor resposta por ele obtida, à melhor resposta obtida em toda população e no mesmo sentido em que já estava se movendo.

2.1 *LookerUp*

Essa estratégia mapeia por meio de uma tabela (Lookup Table) o que aconteceu no passado de forma a determinar respostas ótimas. Essa é criada utilizando os parâmetros que definem o alcance de sua memória. Eles são as n_1 primeiras e m_1 últimas jogadas do oponente e as suas m_2 últimas jogadas. Por exemplo, caso esses parâmetros sejam $n_1 = m_1 = 0$ e $m_2 = 1$, podemos descrever a estratégia determinística conhecida como "Tit-For-Tat" na qual o jogador sempre joga igual à última jogada de seu oponente.

Figura 3 - Representação Diagramática do Arquetipo LookerUp

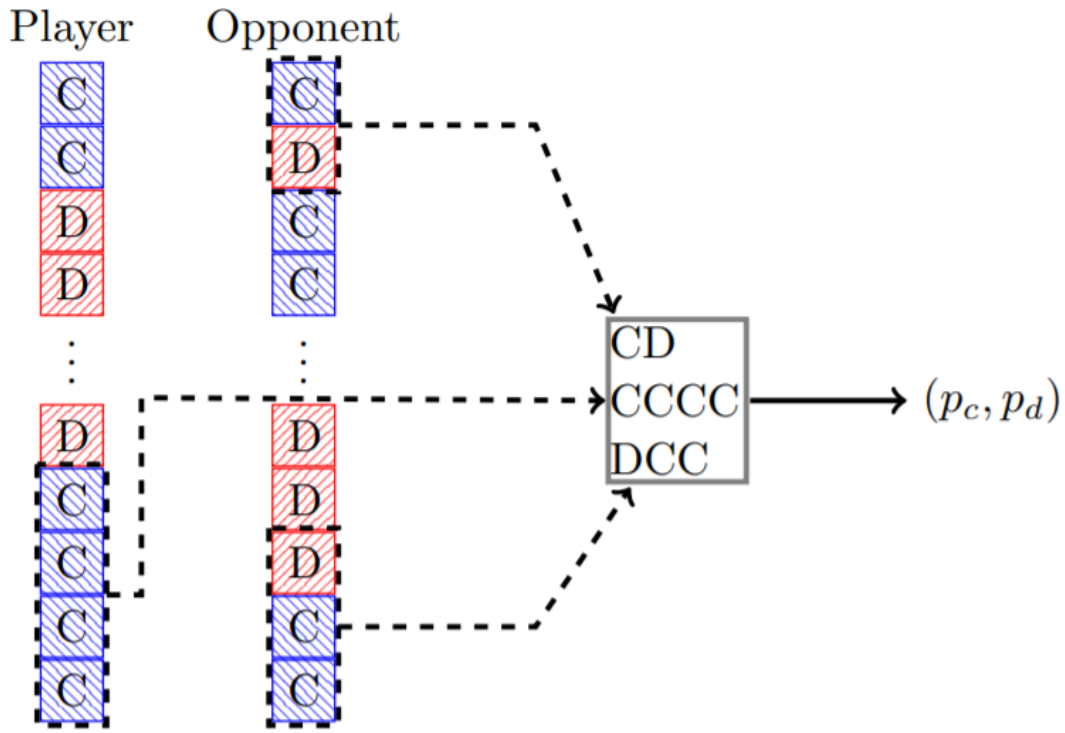


Fonte: (HARPER, KNIGHT, *et al.*, 2017)

2.2 *Gambler*

Essa estratégia se comporta igual à *LookerUp*, porém a tabela guarda não a resposta ótima, mas a relação ótima das probabilidades de cada ação. Entretanto o comportamento resultante do treinamento dessa estratégia muitas vezes confere recorrentemente probabilidades 0 ou 1 às ações gerando um misto de estratégia determinística e estocástica.

Figura 4 - Representação Diagramática do Arquetipo *Gambler*



Fonte: (HARPER, KNIGHT, *et al.*, 2017)

2.3 Rede Neural Artificial (RNA)

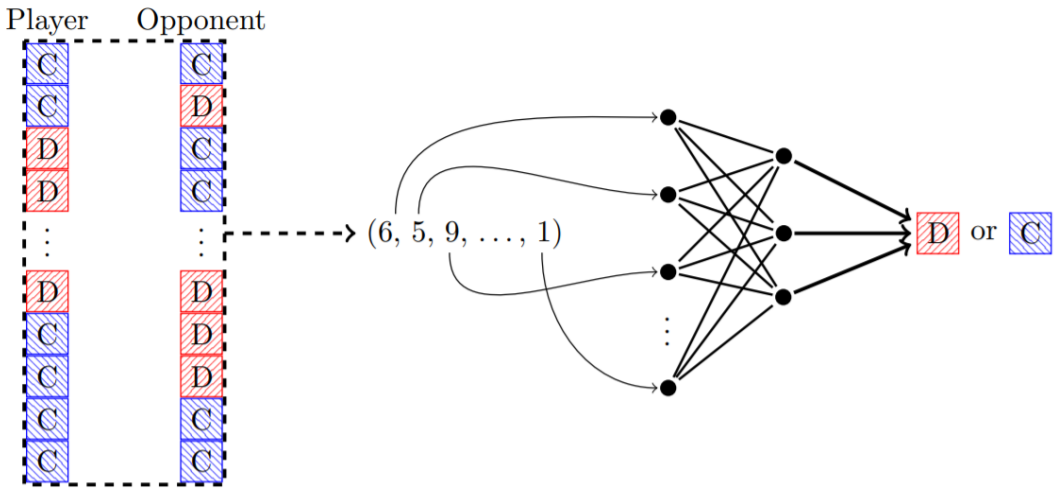
Partindo do conceito de rede neural artificial *feedforward* descrito no livro *Deep Learning* (GOODFELLOW, BENGIO e COURVILLE, 2016), temos uma estratégia que utiliza as seguintes características:

- Primeira escolha do oponente foi C
- Primeira escolha do oponente foi NC
- Segunda escolha do oponente foi C
- Segunda escolha do oponente foi NC
- Última escolha do jogador foi C
- Última escolha do jogador foi NC
- Penúltima escolha do jogador foi C
- Penúltima escolha do jogador foi NC
- Última jogada do oponente foi C
- Última jogada do oponente foi NC
- Penúltima escolha do oponente foi C

- Penúltima escolha do oponente foi NC
- Total de cooperações do oponente
- Total de não cooperações do oponente
- Total de cooperações do jogador
- Total de não cooperações do jogador
- Número da rodada

Esses parâmetros servem de entrada para uma rede neural feedforward (sem retroalimentação) com uma camada oculta. Treinar essa estratégia consiste em encontrar os parâmetros ótimos das funções calculadas pelos neurônios individuais.

Figura 5 - Representação Diagramática do Arquetipo RNA

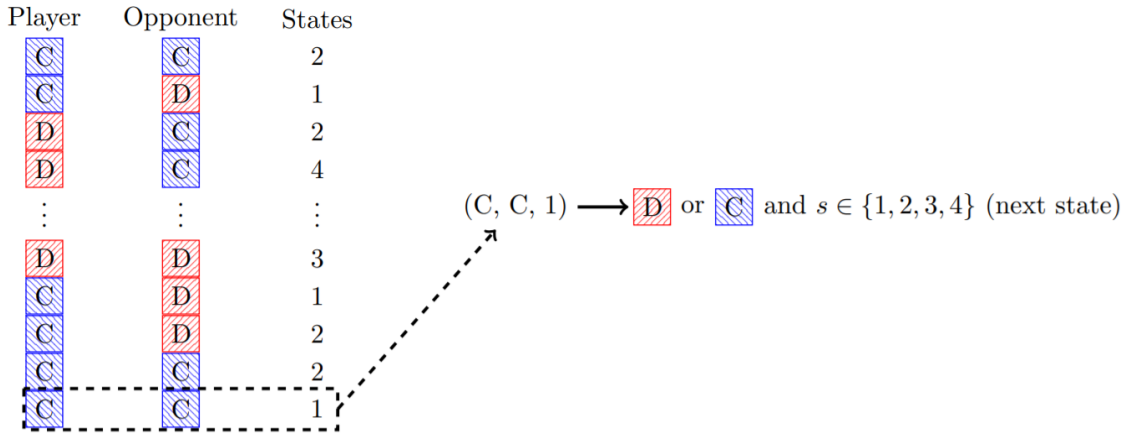


Fonte: (HARPER, KNIGHT, *et al.*, 2017)

2.4 Máquina de estados finita (FSM)

Inicialmente esse método numera todos os possíveis estados do jogo. Em seguida, para definir a estratégia a ser jogada, identifica em qual estado ela se encontra e escolhe a jogada ideal para o estado que ocorre na sequência.

Figura 6 - Representação Diagramática do Arquetipo Máquina de Estados Finita

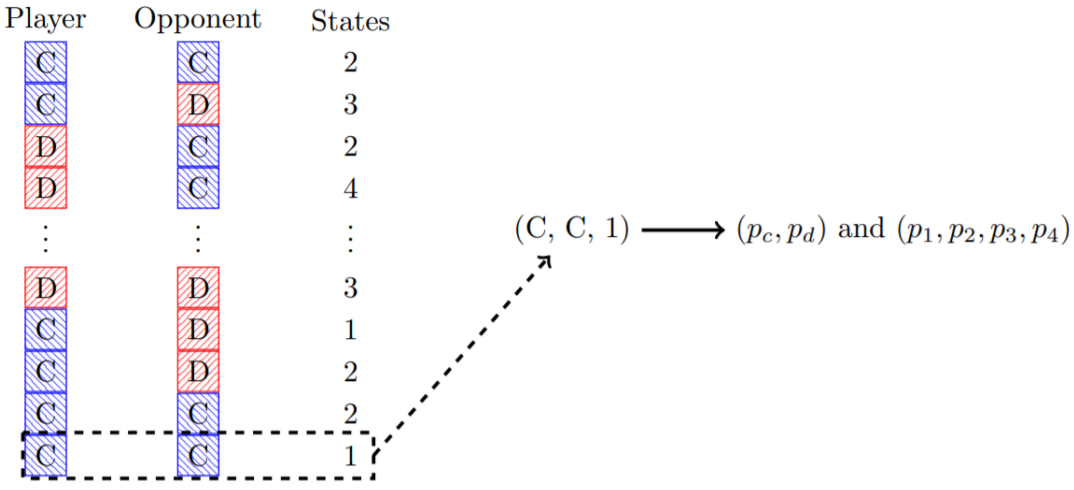


Fonte: (HARPER, KNIGHT, *et al.*, 2017)

2.5 Modelo Oculto de Markov (HMM)

Esse modelo, analogamente à relação entre Gambler e LookerUp, confere comportamento estocástico à estratégia máquina de estados finita. Identifica em qual estado o jogador se encontra e com base nas probabilidades de ocorrer cada estágio em sequência, atribui probabilidades às ações C e NC.

Figura 7 - Representação Diagramática do Arquetipo Modelo Oculto de Markov



Fonte: (HARPER, KNIGHT, *et al.*, 2017)

Consistem na combinação de diferentes estratégias. Um exemplo é o da votação majoritária. Com base em uma população de estratégias, analisa-se a resposta desejada

de cada uma delas e joga-se a mais escolhida. Podem ser compostas tanto por estratégias pré-determinadas quanto mutáveis.

2.6 Q-Learning

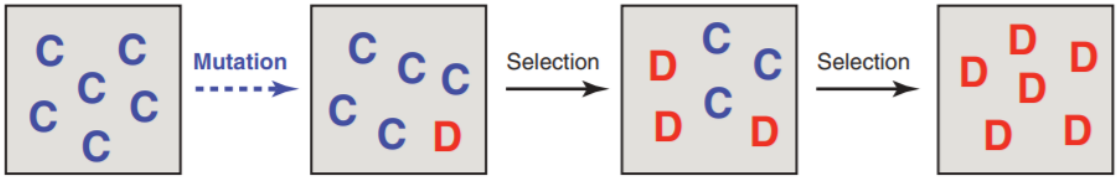
Busca maximizar o retorno esperado em cada escolha equilibrando a probabilidade de utilizar experiencias passadas com a de explorar novas experiências. Calcula esse equilíbrio com base nas taxas de aprendizagem (representa a importância de eventos recentes), desconto intertemporal (descreve a relação de preferência entre retornos presentes e futuros). Sejam a_t a ação executada em t, s_t o estado resultante da escolha a_{t-1} dado o estado s_{t-1}, α a taxa de aprendizagem, r_t o retorno gerado pela ação a_t e β a taxa de desconto. Esse processo estima a seguinte função:

$$Q^{novo}(s_t, a_t) = (1 - \alpha) * Q(s_t, a_t) + \alpha * (r_t + \beta * \max_a Q(s_{t+1}, a))$$

3. Processo de Moran

Moran (1957), estabelece um modelo para como se dá a reprodução de determinada característica dentro de uma população. Cada indivíduo de determinada população possui uma chance de se reproduzir e uma chance de morrer. Podemos utilizar esse processo para analisar uma população em que cada indivíduo representa uma estratégia diferente do dilema dos prisioneiros. E que, com base no seu desempenho, ele terá maior chance de sobreviver ou de morrer.

Figura 8 - Processo de Moran



Fonte: Nowak (2006)

Também podemos observar o estado geral da população quanto ao bem-estar antes e depois do processo de seleção. No caso acima, por exemplo, a população passa para um estado de menor bem-estar social. Quando no início todas as interações resultavam em um retorno de 3, no fim todas as interações resultam em um retorno de 1. Observamos que certas condições são necessárias para que seja possível que a cooperação se preserve no equilíbrio (NOWAK, 2006). Elas são:

1. Grau de afinidade entre os jogadores: Quanto mais próximos eles forem, maior a chance de serem altruístas e buscarem a cooperação.
2. Reciprocidade Direta: Cooperar com o intuito de influenciar o oponente a cooperar no futuro.
3. Reciprocidade Indireta: Cooperar com o intuito de elevar sua reputação social para que isso lhe gere rendimentos no futuro.
4. Reciprocidade em Rede: Dado que as interações sociais não se dão de maneira uniforme na população, ou seja, os indivíduos interagem com aqueles que estão mais próximos. É possível a criação de grupos de cooperação sustentáveis, apesar da distribuição geral da população. E a

disposição dos agentes pode afetar o resultado final (LIEBERMAN, HAUERT e NOWAK, 2005).

5. Seleção de Grupo: Interações não ocorrem somente entre indivíduos, também ocorrem entre grupos. Grupos compostos por cooperadores possuem bem-estar social maior que grupos compostos por delatores. Portanto, sua taxa de reprodução é maior, levando ao aumento de sua população no longo prazo.

Parte 2

Desempenho e Reprodução

4. O torneio

Inicialmente, replicamos o torneio realizado por Harper (2017), entretanto com menos estratégias utilizando o pacote Axelrod para Python (KNIGHT, CAMPBELL, *et al.*, 2019). Selecionamos as estratégias que performaram melhor de cada tipo de aprendizado de máquina especificado na seção anterior e, além dessas, algumas das estratégias mais simples. Foram computados 50000 torneios em que todas as estratégias competem, inclusive contra si mesmas em jogos de 200 turnos, porém nenhum jogador possui essa informação. Obtemos a pontuação média obtida por cada uma e o número de vitórias médio de cada uma (quando sua pontuação obtida excede a do oponente) e por fim analisamos se essas características influenciam no processo de Moran. Segue o código utilizado:

```
import axelrod as axl

# Definimos as estratégias que participarão do torneio
players = (axl.EvolvedLookerUp2_2_2(),
           axl.EvolvedHMM5(),
           axl.EvolvedFSM16(),
           axl.PSOGambler2_2_2(),
           axl.EvolvedANN5(),
           axl.FoolMeOnce(),
           axl.OmegaTFT(),
           axl.Gradual(),
           axl.MetaHunter(),
           axl.Cooperator(),
           axl.Defector(),
           axl.Alternator(),
           axl.TitForTat(),
           axl.Random(),
           axl.Grudger(),
           axl.RiskyQLearner(),
           axl.WinStayLoseShift())
```

```
    # Inicializamos os parâmetros do torneio e em seguida salvamos seus resultados
    tournament = axl.Tournament(players, turns = 200, repetitions = 50000,
                                match_attributes={"length": float('inf')})


    results = tournament.play()



    # Criamos os gráficos referentes às figuras 9, 10 e 11
    plot = axl.Plot(results)
    plot.boxplot()
    plot.winplot()
    plot.payoff()
```

4.1 Os jogadores

- *EvolvedLookerUp2_2_2*
    - *LookerUp* de parâmetros $n_1 = m_1 = m_2 = 2$ previamente treinada por algoritmo evolutivo.
- *EvolvedHMM5*
    - HMM previamente treinada por algoritmo evolutivo com 5 estados ocultos (prevê os 5 próximos estados).
- *EvolvedFSM16*
    - FSM previamente treinada por algoritmo evolutivo com 16 estados.
- *PSOGambler2_2_2*
    - *Gambler* de parâmetros $n_1 = m_1 = m_2 = 2$ previamente treinada por optimização por enxame de partículas.
- *EvolvedANN5*
    - Rede Neural Artificial com camada oculta de tamanho 5. Treinada por algoritmo evolutivo por meio de processos de mutação e seleção.
- *Grudger*

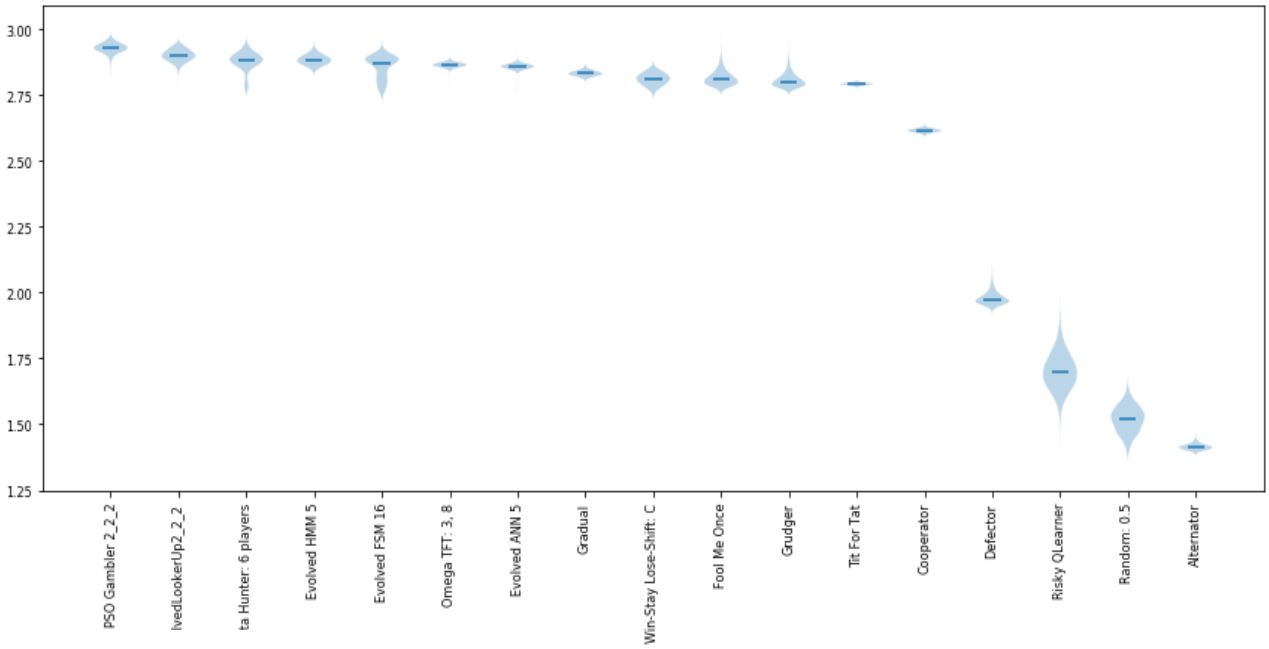
    - Coopera até que o oponente não coopere, após esse evento nunca mais coopera.
- *Fool Me Once*
    - Igual à *Grudger*, porém perdoa a primeira não cooperação.
- *TitForTat*
    - Inicia cooperando e em seguida repete a jogada anterior do oponente.
- *OmegaTFT*
    - Modifica a *TitForTat* ao verificar por ciclos viciosos desvantajosos (ambos jogadores alternando entre C e NC de forma a nunca escolherem a mesma ação simultaneamente) e tenta quebrá-los. Também é mais tolerante à não cooperação (atribui chance de ruído ou não intenção para as jogadas do oponente).
- Gradual
    - Pune a não cooperação por um número cada vez maior de rodadas. Após a punição coopera por dois turnos independente do que o oponente faça.
- *Meta Hunter*
    - Meta composta por 6 jogador especializados em identificar e vencer as estratégias *Defector*, *Alternator*, *Random*, *MathConstant*, *Cycle* e *EventuallyCycle*.
- *Cooperator*
    - Sempre coopera
- *Defector*
    - Nunca coopera
- *Alternator*
    - Alterna entre cooperar e não cooperar.
- *Random*
    - Tem 50% de chance de cooperar ou não.
- *Risky QLearner*
    - *QLearner* com taxa de aprendizagem $\alpha = 0,9$ e taxa de desconto intertemporal $\beta = 0,9$.
- *Win-Stay Lose-Shift: C*

- Inicia cooperando, em seguida repete a ação passada caso o retorno tiver sido 5 ou 3 e muda de ação caso contrário.

4.2 Pontuações Obtidas

O gráfico a seguir nos mostra a pontuação média obtida por cada estratégia, e como se deu a distribuição dos seus resultados no torneio. Observamos ordenamento diferente do experimento realizado por Harper, provavelmente pela ausência de estratégias menos eficientes sobre as quais algumas das utilizadas se sobressairiam mais que outras.

Figura 9 - Pontuação Média por Estratégia



Fonte: Autor

Os dados descritos no gráfico compõem a tabela a seguir:

Tabela 1 - Distribuição das Pontuações por Estratégia

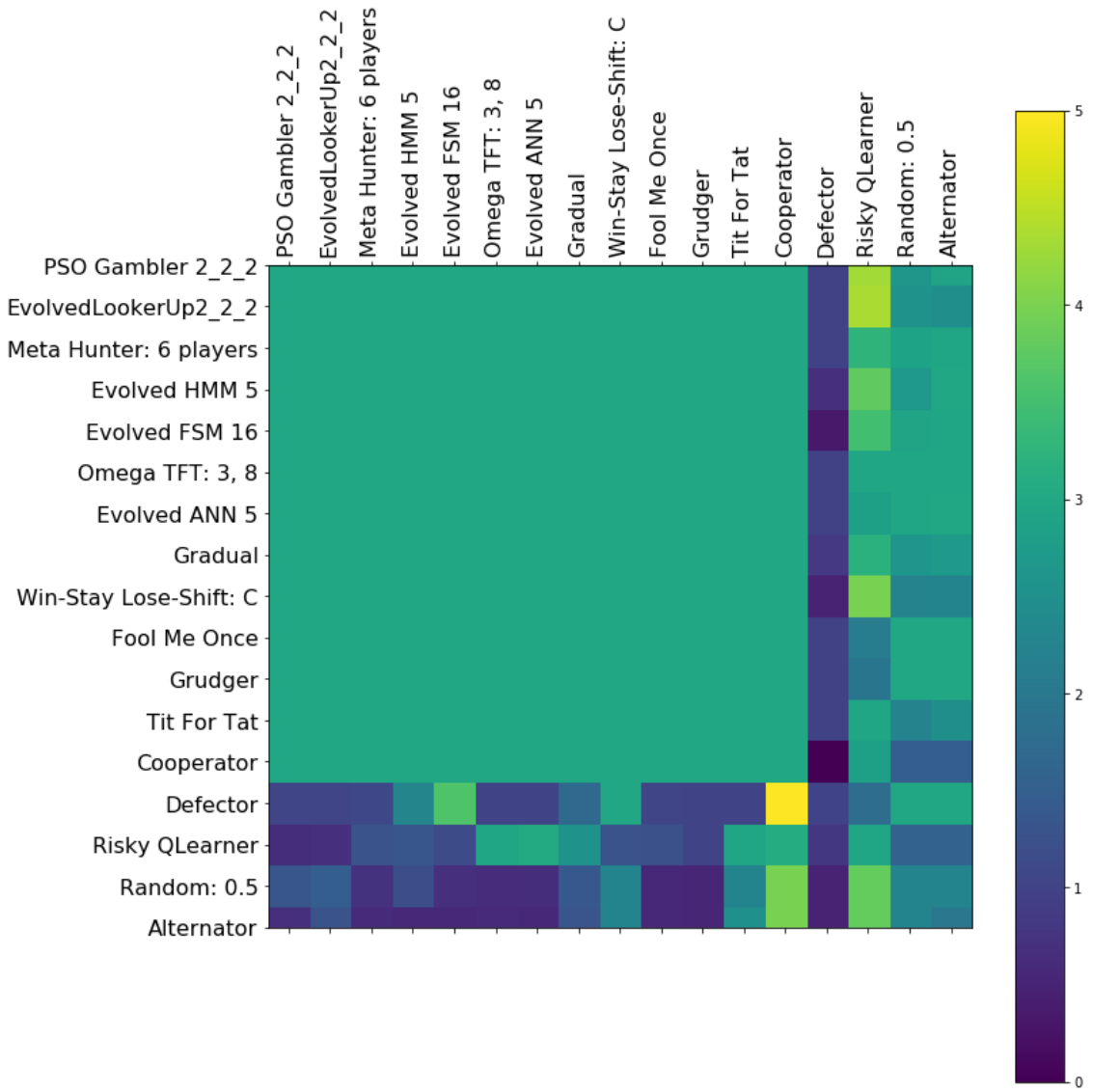
	Média	Desvio Padrão	Mínimo	5%	25%	50%	75%	95%	Máximo
PSO Gambler 2_2_2	2,929	0,020	2,802	2,895	2,915	2,929	2,942	2,962	2,995
EvolvedLookerUp2_2_2	2,901	0,024	2,696	2,861	2,884	2,901	2,917	2,941	2,972
Evolved HMM 5	2,884	0,020	2,814	2,851	2,871	2,884	2,897	2,917	2,966
Meta Hunter: 6 players	2,880	0,036	2,736	2,820	2,855	2,880	2,904	2,940	2,984
Omega TFT: 3, 8	2,865	0,010	2,746	2,849	2,858	2,865	2,872	2,881	2,898

Evolved ANN 5	2,859	0,012	2,750	2,839	2,851	2,859	2,867	2,879	2,967
Evolved FSM 16	2,857	0,045	2,711	2,783	2,827	2,857	2,888	2,932	2,954
Gradual	2,833	0,010	2,789	2,816	2,826	2,833	2,840	2,850	2,880
Fool Me Once	2,815	0,029	2,744	2,767	2,795	2,815	2,834	2,862	2,996
Win-Stay Lose-Shift: C	2,811	0,024	2,729	2,772	2,795	2,811	2,827	2,851	2,893
Grudger	2,806	0,027	2,742	2,762	2,788	2,806	2,825	2,851	3,005
Tit For Tat	2,792	0,005	2,772	2,785	2,789	2,792	2,796	2,800	2,810
Cooperator	2,616	0,007	2,581	2,604	2,611	2,616	2,621	2,628	2,648
Defector	1,979	0,026	1,902	1,936	1,961	1,979	1,996	2,022	2,188
Risky QLearner	1,700	0,071	1,331	1,583	1,652	1,700	1,749	1,818	2,091
Random: 0.5	1,519	0,051	1,358	1,435	1,484	1,519	1,553	1,603	1,729
Alternator	1,414	0,012	1,373	1,394	1,406	1,414	1,422	1,434	1,495

Fonte: Autor

Verificamos também os resultados médios específicos de cada confronto. Observamos a maior preferência pela cooperação nas estratégias que mais pontuaram. Relação essa que não se preserva no parâmetro seguinte.

Figura 10 - Pontuação Média por Confronto

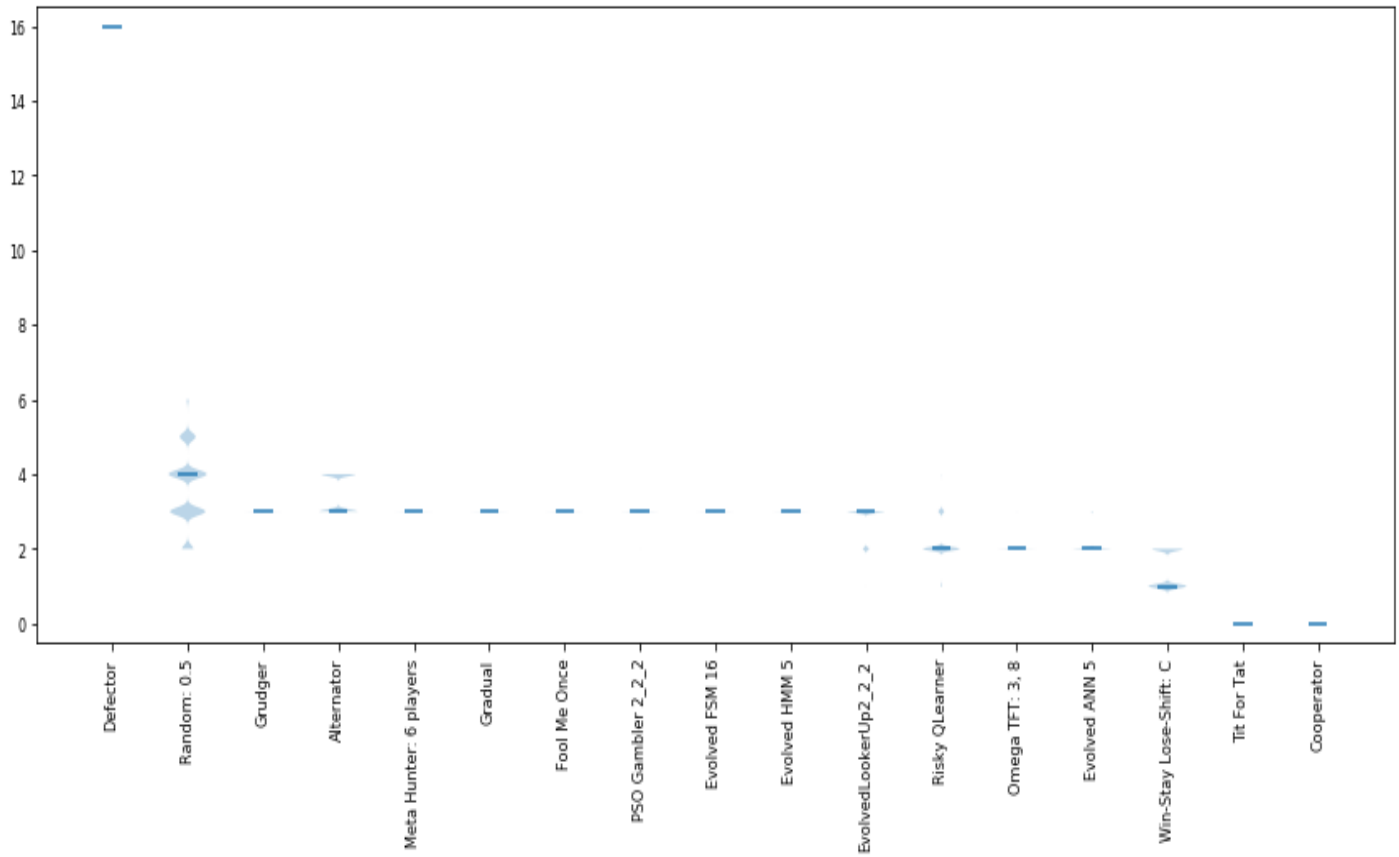


Fonte: Autor

4.3 Número de Vitórias

Obtemos também o gráfico e tabela referentes aos dados dos números de vitórias médio. Dado que o objetivo das estratégias baseadas em aprendizado de máquina não é vencer e sim pontuar o máximo possível, justificam-se suas colocações atrás de estratégias menos eficientes nesse sentido. O fato de *Defector* apresentar o maior número de vitórias é reflexo de seu comportamento de nunca cooperar, única ação que possibilita retorno maior que o do oponente.

Figura 11 - Número Médio de Vitórias por Estratégia



Fonte: Autor

Tabela 2 - Distribuição do Número de Vitórias por Estratégia

	Média	Desvio Padrão	Min	5%	25%	50%	75%	95%	Max
Defector	16,000	0,000	16	16,0	16,0	16,0	16,0	16,0	16
Random: 0.5	3,619	0,928	2	2,1	3,0	3,6	4,2	5,1	6
Alternator	3,473	0,499	3	2,7	3,1	3,5	3,8	4,3	4
Evolved HMM 5	3,000	0,006	2	3,0	3,0	3,0	3,0	3,0	3
Evolved FSM 16	3,000	0,006	2	3,0	3,0	3,0	3,0	3,0	3
Meta Hunter: 6 players	3,000	0,008	2	3,0	3,0	3,0	3,0	3,0	3
Grudger	3,000	0,008	2	3,0	3,0	3,0	3,0	3,0	3
Fool Me Once	2,999	0,024	2	3,0	3,0	3,0	3,0	3,0	3
Gradual	2,999	0,035	2	2,9	3,0	3,0	3,0	3,1	3
PSO Gambler 2_2_2	2,996	0,063	2	2,9	3,0	3,0	3,0	3,1	3
EvolvedLookerUp2_2_2	2,861	0,347	1	2,3	2,6	2,9	3,1	3,4	3
Risky QLearner	2,105	0,416	1	1,4	1,8	2,1	2,4	2,8	4
Evolved ANN 5	2,038	0,192	2	1,7	1,9	2,0	2,2	2,4	3
Omega TFT: 3, 8	2,004	0,065	2	1,9	2,0	2,0	2,0	2,1	3
Win-Stay Lose-Shift: C	1,447	0,497	0	0,6	1,1	1,4	1,8	2,3	2
Cooperator	0,000	0,000	0	0,0	0,0	0,0	0,0	0,0	0
Tit For Tat	0,000	0,000	0	0,0	0,0	0,0	0,0	0,0	0

Fonte: Autor

4.4 Estratégia sobreviventes

Ao realizar o processo de Moran 100 vezes, encerrando o processo quando restar somente uma estratégia em toda a população, obtemos a seguinte distribuição de estratégias sobreviventes. Segue o código utilizado:

```
import axelrod as axl


# Definimos os jogadores participantes
players = (axl.EvolvedLookerUp2_2_2(),
           axl.EvolvedHMM5(),
           axl.EvolvedFSM16(),
          axl.PSOGambler2_2_2(),
          axl.EvolvedANN5(),
          axl.FoolMeOnce(),
          axl.OmegaTFT(),
          axl.Gradual(),
```

```
            axl.MetaHunter(),
            axl.Cooperator(),
            axl.Defector(),
            axl.Alternator(),
            axl.TitForTat(),
            axl.Random(),
            axl.Grudger(),
            axl.RiskyQLearner(),
            axl.WinStayLoseShift())

# Inicializamos uma lista para guardar os vencedores de cada rodada
survivors = []
append = survivors.append

# Repetimos o Processo de Moran 100 vezes
mp = axl.MoranProcess(players)

for i in range(0, 100):
    mp.reset()
    mp.play()
    append(mp.winning_strategy_name)
    print(i)

# Geramos o gráfico referente à figura 12
mp.populations_plot()
```

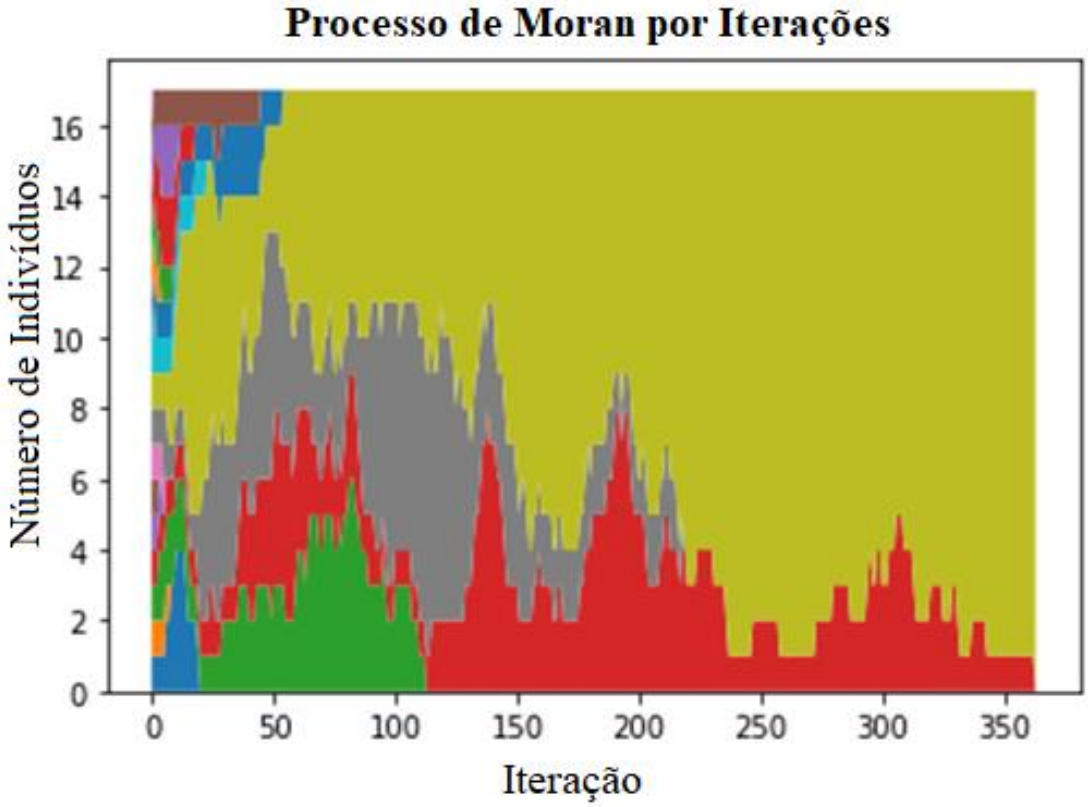
Tabela 3 - Número de Vitórias no Processo de Moran

	Número de vitórias
Evolved FSM 16	14
Win-Stay Lose-Shift: C	10
Evolved HMM 5	9
Grudger	9
Omega TFT: 3, 8	9
PSO Gambler 2_2_2	9
Evolved ANN 5	8
EvolvedLookerUp2_2_2	6
Meta Hunter: 6 players	6
Gradual	5
Cooperator	4
Fool Me Once	4
Tit For Tat	3
Alternator	2
Defector	1
Risky QLearner	1
Random: 0.5	0

Fonte: Autor

A seguir, vemos graficamente um dos processos realizados em que cada cor representa uma das estratégias:

Figura 12 - Exemplo Gráfico do Processo de Moran



Fonte: Autor

4.5 Panorama Geral

Notamos que, apesar do número reduzido de processos de Moran realizado, certas tendências podem ser vistas. O desempenho no torneio do dilema dos prisioneiros iterado está correlacionado com a capacidade de sobrevivência de dada estratégia. Dado que a pontuação adquirida é um dos fatores para a reprodução, existe um grau de causalidade entre ambos.

Tabela 4 - Ranqueamento das Estratégias

	Posição Pontuação	Posição Vitórias	Posição Processo de Moran
PSO Gambler 2_2_2	1	10	6
EvolvedLookerUp2_2_2	2	11	8
Evolved HMM 5	3	4	3
Meta Hunter: 6 players	4	6	9
Omega TFT: 3, 8	5	14	5
Evolved ANN 5	6	13	7
Evolved FSM 16	7	5	1
Gradual	8	9	10
Fool Me Once	9	8	12
Win-Stay Lose-Shift: C	10	15	2
Grudger	11	7	4
Tit For Tat	12	17	13
Cooperator	13	16	11
Defector	14	1	15
Risky QLearner	15	12	16
Random: 0.5	16	2	17
Alternator	17	3	14

Fonte: Autor

5. Conclusão

Verificamos que o resultado de uma estratégia no dilema dos prisioneiros iterado tem relação com sua capacidade de sobreviver ou não a um processo de Moran. Entretanto, realizamos nosso teste para somente interações uniformes entre indivíduos. Para alcançar resultados mais próximos da realidade, pode-se repetir essas simulações considerando distribuições espaciais e analisar como as interações são afetados por se darem entre indivíduos e entre grupos.

Apesar do resultado encontrado coincidir com o esperado intuitivamente, sugerimos que para maior exatidão, seja feita a repetição do experimento incluindo todas as estratégias disponíveis na biblioteca e realizando mais processos de Moran para maior demarcação da distribuição de vitórias. Também pode-se realizar a busca por parâmetros melhores para as estratégias baseadas em aprendizado de máquina, além de acrescentar o método de treinamento com gradiente descendente molecular.

É notável o desempenho superior das estratégias elaboradas pela máquina frente às elaboradas pelos pesquisadores, com a exceção da baseada em Q-Learning que carece de melhor análise para entender seu baixo desempenho. Além de fornecerem soluções facilmente reproduzíveis, contam com maior adaptabilidade ao problema enfrentado. Construímos programas na tentativa de replicar o comportamento humano, podemos fazer o caminho reverso e analisar se os supostos que definem o processo de decisão de uma máquina encontram correspondente nos creditados ao nosso comportamento racional.

Referências

AXELROD, R. Effective Choice in the Prisoner's DIlemma. **The Journal of Conflict Resolution 24, Vol. 24, No 1**, Março 1980a. 3-25.

AXELROD, R. More Effective Choice in the Prisoner's Dilemma. **The Journal of Conflict Resolution, Vol. 24, No 3**, Setembro 1980b. 379-403.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: [s.n.], 2016.

HARPER, M. et al. Reinforcement Learning Produces Dominant Strategies for the Iterated Prisoner's Dilemma. **PLoS ONE 12**, 2017.

KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. **Proceedings of ICNN'95**, 1995.

KNIGHT, V. et al. Axelrod-Python/Axelrod: v4.7.0. **Zenodo**, 2019. Disponivel em: <https://zenodo.org/record/3517155>. Acesso em: 1 Dezembro 2019.

KREPS, D. M. et al. Rational Cooperation in the Finitely Repeated Prisoners' DIlemma. **Journal of Economic Theory 27**, 1982. 245-252.

LIEBERMAN, E.; HAUERT, C.; NOWAK, M. A. Evolutionary dynamics on graphs. **Nature**, v. 433, p. 312-316, Janeiro 2005.

MORAN, P. A. P. Random Processes in Genetics. **Mathematical Proceedings of the Cambridge Philosophical Society Vol. 54(1)**, 1958. 60-71.

NOWAK, M. A. Five Rules for the Evolution of Cooperation. **Science**, v. 314, 2006.

NOWAK, M.; SIGMUND, K. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. **Nature**, v. 364, p. 56-58, Julho 1993.

TADELIS, S. **Game Theory An Introduction**. Princeton, New Jersey: Princeton University Press, 2013.