

Mecanização da Aprendizagem com Dispositivos Adaptativos: Conceitos e Aplicação

Resumo— A aprendizagem incremental requer que o mecanismo de aprendizagem seja baseado no acúmulo dinâmico da informação extraída das experiências realizadas. A palavra adaptatividade sugere a capacidade de modificação do conjunto de regras aprendidas em resposta a eventos que podem ocorrer durante o processo de aprendizagem, ou então autoajustes no conjunto de parâmetros. O objetivo deste trabalho é investigar questões relacionadas à utilização da adaptatividade no processo de aprendizagem de máquina, tais como mecanização da aprendizagem, representação do conhecimento, inferência e tomada de decisão. Para isso, propõe-se aqui a utilização de dispositivos adaptativos para representar o conhecimento adquirido através da aprendizagem incremental.

Palavras-chaves— Adaptatividade, Dispositivos Adaptativos, Aprendizagem de Máquina, Mecanismo de Aprendizagem.

I. INTRODUÇÃO

O TERMO aprendizagem de máquina (ML, no original em inglês *Machine Learning*), refere-se ao funcionamento de sistemas computacionais capazes de aprender e modificar o seu comportamento em resposta a estímulos externos, ou através de experiências acumuladas durante sua operação [1].

Aprendizagem de Máquina é uma área de pesquisa que estuda métodos, técnicas e ferramentas computacionais relacionadas à aquisição de novos conhecimentos e, novas habilidades para melhorar o desempenho de algoritmos por meio da experiência [8][1].

A adaptatividade é uma característica atribuída ao comportamento automodificável de sistemas computacionais. Este comportamento autônomo ocorre em resposta a estímulos de entrada e ao histórico de operação desses sistemas [12]. As pesquisas em adaptatividade investigam soluções para diversos problemas complexos de teoria da computação [11], de aprendizagem de máquina [18], de tomada de decisão [27] e de engenharia da computação [16], entre outros.

A área de aprendizagem de máquina tem-se mostrado uma rica fonte de pesquisa para a exploração prática das aplicações dos fundamentos da tecnologia adaptativa.

[14] apresentam o mecanismo de inferência ativo nos autômatos adaptativos. Como exemplo ilustrativo, o autômato adaptativo é utilizado para o aprendizado supervisionado de linguagens regulares. Um conjunto de amostras positivas¹ e negativas² da linguagem é submetido ao autômato, que deve inferir as sentenças aceitas ou rejeitadas.

[17] propõem um algoritmo de indução de árvores de decisão utilizando técnicas adaptativas, que combina estratégias sintáticas e estatísticas, chamado *AdapTree*.

Outras experiências bem-sucedidas em aprendizagem de

máquina utilizando dispositivos adaptativos incluem: aprendizagem de modelos para distribuição de espécies [15][25], classificação de padrões geométricos [5], decodificação do alfabeto de LIBRAS [3], localização de padrões em imagens [19], identificação de diagnósticos médicos [4] e mineração de dados [28], entre outras.

O processo de aprendizagem de máquina traz a tomada de decisão de forma crucial. A tomada de decisão exige um processo de raciocínio em que as informações já adquiridas e as novas informações, quando comparadas entre si, possam levar a novas informações e, com isso, influenciar o processo [26]. Esse processo de raciocínio é muitas vezes complexo e dinâmico, uma vez que as decisões devem ser flexíveis, pois eventualmente dependem de vários fatores e prioridades que nem sempre são fáceis de identificar antes de iniciar o processo de aprendizagem.

De acordo com [11], a resolução de problemas complexos e de natureza dinâmica utilizando a tecnologia adaptativa pode ser mais expressiva do que a utilização de métodos tradicionais, em alguns casos.

Em aprendizagem de máquina, por exemplo, uma das dificuldades está relacionada à representação do conhecimento humano em uma linguagem simbólica que tenha grande poder de expressividade. Métodos tradicionais para essa representação incluem o uso de regras de produção, árvores de decisão e redes Bayesianas, entre outros. O uso de dispositivos adaptativos pode agregar expressividade à representação do conhecimento e contribuir para o crescimento dessa área. Os autômatos adaptativos, por exemplo, além de possuírem o mesmo poder de expressão da Máquina de Turing [23], são eficientes e de fácil visualização, pois são baseados em modelos de autômatos finitos [10].

Contudo, a exploração de questões referentes à aprendizagem de máquina utilizando a tecnologia adaptativa se aplica, de maneira abrangente, ao tratamento de problemas de tomada de decisão.

A proposta deste trabalho tem como objetivo investigar questões relacionadas à utilização da adaptatividade no processo de aprendizagem de máquina, tais como mecanização da aprendizagem, representação do conhecimento, inferência e tomada de decisão.

Para atingir este objetivo, propõe-se aqui a utilização de dispositivos adaptativos para representar o conhecimento adquirido através da aprendizagem incremental. Além disso, é realizado um estudo de caso que combina aprendizagem de máquina com técnicas adaptativas para implementar um esquema de aprendizagem autônoma de estratégias, com o objetivo de vencer uma particular instância do jogo que é apresentado.

A utilização de dispositivos adaptativos em problemas reais e aplicar técnicas adaptativas para extração de regras

¹ Conjunto de sentenças que pertence à linguagem.

² Conjunto de sentenças que não pertence à linguagem.

desse dispositivos. A aplicação escolhida, apesar de simples, tem sido utilizada na aplicação de diferentes técnicas, tais como redes neurais artificiais [30] e distância de Hamilton [22]. Isso permite a fácil associação entre a utilização de técnicas adaptativas e as diferentes técnicas de aprendizado de máquina.

II. ADAPTATIVIDADE EM APRENDIZAGEM DE MÁQUINA

A adaptatividade possui uma ampla aplicação em aprendizagem de máquina devido às suas principais características, tais como a capacidade de reter regras, representar conhecimento, entre outras.

Em particular, os dispositivos adaptativos são adequados para modelar o processo de aprendizagem devido à maneira que operam. A inferência, aquisição e representação de conhecimento são notadamente favorecidas com utilização de técnicas adaptativas [13].

Os dispositivos adaptativos na mecanização da aprendizagem, por exemplo, permitem a inserção de um artifício de adaptação nas regras enquanto são aprendidas e isso proporciona ao algoritmo de aprendizagem a capacidade de automodificar o conjunto de regras aprendidas apenas em função das instancias de entrada do algoritmo. Outra característica da aplicação de técnicas adaptativas em aprendizagem de máquina é que a utilização de dispositivos adaptativos na representação do conhecimento proporciona um bom entendimento dos resultados obtidos com a aprendizagem. Eventualmente, a capacidade de entender as regras pode ser útil se comparado com outras técnicas de aprendizagem que fornecem bons resultados, mas possuem limitações para explicar como chegaram ao resultado fornecido.

Para representar o processo de mecanização da aprendizagem utilizando um dispositivo adaptativo um modelo para aprendizagem de um jogo foi definido. Este modelo chamado de mecanismo de aprendizado de um jogo é dividido nas seguintes etapas: Interface, Inferência e Memória.

O termo mecanismo de aprendizado se refere a essas três etapas.

A *Interface* estabelece a comunicação entre as informações externas e o mecanismo de aprendizado. Entende-se por informações externas qualquer estímulo de entrada provocado por um agente inteligente, seja ele humano (externo) ou a própria máquina (realimentação). De modo geral, a função da Interface é receber as informações e alimentar o mecanismo de aprendizado.

A *Inferência* está em constante ciclo de aprendizagem e é responsável por analisar as informações capturadas e agregar conhecimento sobre elas, para melhorar o desempenho na tomada de decisão. O conhecimento adquirido é representado por um conjunto de regras, que pode ser alterado a cada ciclo de aprendizagem. A situação corrente do conjunto de regras é representada por dispositivos adaptativos.

A *Memória* representa o conjunto de regras. Tudo que o dispositivo aprende deve ser armazenado em uma base de regras. Posteriormente, a base de regras é utilizada na tomada de decisão e, depois é possível inferir novas regras a partir dos

dados da base. Nos dispositivos adaptativos a memória está acoplada ao próprio dispositivo, que é uma das vantagens de sua utilização.

A Figura 1 mostra o mecanismo de aprendizado refinado para o aprendizado de estratégias para um jogo dinâmico com informação completa.

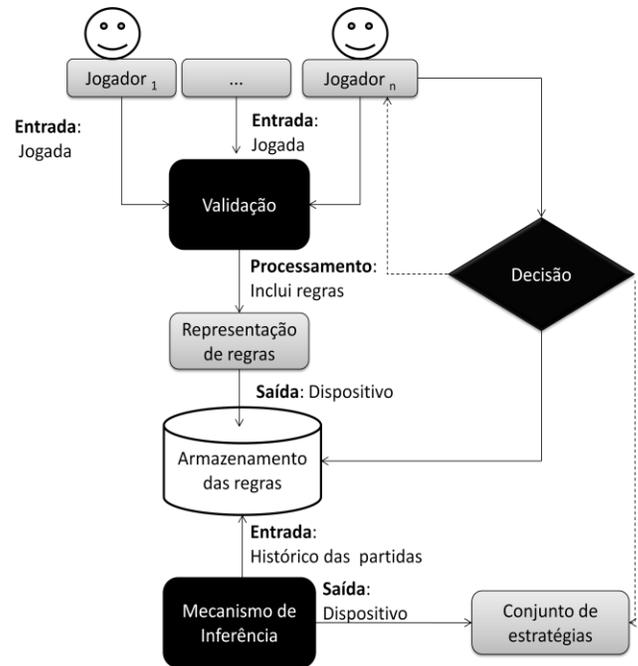


Figura 1. Mecanismo de aprendizagem de um jogo.

O mecanismo de aprendizado de um jogo mostra as relações entre os componentes que compõem o mecanismo de aprendizado, a saber: *Jogadores* (Jogador₁, Jogador₂,..., Jogador_n), *Validação*, *Representação de Regras*, *Armazenamento de Regras*, *Mecanismo de Inferência*, *Conjunto de Estratégias* e *Decisão*.

A componente *Jogadores* representa o canal de comunicação entre o mecanismo de aprendizado e o ambiente externo (Interface). Cada jogador deve fornecer uma *entrada* de dados para o funcionamento do mecanismo de aprendizado.

A *Validação* é a componente que recebe os estímulos de entrada.

No aprendizado de uma partida, cada jogada é uma regra a ser validada e deve ser capturada e posteriormente representada por um dispositivo apropriado.

Na *Representação de Regras*, um dispositivo adaptativo pode ser uma solução alternativa se considerarmos que, idealmente, o mecanismo de aprendizagem deve ser capaz de incorporar novas regras de forma autônoma.

Segundo [20], sistemas com características autônomas favorecem a utilização de modelos formais em sua especificação. As Máquinas de Estados Finitos, por exemplo, são modelos formais que favorecem a Engenharia de Software na especificação do comportamento de sistemas reativos, que se torna mais precisa e menos sujeita a ambiguidades. Esses modelos facilitam o entendimento do sistema.

Particularmente, os autômatos adaptativos podem evoluir gradualmente, modificando sua topologia inicial e, sucessivamente, inserir ou excluir transições de seu próprio conjunto de transições, como resultado da execução de ações adaptativas [14].

Neste trabalho, na *Representação das Regras*, a proposta é utilizar um autômato adaptativo para representar o conjunto de regras aprendidas em cada partida disputada. Porém, sua representação possui aplicações práticas restritas, se for considerado como dispositivo subjacente o autômato finito, que possui a informação de saída limitada à lógica binária aceita/rejeita [6]. Assim, com a finalidade de extrair cada movimento do autômato no decorrer de seu aprendizado, aplica-se uma extensão de autômatos conhecida como *Máquina de Mealy*, que possui saídas associadas às transições. A Máquina de *Mealy* é um autômato finito modificado capaz de gerar uma palavra de saída para cada transição da máquina, a qual inclusive pode ser vazia [6].

Na definição da Máquina de *Mealy* é incorporado um alfabeto de símbolos de saída Δ , que pode ser o mesmo alfabeto de entrada. A função de transição pode ser representada como um diagrama, assim como nos autômatos finitos, adicionando a cada transição a saída associada, quando diferente da palavra vazia.

A representação gráfica deste autômato é mostrada na Figura 2.

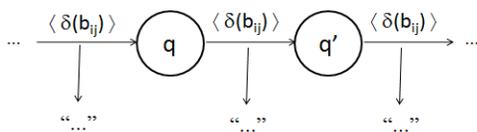


Figura 2 – A Máquina de Mealy.

A seguir a simbologia utilizada no diagrama de transição da Máquina de *Mealy*.

$\langle \dots \rangle$ entrada fornecida pelo usuário (ex.: por “Jogador”).
 “...” saída gerada pela transição (ex.: jogada realizada).

A entrada representa o conjunto (finito) de todos os possíveis símbolos que são estímulos de entrada válidos do autômato. Os estados q e q' representam os estados de origem e destino de uma transição.

Para fins de apresentação, as saídas da máquina serão ocultadas e será definido como representação equivalente o autômato adaptativo, apresentado na Figura 3. Assim a transição (q, $\langle \dots \rangle$) \rightarrow q' da Máquina de *Mealy* é equivalente (q, Θ) \rightarrow q' do autômato adaptativo.

As saídas do autômato marcam as escolhas efetuadas por “Jogador” e “Oponente”, a cada jogada e a cada partida. O caminho que determina o sucesso ou o insucesso na partida é mapeado através das saídas geradas pelo autômato.

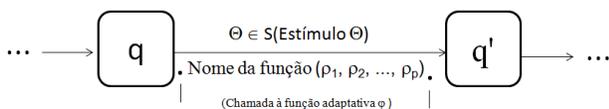


Figura 3 – Autômato Adaptativo com saída oculta.

Ao se pensar no jogo como uma máquina de estados que recebe uma entrada, sendo a entrada uma jogada $\delta(b_{ij})$, o tabuleiro começa no estado vazio e os estados mudam a cada vez que a máquina recebe uma entrada. Alguns estados são especiais quando são atingidos, no caso do jogo da velha, quando a partida acaba um estado final é atingido. Alguns estados finais determinam uma vitória para “Jogador” e outros, uma vitória para “Oponente”, enquanto os demais estados significam que o jogo terminou com um empate.

No *Armazenamento de Regras*, as jogadas capturadas pelos autômatos são armazenadas na memória do mecanismo de aprendizado e representam o histórico das partidas. A memória também é chamada de base de regras e as informações contidas nesta base podem ser utilizadas futuramente para melhorar o modelo de aprendizagem através da experiência. Em dispositivos adaptativos, as regras ficam armazenadas na estrutura do próprio dispositivo. Nos autômatos adaptativos, as regras ficam armazenadas através dos estados e transições, nas tabelas de decisão adaptativas, são representadas pelos valores de condições e ações e, nas árvores de decisão adaptativas, são representadas através dos caminhos que guiam uma decisão da raiz até as folhas.

Para exercitar a utilização da adaptatividade em aprendizagem de máquina foi escolhido um tipo particular de jogo, classificado como jogos dinâmicos. Esse é um tipo de jogo em que o processo de interação estratégica se desenvolve em etapas sucessivas. Nestes jogos, a decisão tomada por um jogador considera as decisões tomadas pelos demais jogadores. Assim, os jogadores fazem escolhas a partir do que os outros jogadores decidiram no passado. Além disso, nesse tipo de interação, as escolhas presentes exigem considerar as consequências futuras, uma vez que o oponente pode retaliar em etapas posteriores do jogo [29].

O matemático e filósofo alemão Ernst Friedrich Ferdinand Zermelo (1817-1953) provou um teorema que conduz à ideia de que jogos dinâmicos de duas pessoas, com ações sequenciais e informação completa, tais como os jogos da velha ou de xadrez, são perfeitamente determinados, ou seja, possuem solução definida. Neste caso, se um jogador souber aplicar a estratégia correta, dificilmente perderá o jogo.

Em linhas gerais, o processo de aprendizagem de uma estratégia para vencer uma instância de um jogo provoca a adaptação do comportamento dos jogadores para melhorar o desempenho na tomada de decisão. Cada tomador de decisão em um jogo é denominado jogador. Esse adota estratégias, que conduz às decisões para atingir seus objetivos, para cada informação que um jogador possa ter e em cada momento que ele é chamado a realizar uma ação.

A seguir é o mecanismo de aprendizado para um jogo é exemplificado através do processo de aprendizagem de uma estratégia para vencer o jogo da velha.

III. ESTUDO DE CASO

Este estudo de caso é realizado com um jogo de tabuleiro popularmente conhecido como Jogo da Velha. Os jogos de tabuleiros são casos de estudos interessantes em aprendizagem de máquina, se consideramos que o resultado dessa

experiência representa uma habilidade humana [29]. A capacidade de jogar, neste caso, representa muitas vezes um desafio aos jogadores que devem ser criativos para criar planos de ação, combinar estratégias e aprender com a experiência.

O nível de dificuldade de um jogo de tabuleiro varia de um simples jogo como o do jogo da velha ou um quebra-cabeça até os jogos que necessitam de uma estratégia mais sofisticada, como as aplicadas nos jogos de dama ou xadrez.

No jogo da velha participam dois jogadores, onde cada jogador tem a sua vez para marcar com o seu próprio símbolo (ex.: 'X' ou 'O') uma posição em um tabuleiro de tamanho 3x3.

As posições do tabuleiro do jogo da velha são representadas por b_{ij} , onde i é linha da matriz 3x3 e j é a coluna, representadas na Figura 4.

b_{11}	b_{12}	b_{13}
b_{21}	b_{22}	b_{23}
b_{31}	b_{32}	b_{33}

Figura 4. Posições do tabuleiro do jogo da velha.

As regras do jogo da velha são as seguintes [2]:

- O jogo começa com o tabuleiro vazio;
- Um jogador assume a vez marcando o seu símbolo em uma das posições vazias do tabuleiro;
- O jogador que forma uma linha reta completa (de ponta a ponta), ou seja, uma linha vertical, horizontal ou diagonal primeiro é o vencedor;
- O jogo empata se nenhuma linha reta é formada e não existem posições vazias no tabuleiro.

O estado inicial da partida é o tabuleiro vazio. Em cada etapa da partida, que será chamada de jogada, um jogador faz um movimento que consiste em colocar um 'X' em uma das posições vazias do tabuleiro e o outro jogador faz um movimento que consiste em colocar um 'O' em uma das posições vazias.

A Figura 5 mostra uma possível configuração de três tabuleiros de jogos.

(a)	(b)	(c)																											
<table border="1" style="width: 100px; height: 100px; border-collapse: collapse;"><tr><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td></tr></table>										<table border="1" style="width: 100px; height: 100px; border-collapse: collapse;"><tr><td>X</td><td>X</td><td>X</td></tr><tr><td> </td><td>O</td><td>O</td></tr><tr><td>O</td><td> </td><td>X</td></tr></table>	X	X	X		O	O	O		X	<table border="1" style="width: 100px; height: 100px; border-collapse: collapse;"><tr><td>X</td><td>O</td><td>X</td></tr><tr><td>X</td><td>O</td><td>O</td></tr><tr><td>O</td><td>X</td><td>X</td></tr></table>	X	O	X	X	O	O	O	X	X
X	X	X																											
	O	O																											
O		X																											
X	O	X																											
X	O	O																											
O	X	X																											

Figura 5. Três diferentes configurações para o tabuleiro do jogo: (a) Configuração inicial, (b) Jogador 'X' vence o jogo, (c) Jogo empatado.

Com a finalidade de obter um sistema de aprendizado capaz de representar computacionalmente um determinado problema, bem como a sua solução, é necessário descrever

objetos, processos e situações que fazem parte do seu domínio [9]. Existem diferentes linguagens de descrição com diferentes complexidades capazes de descrever exemplos (casos observados), hipóteses e conhecimento de domínio (ex.: lógica de atributos, lógica proposicional, lógica relacional, funções matemáticas, etc.).

Neste trabalho é adotado um tipo de linguagem para descrição de exemplos baseada na lógica de atributos, amplamente adotada em algoritmos de aprendizagem. Neste tipo de linguagem, um exemplo é descrito por um conjunto de atributos ou características que assumem diversos valores. Cada exemplo é formado pela conjunção do par atributo e valor que representa um caso observado, ou uma instância do problema (ex.: dor = sim \wedge febre=sim \wedge classe=doente). A classe é um atributo especial definido em alguns exemplos, que representa a saída do algoritmo para aquela instância. No caso do jogo da velha, cada partida é um exemplo de treinamento, as jogadas são atributos e a classe é o resultado obtido em cada partida, sendo os valores possíveis "P", "N" ou "E" respectivamente para resultados positivos, negativos ou empate.

Assim utilizando uma notação de aprendizagem de máquina, o jogo é representado da seguinte forma:

- $A = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}$ é o conjunto de atributos, onde: α_0 = "1ª jogada", α_1 = "2ª jogada", α_2 = "3ª jogada", α_3 = "4ª jogada", α_4 = "5ª jogada", α_5 = "6ª jogada", α_6 = "7ª jogada", α_7 = "8ª jogada", α_8 = "9ª jogada", α_9 = "Resultado da partida".
- $\delta(b_{ij})$ é um possível valor para cada atributo de A , onde $\delta = \{X, O\}$ e b_{ij} a posição escolhida do tabuleiro. Assim, cada atributo de α_0 a α_8 pode conter os seguintes valores $v_{\text{atributos}} = \{X(b_{11}), X(b_{12}), X(b_{13}), X(b_{21}), X(b_{22}), X(b_{23}), X(b_{31}), X(b_{32}), X(b_{33}), O(b_{11}), O(b_{12}), O(b_{13}), O(b_{21}), O(b_{22}), O(b_{23}), O(b_{31}), O(b_{32}), O(b_{33}), \theta\}$, onde θ significa posição do tabuleiro vazia e $\delta(b_{ij})$ a jogada escolhida.
- $\Omega = \{\omega_1, \omega_2, \omega_3\}$ são os possíveis valores de α_9 , onde: ω_1 = "P"; ω_2 = "N"; e ω_3 = "E", representando respectivamente os resultados: "Jogador" venceu a partida (Positivo), "Jogador" perdeu a partida (Negativo) e "Jogador" empatou a partida.
- T é o conjunto de exemplos treinamento composto por i partidas, onde $i = \{0, 1, 2, 3, \dots\}$. Cada partida é representada pelo conjunto ρ_i .

Considere o conjunto de dados de treinamento T , apresentados na Tabela 18.

TABELA I
DADOS DE TREINAMENTO REFERENTES A SEIS PARTIDAS DE TTT.

	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6
α_0	X(b_{22})	X(b_{11})	X(b_{12})	X(b_{31})	X(b_{22})	X(b_{22})
α_1	O(b_{11})	O(b_{33})	O(b_{31})	O(b_{22})	O(b_{13})	O(b_{11})
α_2	X(b_{33})	X(b_{31})	X(b_{33})	X(b_{11})	X(b_{33})	X(b_{13})
α_3	O(b_{12})	O(b_{21})	O(b_{11})	O(b_{21})	O(b_{11})	O(b_{31})
α_4	X(b_{13})	X(b_{13})	X(b_{13})	X(b_{23})	X(b_{21})	X(b_{21})
α_5	O(b_{31})	O(b_{22})	O(b_{31})	O(b_{12})	O(b_{12})	O(b_{23})
α_6	X(b_{23})	X(b_{12})	θ	X(b_{31})	θ	X(b_{12})
α_7	θ	θ	θ	O(b_{33})	θ	O(b_{31})
α_8	θ	θ	θ	X(b_{13})	θ	X(b_{33})
α_9	P	P	N	E	N	E

Baseado no mecanismo de aprendizagem de um jogo proposto e apresentado na Figura 1. O estudo de caso é descrito a seguir.

A representação de *Jogadores*, neste caso, é dada por dois jogadores, sendo um deles a própria aplicação, a partir deste momento chamada de “Jogador”, e o outro um humano, que será chamado de “Oponente”. A jogada de “Jogador” é representada pelo símbolo ‘X’ e a jogada de “Oponente” pelo símbolo ‘O’. Cada jogada é uma *entrada*, representada por $\delta(b_{ij})$. Para escolher uma jogada (*Decisão*, Figura 1), “Jogador” pode consultar a base de regras (*Armazenamento de regras*, Figura 1) e/ou adotar uma estratégia (*Conjunto de Estratégias*, Figura 1), caso estas já possuam regras.

Após Jogador ou Oponente escolher uma jogada, esta deve ser validada. A *Validação* recebe uma jogada $\delta(b_{ij})$ e determina se esta entrada é válida. Por exemplo, se um jogador escolhe a posição b_{23} do tabuleiro a componente deve verificar se a posição existe e está vazia, caso contrário, o jogador deve escolher uma nova posição. Após validação, a jogada passa a ser uma nova regra e deve ser representada adequadamente (*Representação de regras*, Figura 1).

Na fase de *Representação de regras*, cada regra ou jogada deve ser capturada por um autômato distinto, seja de “Jogador” ou de “Oponente”. Isso porque as jogadas de “Oponente” são importantes para que “Jogador” aprenda com a derrota. Assim, “Jogador” pode fazer escolhas a partir do que “Oponente” decidiu no passado. Cada exemplo ρ_i é capturado por um autômato.

A seguir, é descrito a técnica utilizada para a construção do autômato baseada em [14], que representa o comportamento dos jogadores.

- Passo 1:** Iniciar com um autômato conhecido;
- Passo 2:** Modificar o autômato incrementalmente para aceitar jogadas sucessivas na qual o jogador vence a partida;
- Passo 3:** Modificar o autômato incrementalmente para rejeitar jogadas sucessivas na qual o jogador perca a partida;
- Passo 4:** Modificar o autômato incrementalmente para aceitar jogadas sucessivas na qual jogador

empate a partida;

- Passo 5:** Repetir as ações até que o treinamento seja considerado aceitável.

Antes de cada jogada “Jogador” consulta a base de regras (*Decisão*, Figura 1, *Armazenamento de regras*, Figura 1). Para isso executa o algoritmo abaixo:

- Passo 1:** Para $i:= 0$ até n faça
Pesquisar no conjunto de transições do autômato se existe alguma transição
- Passo 2:** $?(q_i, \sigma?) \gg q?$, onde q_i é conhecido. Para todas as transições consumindo σ com estado inicial q_i e estado destino $q?$, retornar os valores $\sigma?$ e $q?$;
 - a) Se encontrar apenas uma transição: retorna os valores de $\sigma?$ e $q?$, aplique a regra;
 - b) Se não encontrar nenhuma transição: Criar uma nova transição consumindo um símbolo $A \neq \epsilon$ com estado inicial q_i e estado destino $q_{\#j}$, para $j:= i+1$. O símbolo # é utilizado para indicar que um novo estado foi gerado;
 - i) Remova a transição $(q_i, \sigma?) \gg q?$;
 - ii) Insira a transição $(q_i, A) \gg q_{\#j}$ onde A é uma posição vazia do tabuleiro escolhida e $j:= i+1$;
 - iii) Insira a transição: $(q_i, \epsilon[\cdot\phi_i]) \gg q_{\#j}$;
 - iv) Inserir a transição: $(q_{\#j}, \epsilon[\cdot\phi_j]) \gg q_f$;
- Passo 3:** Escolher a posição σ do tabuleiro.

A SEGUIR, A EVOLUÇÃO DO AUTÔMATO QUE REPRESENTA AS JOGADAS DE “JOGADOR” PARA A PRIMEIRA JOGADA α_0 DE ρ_1 CONSIDERANDO O CONJUNTO T DA

Tabela I.

Assim, para $i:= 0$, dado o autômato inicial representado na Figura 6, aplica-se a função a ação elementar de inspeção $?(q_i, \sigma?) \gg q?$.

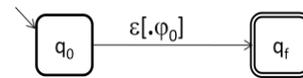


Figura 6 - Configuração Inicial do Autômato.

Ação elementar de Inspeção 1: Se encontrar uma transição retorna os valores de $\sigma?$ e $q?$. Aplique a regra.

O resultado da execução da ação elementar de inspeção para a Figura 6 retorna $\sigma = \epsilon[\cdot\phi_0]$ e $q = q_f$.

Ação elementar de Remoção 1: Remova a transição $(q_i, \sigma?) \gg q?$

Na sequência aplica-se a função a ação elementar de remoção $-(q_i, \varepsilon[\cdot\varphi_i]) \gg q_f$. O resultado da execução da ação elementar de remoção $-(q_0, \varepsilon[\cdot\varphi_0]) \gg q_f$ é apresentado na Figura 7.



Figura 7 - Resultado da execução da Ação elementar de Remoção

Ação Elementar de Inserção 1: Insira a transição $(q_i, A) \gg q_{\#j}$ onde A é uma posição vazia do tabuleiro escolhida e $j := i+1$.

A próxima ação elementar a ser executada é a de inserção $+(q_i, A) \gg q_j$. Que neste momento é representada por $(q_0, X(b_{22})) \gg q_{\#1}$.

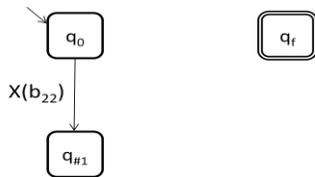


Figura 8 - Resultado da execução da Ação elementar de Inserção $+(q_i, A) \gg q_j$

Ação Elementar de Inserção 2: Insira a transição: $+(q_i, \varepsilon[\cdot\varphi_i]) \gg q_{\#j}$

Outra ação elementar de inserção é executada na sequência: $+(q_i, \varepsilon[\cdot\varphi_i]) \gg q_f$, neste caso $+(q_0, \varepsilon[\cdot\varphi_0]) \gg q_f$.

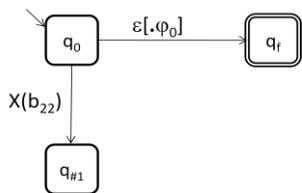


Figura 9 - Resultado da execução da Ação elementar de Inserção $+(q_i, \varepsilon[\cdot\varphi_i]) \gg q_f$

Ação Elementar de Inserção 3: Insira a transição: $+(q_i, \varepsilon[\cdot\varphi_i]) \gg q_f$. Neste caso, a execução de $(q_{\#1}, \varepsilon[\cdot\varphi_1]) \gg q_f$ resulta no autômato da Figura 10.

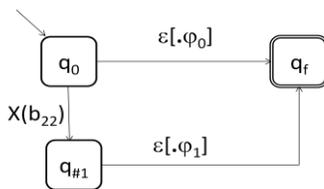


Figura 10 - Resultado da execução da Ação elementar de Inserção $(q_{\#1}, \varepsilon[\cdot\varphi_1]) \gg q_f$

O AUTÔMATO É RESULTADO DA PRIMEIRA JOGADA, CONFORME EXEMPLO DE TREINAMENTO ρ_1 DA

Tabela I. No final da partida, uma possível configuração final do autômato adaptativo inferido, após quatro jogadas de "Jogador", é representada pela Figura 11.

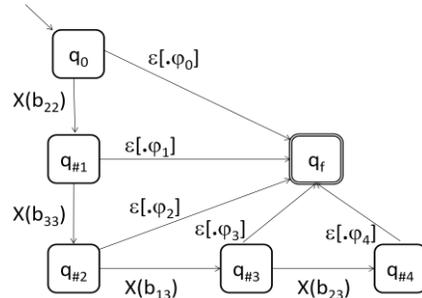


Figura 11 – Autômato Coletor de jogadas de "Jogador".

As funções adaptativas do autômato impõem uma ordenação, pré-determinada, ao conjunto de atributos. É essa ordenação que determina em que jogada da partida irão ocorrer os diversos atributos. Na medida em que exemplos vão sendo submetidos ao dispositivo, a estrutura vai se modificando, na forma de um autômato.

O MESMO ALGORITMO CAPTURA AS JOGADAS DE "OPONENTE", QUE CONSIDERANDO O EXEMPLO DE TREINAMENTO ρ_1 DA

Tabela I, apresenta na Figura 12 uma possível configuração final do autômato adaptativo inferido.

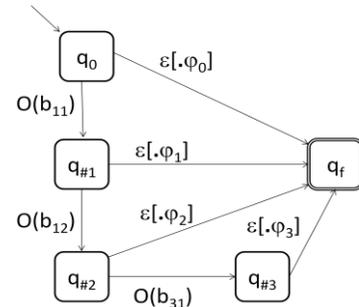


Figura 12 – Autômato Coletor de jogadas de "Oponente".

A primeira partida tem a função de calibrar o mecanismo de aprendizado, "Jogador" realiza suas jogadas sem auxílio do tomador de decisão (*Decisão*, Figura 1), pois ainda não existe um histórico de operações (partidas) que possa ser consultado (*Armazenamento de Regras*, Figura 1).

O mecanismo de aprendizado captura em diferentes autômatos tanto as jogadas de "Jogador" quanto às de "Oponente" e, dependendo do resultado da partida (venceu, perdeu ou empatou), repete as jogadas nas partidas posteriores.

Neste trabalho, se considerarmos que o objetivo final é obter um conjunto de regras para auxílio à tomada de decisão em um jogo de tabuleiro, é conveniente à utilização da Tabela de decisão Adaptativa (TDA). [13] considera que sistemas que envolvem tomadas de decisão podem utilizar Tabelas de Decisão Adaptativas para a construção de sistemas inteligentes que recebem dados de treinamento, aprendem a classificá-los e, futuramente, classificam novos dados.

O conjunto de transições (jogadas) resultante de cada

partida é inserido em uma Tabela de Decisão Adaptativa, porém é relevante destacar que o próprio autômato adaptativo pode representar o conjunto de regras adquiridas. Apesar disso, diferentes dispositivos adaptativos podem apresentar uma solução mais aderente a um problema em particular, assim a TDA representa a memória do mecanismo de aprendizagem.

Outra motivação para a representação utilizando uma TDA está fundamentada em [12] (*apud* [28]), que articula “[...] os dispositivos adaptativos dirigidos por regras podem dar maior flexibilidade às tabelas de decisão, permitindo, não somente a consulta às regras, como também a inclusão e a exclusão de regras durante a operação do dispositivo, transformando, assim, a tabela de decisão numa ferramenta mais poderosa”.

A seguir é descrito o algoritmo para construção da TDA.

Passo 1: Iniciar uma TDA conhecida;

Passo 2: Buscar no conjunto de regras R uma ou mais regras que satisfaçam as condições de entrada P_k ;

- Caso encontre uma ou mais regras aplicáveis, inclua todas elas em um conjunto de regras aplicáveis;
- Caso contrário, prepara o dispositivo para uma nova entrada, aplicando o **Passo 4**;

Passo 3: Extrair as regras aplicáveis do conjunto de regras:

- Caso apenas uma regra seja aplicável, o dispositivo executa a ação determinada pela regra; essa situação define uma operação determinística;
- Caso mais de uma regra seja aplicável, uma situação de não determinismo é detectada. As regras são executadas em paralelo;

Passo 4: Decodificar as ações.

- No caso de uma regra não adaptativa, extrair as ações associadas às regras;
- No caso de uma regra adaptativa, extrair as ações adaptativas associadas às regras.

Passo 5: Executar as ações:

- No caso de uma regra não adaptativa, executar as rotinas semânticas associadas a cada ação.
- No caso de uma regra adaptativa:
 - Para uma função adaptativa posterior, executar o **Passo 5.a** e na sequência o **Passo 5.c**;
- Executar as ações adaptativas.

A configuração inicial da TDA está representada na Tabela II. Na configuração inicial apenas uma regra adaptativa, P_a , é inserida na TDA. No decorrer de sua execução, novas regras vão sendo inseridas.

As jogadas capturadas pelo autômato são representadas por um conjunto de regras do tipo “Se *condições então Ação*”, onde as condições são as jogadas válidas que levam o “Jogador” a um resultado. A ação é o resultado obtido na partida. O conjunto de regras é utilizado para jogar novas partidas.

A saída do algoritmo é um reconhecedor de partidas τ , neste caso representado por um dispositivo do tipo tabela de decisão adaptativa. Dado uma nova partida χ , o reconhecedor deve verificar se a partida já foi realizada anteriormente, caso já tenha sido realizada aplica a regra aprendida na partida.

Caso contrário, aprende uma nova regra.

TABELA II
CONFIGURAÇÃO INICIAL DA TABELA DE DECISÃO ADAPTATIVA.

Cabeçalhos (Tags)		H	?	+	P_a	
Condições	Condição ₁		P_0	V_0	-	
	Condição ₂		P_1	V_1	-	
	Condição ₃		P_2	V_2	-	
	Condição ₄		P_3	V_3	-	
	Condição ₅		P_4	V_4	-	
	Condição ₆		P_5	V_5	-	
	Condição ₇		P_6	V_6	-	
	Condição ₈		P_7	V_7	-	
Ações	Ação ₁		P_8	V_8	?	
Funções Adaptativas	Anterior	f_1	B	√	√	
	Posterior					
	Parâmetros	P_0	P			P_0
	
	Variáveis	P_2	P			P_2
		V_0	V			
				
Geradores	V_2	V				

A partir da segunda partida, representada pelo conjunto ρ_2 da Tabela I, já existem regras na base de regras (*Memória*) e o “Jogador” pode decidir as jogadas de acordo com as jogadas passadas. Porém, alguns movimentos podem ser aleatórios, sem qualquer auxílio do tomador de decisões, considerando que a posição do tabuleiro está ocupada. A *Decisão* é baseada nas regras armazenadas na Tabela de Decisão Adaptativa, caso nenhuma regra seja aplicável, deve-se decidir aleatoriamente e aprender uma nova regra.

O RESULTADO DA EXECUÇÃO DA TDA PARA O CONJUNTO DE JOGADAS CAPTURADAS PELOS AUTÔMATOS DE FIGURA 11 E FIGURA 12 ATRAVÉS DO CONJUNTO DE TREINAMENTO DA

Tabela I é o conjunto de partidas $P = \{P_1, P_2, \dots, P_k\}$, onde k é o número de partidas, conforme Tabela III. As condições representam a ordem das jogadas capturadas pelo autômato, tanto de “Jogador” quanto de “Oponente”.

O exemplo a seguir ilustra a operação da TDA para representar o conhecimento adquirido pelo autômato.

Considere o exemplo de uma partida $\chi = \{X(b_{22}), O(b_{11}), X(b_{13}), O(b_{31}), X(b_{21}), O(b_{23}), X(b_{12}), O(b_{31}), X(b_{33}), \text{Empate}\}$.

Neste caso, não existe uma regra aplicável, ou seja, uma partida já realizada com a mesma configuração. Porém, é sabido que χ é um exemplo de partida que “Jogador” venceu e deve ser adicionado à TDA. Na TDA a regra adaptativa P_a é satisfeita toda vez que nenhuma regra não adaptativa for aplicável. A regra P_a possui a seguinte função adaptativa

associada:

TABELA III
TABELA DE DECISÃO ADAPTATIVA APÓS 6 MODIFICAÇÕES.

Cabeçalhos (Tags)		H	?	+	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P _a	
Condições	Jogada_1_x	p0	v0	X(b22)	X(b11)	X(b12)	X(b31)	X(b22)	X(b22)	X(b22)	θ	
	Jogada_1_o	p1	v1	O(b11)	O(b33)	O(b31)	O(b22)	O(b13)	O(b11)	O(b11)	θ	
	Jogada_2_x	p2	v2	X(b33)	X(b31)	X(b33)	X(b11)	X(b33)	X(b13)	X(b13)	θ	
	Jogada_2_o	p3	v3	O(b12)	O(b21)	O(b11)	O(b21)	O(b11)	O(b31)	O(b31)	θ	
	Jogada_3_x	p4	v4	X(b13)	X(b13)	X(b13)	X(b23)	X(b21)	X(b21)	X(b21)	θ	
	Jogada_3_o	p5	v5	O(b31)	O(b22)	O(b31)	O(b12)	O(b12)	O(b23)	O(b23)	θ	
	Jogada_4_x	p6	v6	X(b23)	X(b12)	θ	X(b31)	θ	X(b12)	θ	θ	
	Jogada_4_o	p7	v7	θ	θ	θ	O(b33)	θ	O(b31)	θ	θ	
	Jogada_5_x	p8	v8	θ	θ	θ	X(b13)	θ	X(b33)	θ	θ	
Ações	Resultado	p9	v9	P	P	N	E	N	E	?	?	
	Anterior	f1	B	√	√						√	
Funções Adaptativas	Posterior											
	Parâmetros	p0	P									P0
	
		p9	P									P9
	Variáveis	v0	V									
										
		v9	V									

Exemplo do pseudocódigo para declaração de uma função adaptativa.

```

Função Adaptativa f1(Parâmetros: p1, p2,
p3, p4, p5, p6, p7, p8, p9) {
  Variáveis: v1, v2, v3, v4, v5, v6, v7, v8,
v9
  Geradores: necessários para gerar novas
  regras
  Ações elementares Δ {
    δ1:?[v1, v2, v3, v4, v5, v6, v7, v8, v9]
    (Existe uma regra que satisfaça as
    condições χ).
    δ2:+[v1, v2, v3, v4, v5, v6, v7, v8,
v9](Inserir uma regra Pk).
  }
}

```

No caso de aplicação da regra P_a em χ , os valores dos atributos são atribuídos aos parâmetros da função f_1 . A função recebe os valores como parâmetro e atribui os valores dos parâmetros para as variáveis. Na sequência, a ação elementar de consulta pesquisa se existe uma regra em R que satisfaça χ e a ação elementar de inserção inclui uma regra P_k , para $n=1, 2, \dots, k-1$, onde n é o número de regras atuais da TDA.

No domínio do Jogo da velha, cada jogador enfrenta um desafio básico: como criar uma estratégia flexível o suficiente para lidar com certas situações de mudança a cada jogada. A estratégia é formada por um conjunto de regras que permite aumentar as chances de vitória de um jogador. O mecanismo de aprendizagem deve aprender não somente a aplicar jogadas válidas, mas também como aplicar uma jogada que garanta uma vitória ou, no pior caso, leve a partida a um empate.

Após várias partidas, a base de regras se torna uma rica fonte de dados que representa o comportamento dos jogadores e estimula a execução de uma próxima etapa do aprendizado, que é a descoberta de estratégias de jogo.

O objetivo da descoberta de uma estratégia para jogar o Jogo da velha considera que os jogadores devem usar os meios mais adequados para atingir seus objetivos, seja vencer ou não perder. Assim, após diversas partidas o jogador acumula experiências e essa experiência pode conduzir os jogadores a um comportamento estratégico. Cada jogador toma decisões considerando que elas terão efeitos sobre os outros jogadores, bem como as decisões dos outros jogadores terão efeitos sobre suas decisões. A estratégia tenta minimizar o efeito negativo de cada decisão.

Esta etapa de aprendizagem ou descoberta de uma estratégia é representada pelo *Mecanismo de Inferência*.

O *Mecanismo de Inferência* atua sobre o conjunto de regras aprendidas, armazenado na base de regras, e infere um conjunto de estratégias que poderão ser consultadas na tomada de decisão dos jogadores.

Neste trabalho, a aprendizagem de estratégias é baseada nas medidas de ganho de informação e entropia, similar ao ID3[21].

A entropia, no contexto da teoria de informação, pode ser considerada como uma medida da quantidade de informação que uma pessoa necessita para organizar seus conhecimentos e descobrir uma regra. Analogamente, será adotada como a medida para organizar o conhecimento adquirido por um jogador e descobrir um conjunto de regras para realizar as jogadas de forma estratégica.

Quanto mais alternativas uma sistema de tomada de decisão possui (ex.: mais jogadas possíveis), mais informações são necessárias para aprender a tomá-las (maior entropia). Se um sistema de tomada de decisão não tem alternativas, não é necessária nenhuma informação (a entropia é 0).

A seguir é descrito um algoritmo para inferência de regras utilizando ganho de informação.

- 1: Para cada jogada K de “Jogador” faça:
 - a. Crie um estado que representa a jogada K.
 - b. Calcule o ganho de informação de cada jogada (valor de atributo).

$$Ganho(S, A) = Entropia(S) - \sum_{v \in V} \frac{|S_v|}{|S|} \cdot Entropia(S_v)$$

Onde:

$$Entropia(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

- c. Estender a árvore adicionando um ramo para cada valor do atributo.
- d. Dividir o conjunto de exemplos P (tendo em conta o valor do atributo escolhido) e passe os subconjuntos para as folhas da árvore.
- e. Repetir os passos para cada novo nó gerado.

Seja P o conjunto de exemplos de treinamento armazenado na base de regras do mecanismo de aprendizagem. A base de

regras é composta por 958 exemplos (626 exemplos de vitória e 332 exemplos de derrota) extraídos do repositório de dados UCI *Machine Learning Repository*. Estes dados foram tratados de forma que a ordem das jogadas fossem representados no conjunto. Seja 9 o total de atributos, cada atributo corresponde a um uma posição do tabuleiro do Jogo da Velha.

Aplicando a fórmula ao subconjunto S temos:

$$\text{Entropia (S)} = -\frac{626}{958} \log_2 \frac{626}{958} - \frac{332}{958} \log_2 \frac{332}{958} = 0.93$$

Considere os resultados obtidos com o calculo do ganho de informação: $\text{Ganho}(S, b_{11}) = 0.93 - 0.91 = 0.02$, $\text{Ganho}(S, b_{12}) = 0.93 - 0.92 = 0.01$, $\text{Ganho}(S, b_{13}) = 0.93 - 0.91 = 0.02$, $\text{Ganho}(S, b_{21}) = 0.93 - 0.92 = 0.01$, $\text{Ganho}(S, b_{22}) = 0.93 - 0.84 = 0.09$, $\text{Ganho}(S, b_{23}) = 0.93 - 0.96 = -0.03$, $\text{Ganho}(S, b_{31}) = 0.93 - 0.91 = 0.02$, $\text{Ganho}(S, b_{32}) = 0.93 - 0.92 = 0.01$, $\text{Ganho}(S, b_{33}) = 0.93 - 0.91 = 0.01$.

Na primeira iteração do algoritmo, é possível concluir que a jogada com maior ganho de informação é escolher a posição central do tabuleiro, ou seja, $\delta(b_{22})$. A partir da segunda jogada, o ganho de informação é calculado baseado na jogada de “Oponente”.

A Figura 13 mostra a possibilidade de jogadas e seus valores correspondentes.

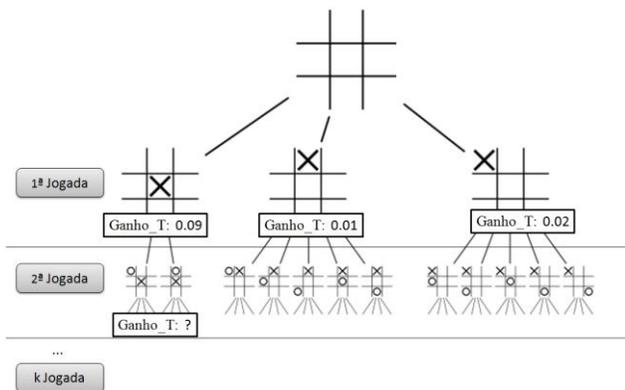


Figura 13 – Representação das possibilidades de jogadas com ganho de informação.

Na sequência, os cálculos de ganho de informação são refeitos, considerando cada possível jogada de “Oponente”, e assim sucessivamente até obter uma árvore que seja capaz de auxiliar o jogador na tomada de decisão aplicando uma estratégia.

A seguir é descrito como analisar o comportamento dos jogadores a partir das regras geradas do conjunto de estratégias. As estratégias de “Jogador” consideram que este deve iniciar a partida, porém é perfeitamente possível extrair uma estratégia que permita a “Oponente” iniciar a partida.

A *Decisão* das jogadas utilizando uma estratégia é baseada no *Conjunto de estratégias*.

Existem diversos algoritmos disponíveis para decidir as jogadas de “Jogador” em resposta aos movimentos do “Oponente” durante o jogo. A seguir é apresentada uma abordagem heurística simples para automatizar os movimentos do computador, para o nível iniciante.

Passo 1: Verifique se existe um movimento que o computador pode fazer de modo que obterá maior ganho de informação.

- a) Em caso afirmativo, preencher o quadrado relevante.
- b) Senão, verifique se há um movimento que irá bloquear uma vitória para o outro jogador e preencha o quadrado correspondente.
- c) Senão, preencha o quadrado que fica na linha/coluna com o número máximo de células livres de marcas.

O Jogo da Velha com estratégias deve permitir que, para uma particular instância do jogo, “Jogador” vença ou pelo menos empate a partida. As estratégias inferidas a partir da base de regras são as seguintes:

1. Se $Jogada_1_x$ então escolha $X(b_{22})$: “Jogador” deve fazer a jogada de abertura, neste caso a melhor jogada é escolher a posição central do tabuleiro com maior ganho de informação (Ganho_T: 0.09).

Na sequência, o adversário pode marcar uma lateral ou um canto. Por exemplo, caso o “Oponente” marque um dos cantos, é observado que “Jogador” deve escolher o canto oposto, formando uma linha diagonal, conforme a Figura 14.

2. Se $Jogada_2_o = O(b_{33})$ então escolha $X(b_{11})$: Dependendo da opção escolhida pelo “Oponente”, “Jogador” deve escolher aquela com maior ganho de informação. Para a jogada $O(b_{33})$, $O(b_{11})$, $O(b_{13})$ e $O(b_{31})$ a melhor opção é o canto oposto, caso ele esteja vazio. Caso contrário, escolhe-se a próxima opção com melhor ganho de informação, e assim sucessivamente.

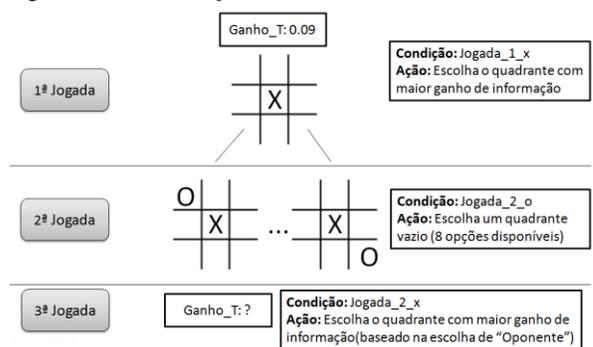


Figura 14 – “Jogador” escolhe a 3ª jogada baseada na escolha do oponente.

Se o próximo movimento de “Oponente” for adjacente da marca anterior, “Jogador” vai ter boas chances de vencer, como mostrado Figura 15.

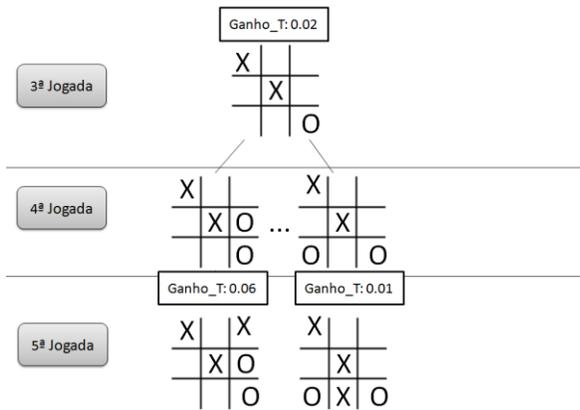


Figura 15 – “Jogador” escolhe a 5ª jogada baseada na escolha do oponente.

- Se $Jogada_2_o = O(b_{23})$ então escolha $X(b_{13})$: Neste caso, o “Jogador” deve retaliar o “Oponente”. O termo retaliar significa que “Jogador” deve bloquear todas as tentativas de vencer do “Oponente”.

Considerando que “Jogador” tome as decisões baseadas nas regras que definem uma estratégia, o clássico Jogo da Velha pode ser jogado de modo que “Jogador” é direcionado para uma vitória ou um empate.

IV. DISCUSSÃO DOS RESULTADOS

A aprendizagem de máquina usando adaptatividade considera a integração de técnicas de aprendizagem de máquina, determinísticas ou estatísticas, e técnicas adaptativas para a construção de sistemas de aprendizado.

Na aprendizagem tradicional, ou seja, sem a utilização de técnicas adaptativas, os ajustes para a melhoria no processo de aprendizagem, em geral, ocorrem na fase de treinamento. Porém, muitas informações relevantes podem ser capturadas na fase de utilização do sistema (ex.: na fase de classificação).

Em geral, isso ocorre quando o problema é de natureza dinâmica. Com a utilização de técnicas adaptativas, através dos dispositivos adaptativos, é permitido descrever de forma natural os aspectos dinâmicos dos problemas de aprendizagem.

A aprendizagem adaptativa é uma forma de adaptar gradualmente o modelo aprendido. A adaptatividade pode ser útil para detectar ajustes no conjunto de regras e modificar o modelo, de maneira incremental. Embora existam maneiras de refazer a estrutura de dispositivos convencionais, tais como árvores e tabelas de decisão, após a obtenção de novos dados, refazer o conjunto de regras pode ser ineficiente, podendo levar à perda parcial ou total de informações que haviam sido anteriormente aprendidas.

A grande vantagem da adaptatividade sobre outras técnicas correntemente utilizadas para a formulação de modelos de representação e de manipulação do conhecimento reside no fato de que [24]:

(a) a informação total, encerrada no dispositivo adaptativo, está representada integralmente no conjunto de regras. A memória é representada pelo próprio dispositivo;

(b) a aprendizagem angariada em cada passo adaptativo do dispositivo encontra-se integralmente confinada à variação sofrida pelo conjunto de regras. É possível determinar o que foi aprendido pelo dispositivo adaptativo analisando as alterações de configuração durante o processo de aprendizagem;

Em segundo plano, a adaptatividade permite que as diversas características inerentes aos problemas reais de aprendizagem, tais como comportamento dinâmico e estocástico, possam ser tratadas de forma transparente. A interação do especialista no mecanismo de aprendizagem é simplificada e não apresenta complexidades técnicas, pois estas devem ficar a cargo do mecanismo de aprendizagem. De fato, o especialista deve manter o controle da definição do problema, incluindo o conhecimento sobre o domínio, restrições ou preferências através da definição das funções adaptativas. Esta é a parte que, dependendo do nível da adaptatividade, não pode ser automatizada.

Uma vantagem comparada a outros mecanismos de representação, tais como naïve Bayes, é que o autômato armazena a sequência das jogadas, assim é possível saber a ordem em que as jogadas foram efetuadas no tabuleiro. Essa relação de ordem das jogadas fica incorporada na estrutura do autômato.

Contudo, os métodos que utilizam dispositivos simbólicos para representar o que foi aprendido são adequados quando é desejável obter um conjunto de regras mais compreensível por especialistas humanos. Considerando que esses dispositivos podem ser representados por um conjunto de regras do tipo “Se...então”, os dispositivos adaptativos se enquadram na representação simbólica das regras aprendidas. Tal formato utilizado em dispositivos adaptativos fornece uma perspectiva unificada sob a qual as regras de um mecanismo de aprendizado podem ser convertidas e analisadas.

Outro aspecto percebido, similar ao observado em dispositivos baseados em regras, tais como árvores de decisão, expressões lógicas, regras de produção e tabelas de decisão, é a naturalidade que se tem em modelar os problemas de aprendizagem de máquina utilizando diferentes dispositivos adaptativos. Os autômatos adaptativos, as tabelas de decisão adaptativas e as árvores de decisão são equivalentes e facilmente convertidos.

Em particular, os dispositivos adaptativos, tais como autômatos adaptativos, árvores ou tabelas de decisão adaptativas, permitem a representação estrutural das regras. Segundo [7], a criação de estruturas simbólicas que sejam compreensíveis e utilizadas por modelos mentais na aprendizagem é mais interessante do que os modelos estatísticos.

A combinação equilibrada entre a obtenção de modelos de aprendizagem compreensíveis e expressivos pode ser adquirida com a utilização de dispositivos adaptativos. Tanto os autômatos adaptativos quanto as tabelas de decisão adaptativas, e intuitivamente, outros dispositivos adaptativos utilizados para representar o conhecimento adquirido em sistemas de aprendizagem, facilitam o entendimento das regras. A estrutura simples e compreensível dos dispositivos

adaptativos são ferramentas úteis para a descoberta do conhecimento.

V. CONSIDERAÇÕES FINAIS

A aprendizagem de máquina utilizando tecnologia adaptativa pode ser considerada uma técnica inspirada na aprendizagem indutiva supervisionada com o objetivo de melhorar o entendimento das regras, mas principalmente cumprir exigências de eficiência. Na técnica, uma base de regras é utilizada para representar o comportamento inteligente de um jogador em função de experiências passadas.

As regras (ex.: uma jogada é uma regra) são extraídas da memória e aplicadas durante as partidas podendo gerar novas regras que alteram o comportamento dos jogadores ou a base de regras (inclusão ou exclusão de regras). A probabilidade de aplicação de uma regra que leva o jogador a uma vitória é proporcional ao ganho de informação associada a esta regra (ex.: jogadas com maior ganho de informação têm maior probabilidade de vencer a partida).

A aplicação da medida estatística baseada no ganho de informação pode ser considerada uma forma de avaliar os dispositivos adaptativos. A avaliação é interessante para testar a eficácia e eficiência dos dispositivos adaptativos. Neste caso, não parece ser vantajoso considerar que os dispositivos adaptativos apenas melhoram a expressividade e a compreensibilidade da solução, comparados a dispositivos não adaptativos. Porém, ainda é necessário avançar em formas de avaliar os dispositivos para analisar o funcionamento das funções adaptativas e determinar se as ações correspondentes estão atingindo os resultados esperados.

REFERÊNCIAS

- [1] ALPAYDIN, E.: "Introduction to Machine Learning". MIT Press, 2ª Edição, 2010. ISBN-10: 0-262-01243-X
- [2] BECK, J.: "Combinatorial Games: Tic-Tac-Toe Theory", Cambridge University Press, 2008.
- [3] DIAS, J. B.; SOUZA, K. P. de; PISTORI, H.: Conjunto de Treinamento para Algoritmos de Reconhecimento de LIBRAS. II Workshop de Visão Computacional, São Carlos, Outubro 16-18, 2006.
- [4] GANZELI, H. S.; BOTTESINI, J. G.; PAZ, L. O.; RIBEIRO, M. F. S.: Skan-Skin Scanner: software para o reconhecimento de câncer de pele utilizando técnicas adaptativas. Memórias do WTA 2010 – IV Workshop de Tecnologia Adaptativa, São Paulo, 2010.
- [5] HIRAKAWA, A. R.; SARAIVA, A. M.; CUGNASCA, C. E.: Autômatos Adaptativos Aplicados em Automação e Robótica. Revista IEEE América Latina. Vol. 5, Num. 7, ISSN: 1548-0992, Novembro 2007. (p. 539-543)
- [6] MENEZES, P. B.: "Linguagens Formais e Autômatos". 6ª Edição. Bookman, 2011. ISBN: 9788577807659
- [7] MICHALSKI, R. S.: "A theory and methodology of inductive learning". In Machine Learning L. R. S. Michalski, J. Carbonelli, and T. Mitchell, eds. Palo Alto, CA: Tioga Publishing, 1983.
- [8] MITCHELL, T. M.: "Machine Learning". 1ª Edição. McGraw-Hill, 1997. ISBN: 0070428077.
- [9] MONARD, M. C.; BARANAUKAS, J. A.: "Aplicações de Inteligência Artificial: Uma Visão Geral". São Carlos: Instituto de Ciências Matemáticas e de Computação de São Carlos, 2000.
- [10] NETO, J. J.: "Contribuição à metodologia de construção de compiladores". São Paulo, 272p. Tese (Livro-Docência), Escola Politécnica, Universidade de São Paulo, 1993.

- [11] NETO, J. J.: "Solving complex problems with Adaptive Automata". Lecture Notes in Computer Science. S. Yu, A. Paun (Eds.): Implementation and Application of Automata, CIAA 2000, Vol.2088, London, Canada, Springer-Verlag, pp.340, 2000.
- [12] NETO, J. J. "Adaptive Rule-Driven Devices – General Formulation and Case Study". Revista de Engenharia de Computação e Sistemas Digitais, São Paulo, v.1, n.1, 2001.
- [13] NETO, J. J. "Adaptive Technology and Its Applications". Encyclopedia of Artificial Intelligence. New York, v.1, p.: 37-44, 2009.
- [14] NETO, J. J., IWAI, M. K.: "Adaptive Automata for Syntax Learning". In Anais da XXIV Conferência Latino-americana de Informática - CLEI 98, pg. 135–149, Quito, Equador, 1998.
- [15] PARIENTE, C. B.; NETO, J. J.; SANTANA, F. S.: Towards an Adaptive Implementation of Genetic Algorithms. I Taller Latinoamericano de Informática para la Biodiversidad (INBI) - CLEI 2007, San José, Costa Rica, 9-12 Octubre, 2007.
- [16] PISTORI, H. "Tecnologia Adaptativa em Engenharia de Computação: estado da arte e aplicações". Tese (Doutorado), Escola Politécnica da USP, 2003.
- [17] PISTORI, H.; NETO, J. J.: AdapTree: Proposta de um Algoritmo para Indução de Árvores de Decisão Baseado em Técnicas Adaptativas. Anais Conferência Latino Americana de Informática - CLEI 2002. Montevideu, Uruguai, Novembro, 2002.
- [18] PISTORI, H.; NETO, J. J.: "Decision Tree Induction using Adaptive FSA". CLEI Electronic Journal. Volume 6, Number 1, 2003.
- [19] PISTORI, H.; NETO, J. J.: "An Experiment on Handshape Sign Recognition using Adaptive Technology: Preliminary Results". XVII Brazilian Symposium on Artificial Intelligence - SBIA 04. São Luis, September 29 - October 1, 2004
- [20] PRESSMAN, R. S.: "Engenharia de Software". 6ª Edição. Rio de Janeiro: McGraw-Hill, 2006. ISBN: 8586804576.
- [21] QUINLAN, J. R.: "Induction of decision trees". Machine Learning, 1, 81-106, 1986.
- [22] RAJANI, N.F. DAR, G. BISWAS, R. RAMESHA, C.K.: Solution to the Tic-Tac-Toe Problem Using Hamming Distance Approach in a Neural Network. In: Second International Conference on Intelligent Systems, Modelling and Simulation - ISMS 2011, Cambodia, 2011.
- [23] ROCHA, R.L.; NETO, J.J.: "Autômato adaptativo, limites e complexidade em comparação com máquina de Turing". In: Proceedings of the second Congress of Logic Applied to Technology - LAPTEC 2000. São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, p. 33-48, 2000.
- [24] STANGE, R. L.; NETO, J. J.: Aprendizagem Incremental Usando Tabelas De Decisão Adaptativas. Memórias do WTA 2011 – Workshop De Tecnologia Adaptativa, EPUSP, São Paulo, 2011.
- [25] STANGE, R. L.; GIANNINI, T. C.; SANTANA, F. S.; JOSE, J.; MAURO SARAIVA, A.: Evaluation of Adaptive Genetic Algorithm to Environmental Modeling of Peponapis and Cucurbita. Revista IEEE América Latina, v. 9, p. 171-177, 2011.
- [26] TCHEMRA, A. H.: Aplicação da Tecnologia Adaptativa em Sistemas de Tomada de Decisão. I WTA – Workshop sobre Tecnologia Adaptativa, São Paulo, 2007.
- [27] TCHEMRA, A. H.: "Tabela de Decisão Adaptativa na Tomada de Decisão Multicritério". Tese (Doutorado), EPUSP, São Paulo, 2009.
- [28] TCHEMRA, A. H.; CAMARGO, R.: Descoberta de padrões em bases de dados utilizando Técnicas Adaptativas. III WTA – Workshop sobre Tecnologia Adaptativa, São Paulo, 2009.
- [29] WIDYANTORO, D. H.; VEMBRINA, Y. G.: Learning to play tic-tac-toe. In: International Conference on Electrical Engineering and Informatics, 2009.
- [30] YAU Y.J., TEO J. AND ANTHONY P.: Evolution and Co-Evolution in Cognitive Neural Agents Synthesis for Tic-Tac-Toe. IEEE Symposium on Computational Intelligence and Games (CIG 2007), pages 304-311, Hawaii, USA, 2007.