

1º Trabalho prático de Inteligência Artificial

Avaliação Experimental de Agentes Inteligentes

Silvio Eduardo Sales de Britto Ribeiro
Lamartine Cabral de Oliveira Neto

¹Centro de Ciência e Tecnologia – Universidade Estadual do Ceará

Fortaleza, 12 de Maio de 2021

lamartine.cabral@aluno.uece.br, silvio.eduardo@aluno.uece.br

Resumo. Neste trabalho abordaremos conceitos fundamentais para o entendimento de Inteligência Artificial como subárea da Ciência da Computação, através do estudo de Agentes Inteligentes, sua forma mais básica, seus ambientes, medidas de desempenho, até modelos intermediários utilitários, dentre outros. Exemplos práticos e implementações na plataforma NetLogo, assim como medidas de desempenho e inferências estatísticas auxiliarão no entendimento dos conceitos e suas aplicações no cotidiano, como é o objetivo principal deste trabalho.

1. Introdução

Um Agente Inteligente é caracterizado por suas ferramentas e tarefas. O que se faz com as entradas percebidas pelos seus sensores e como essas percepções são processadas é o que principalmente diferencia os vários tipos de Agentes. De um simples robô aspirador de pó, passando por algoritmos de recomendação e até carros e drones autônomos, todos têm em comum, no nível mais básico, a Função do Agente, que é o comportamento matemático abstrato do agente, e um Programa Agente que é o sistema concreto onde ele será implementado [Norvig and Russell 2014].

2. Agentes Inteligentes

De forma geral, um Agente é qualquer coisa que possa perceber o seu devido ambiente através de sensores e que atue nesse mesmo ambiente através de atuadores. [Norvig and Russell 2014, p. 34]. Robôs e algoritmos, são agentes artificiais, e também podem ser caracterizados como agentes racionais artificiais, a depender do seu comportamento e do estado do ambiente em que atua.

Um agente racional é aquele que faz a coisa certa, isto é, o que foi projetado a fazer. Em alguns casos é mais simples medir se o agente está agindo como desejado ou não: é fácil perceber algumas medidas de desempenho satisfatórias para um carro autônomo, por exemplo, ou um robô aspirador. Já para outros sistemas é mais complexo, pois não há parâmetros muito claros. O ser humano, por exemplo, pode ser caracterizado como um agente racional, mas como medir se ele está fazendo o que foi projetado a fazer? Para mensurar isso, no caso de agentes racionais artificiais, observa-se o seu desempenho através dos estados observados no ambiente que ele está inserido, e não pelos seus estados. O resultado da atuação do agente no ambiente é que nos diz se ele está agindo corretamente. Não é possível medir a performance do agente através dos seus próprios estados, pois ele pode atribuir a si um desempenho inexistente, onde a percepção dos parâmetros pode ser deturpada, porque é uma estimativa suspeita e trata-se de um agente racional. Seria preciso observar o chão, no caso do aspirador de pó, para estimar o quanto de poeira limpou, o tempo, o gasto de energia e inferir, através dessas e outras informações, o seu real desempenho.

Um ambiente de tarefas (*Task Environment*) é uma especificação das medidas de performance, do ambiente, junto com os atuadores e sensores do agente. Usamos o acrônimo P.E.A.S. (*Performance, Environment, Actuators, Sensors*) como referência a esse ambiente. *Performance* designa, objetivamente, as qualidades desejadas do sistema, considerando um custo-benefício. *Environment* sugere o ambiente de atuação do agente em questão que, por sua vez, possui *Actuators e Sensors* que são mecanismos de reação aos sensores e mecanismos de entrada de dados ou percepção, respectivamente.

Em particular, no caso de um táxi autônomo, o seu ambiente de tarefas é definido por: Medidas de performance como economia de combustível, passageiros satisfeitos, seguros, rotas seguras e otimizadas em ambientes de estrada, urbano, estradas de pedras; sendo controlado por um painel, volante e alarmes que o permite atuar nesse ambiente através das rodas, faróis, câmeras e sensores de calor e movimento. Outro exemplo seria um algoritmo de recomendação que poderia ser mensurado através da confirmação, por parte do cliente, da sua recomendação, da taxa de acerto em recomendações, novas recomendações precisas com menos informações, em ambientes diversos: web, intranet, streaming, onde atua nesse ambiente expondo recomendações como saída de dados, geralmente em uma tela e recebendo requisições dos clientes que usam a plataforma e disponibiliza informação a ser percebida e processada.

Um programa agente implementa uma função do agente, em uma arquitetura de dispositivos e usa os sensores e atuadores dessa arquitetura. A estrutura básica de um programa agente consiste em receber a informação atual como entrada pelos sensores e dar uma resposta através dos atuadores. Diferentemente de uma função agente, que processa toda uma cadeia de percepções. No caso de um Agente de Aprendizagem, este possui um sistema razoavelmente sofisticado de tratamento de entradas envolvendo elementos de aprendizagem que indicam como o elemento de performance irá modificar o sistema, baseado no comportamento do Crítico e nas sugestões do gerador de problemas. Os outros tipos de programas de agentes não são capazes de aprender com o ambiente e ter uma leitura mais fiel, porém conseguem produzir respostas simples, considerar um modelo, estabelecer objetivos e aumentar as probabilidades de sucesso, através da utilidade. É possível ter uma noção da variabilidade de agentes através de suas arquiteturas, que pode ser, por exemplo, uma arquitetura reativa simples, cujo agente define suas ações em tempo real sem o armazenamento de qualquer representação do ambiente e não utiliza racionalidade complexa. Ou, também, uma arquitetura cognitiva, onde o agente cognitivo tem uma representação simbólica do ambiente e suas decisões são tomadas baseados em um processo lógico.

3. Metodologia

3.1

As implementações consistiram no uso do programa já existente e estruturado em NetLogo “meufirst.netlogo” disponibilizado pelo Professor. Este foi atualizado em versões à medida que eram feitas mudanças. Foram implementadas novas *features* como um agente novo mais inteligente, mudança em parâmetros de desempenho, mudança dos relatórios, mudanças no agente já implementado, dentre outras melhorias. Com a mudança de parâmetros de desempenho, foi possível criar medidas mais aplicadas a uma situação de aspirador de pó limpando um ambiente com número de obstáculos fixos (uma casa inteira ou uma sala, por exemplo), e a partir desses dados medir performance.

3.1.1

O novo agente implementado executa uma DFS (Depth-first Search), cuja estrutura auxiliar para a tomada de decisão é uma pilha, que guarda o caminho de volta. O modelo utilizado foi o do robô aspirador. O estado interno do robô inclui a pilha já mencionada e uma matriz para marcar posições já visitadas, a fim de evitar visitas repetidas e garantir o caminho acíclico da busca em profundidade.

No início e quando o robô chega em uma nova posição, ele sorteia uma direção para se mover. Como a busca em profundidade volta pelo mesmo caminho de ida quando não há mais opções, é necessário inserir na pilha o movimento de volta. Lembrando que a direção sorteada deve ser válida. Ou seja, deve levar a uma posição ainda não visitada. Se não houver direções válidas para sortear, então o próximo movimento do robô será voltar para onde ele veio, que será justamente a ação do topo da pilha. Quando não houver mais ações na pilha, significará que o robô já percorreu todas as posições possíveis.

```
to move-robot-dfs|
ask robots [

  if not começou [
    let lado sqrt(Quadrados-totais)
    set visitados matrix:make-constant (lado + 2) (lado + 2) 0
    set começou true
  ]

  set-visit

  let nr try-some-direction
  let indo true
  if nr = -1 [
    ;; entra aqui se nao tem pra onde ir
    if length pilha = 0 [
      ;; entra aqui se nao tem pra onde voltar
      set começou false ;; isso da um restart pra manter o robo em loop
      stop
    ]
    set nr last pilha
    set pilha but-last pilha
    set indo false
  ]
  set heading (90 * nr)

  ifelse (can-move? (one-quad-size))
  [
    forward one-quad-size

    ifelse (any? obstacs-here)[
      ;; Entre aqui se tem um bloco/obstc pra frente
      back one-quad-size
      set-parede nr

      if(perde-pt-obstac)[
        set points points - 1
      ]
    ]
    [
      ;; Entra aqui se pode mover pra frente
      if indo [
        set pilha lput (volta nr) pilha ;; empilha a volta
      ]

      set casas-andad (casas-andad + 1)
      att-energia

      ifelse(limpar-sujeira)[
        set points (points + 50)
        plot-graph true
      ]
      [
        set points (points - 1)
        plot-graph false
      ]
    ]
  ]
  ;; Entra aqui se tem paredes na frente
  set-parede nr
  if(Perde-pt-parede)[
    set points points - 1
  ]
]
```

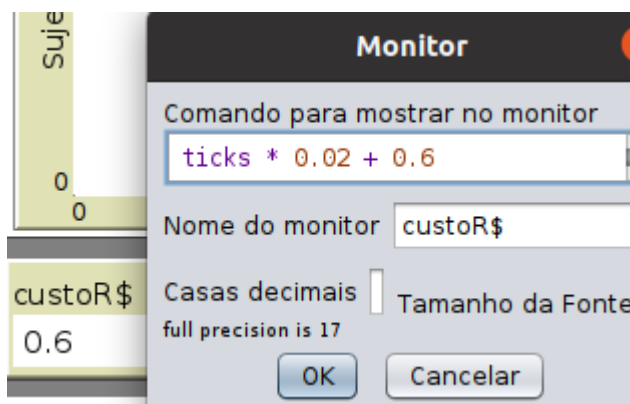
Implementação do agente DFS no Netlogo

3.1.2

Dentre as alterações, adaptamos alguns parâmetros para melhorar o entendimento aplicado ao mundo real do desempenho do agente. Foi adicionado uma medida “custo(R\$)” que é uma variável em função dos ticks, que optamos por igualar a minutos (1 tick = 1 minuto), sendo assim, uma variável de custo em função do tempo de modo que o resultado em reais (R\$) por ambiente completamente limpo deriva do seguinte fator:

$$\text{ticks} \times 0.02 + 0.6 ,$$

sendo 0.02 a nossa atribuição direta de R\$ 0,02 para cada minuto do aspirador ligado adicionado de 0.06 que é o resultado de $15 \times 2 \times 0.02$, onde 15 é o número de quadrados sujos onde foi definido um fator de 2x mais gasto energético para o aspirador aspirar locais que estejam de fato sujos vezes o custo de um minuto ligado (R\$ 0,02), resultando em 0,6.



Fator de cálculo do custo no monitor do NetLogo

Foi alterada a medida de “energia” para “bateria”, onde o agente está munido de 100% de carga da sua bateria antes de toda rodada de teste, e a medida que o agente vai limpando o ambiente, a bateria vai diminuindo. Mantivemos o custo-benefício como uma razão da sujeira limpa pelo número de casas percorridas. Os devidos relatórios externos também foram adaptados aos novos parâmetros.

3.1.3

Percebemos que os agentes “Inteligente” e o “Super-Inteligente” tem o comportamento muito parecido, praticamente igual. Então optamos por usar somente o último como sendo “Inteligente”, sobrepondo o código do “Super-inteligente” no do “Inteligente”, sobrando três agentes de tipos diferentes que servem muito bem para fazer as comparações: “burro”, “inteligente” e “dfs”.

3.1.4

Para medir o desempenho, foi coletada uma população de **50 rodadas de testes** para cada agente em um ambiente de 49 quadros ou 7m², com 10 obstáculos e 15 pontos de sujeira. À medida que as rodadas de teste vão ocorrendo, a coleta dos resultados acontece em relatórios externos a cada rodada. A partir da análise desses dados podemos observar algumas medidas dos parâmetros e compará-las entre si. Média de custo-benefício, de gasto em reais por ambiente limpo, de bateria, tempo, pontuação e metros percorridos para a cada limpeza. Partindo desses dados, podemos afirmar sobre a performance de cada agente e compará-los.

3.1.5

Com o total de rodadas para cada agente em mãos, prosseguimos com o cálculo de uma amostragem que seja fiel ao custo total das rodadas de teste do agente DFS, com um erro amostral de 15%, e admitindo um nível de confiança de 95%. Para isso, usaremos a fórmula:

$$n = \frac{Z_c^2 \times \delta^2}{e^2}$$

onde Z_c é o ponto crítico da curva normal, δ é o desvio padrão populacional e e o erro amostral admitido.

3.1.6

Possuindo uma amostra menor, obtemos sua média e desvio padrão para inferir a média populacional (que já temos) e observar se são compatíveis. Calculamos um intervalo de confiança de 95% para a média populacional do custo em R\$ do aspirador DFS, esse será o parâmetro usado.

Como se trata de uma amostra pequena, é usado a distribuição t de student, de modo que primeiro é preciso encontrar o Erro máximo com a seguinte fórmula:

$$E = tc \times \frac{s}{\sqrt{n}}$$

onde tc é o valor crítico para determinado nível de confiança e grau de liberdade. s é o desvio padrão amostral dos custos em R\$ do aspirador, e n é o tamanho da amostra.

Após descoberto o Erro, estimamos que a média populacional dos custo em R\$ do agente está no intervalo:

$$\bar{x} - E < \mu < \bar{x} + E$$

onde estimamos com 95% de confiança que a nossa média populacional está dentro dessa faixa de valores.

4. Resultados

4.1

Como resultado dos experimentos, foram obtidas muitas informações relevantes sobre o comportamento dos agentes, principalmente do agente de busca, que os ganhos em tempo, custo, bateria são muito significativos. Na tabela a seguir podemos ter uma noção dos ganhos de performance do agente DFS comparado aos outros.

Rodadas: 50

DFS	Custo-benef	Pontos:	Percorrido:	Bateria:	Tempo(min):	Custo(R\$):
Moda	0,46	674	59	99,41%	74	2,08
Mediana	0,64	685	55	99,45%	79	2,18
Média pop	70,2%	682,824	56,62m	99,43%	81,176	2,22
Var pop	0,052	201,871	107,371	107,371	201,871	0,08
Desvp pop	0,229	14,208	10,362	10,362	14,208	0,28
						T: 113,40

Rodadas: 50

INTELIGEN	Custo-benef:	Pontos:	Percorrido:	Bateria:	Tempo(min):	Custo(R\$):
Moda	0,04	431	239	9666	238	5,36
Mediana	0,07	350	415	9585	414	8,88
Média pop	7,8%	256,059	508,94m	94,91%	507,941	10,76
Var pop	0,002	85701,781	85701,781	85701,781	85701,781	34,28
Desvp pop	0,045	292,749	292,749	292,749	292,749	5,85
						T: 548,70

Rodadas: 50

BURRO	Custo-benef:	Pontos:	Percorrido:	Bateria:	Tempo(min):	Custo(R\$):
Moda	0,1	220	245	9689	141	3,42
Mediana	0,090	235	344,5	9655,5	529	11,18
Média pop	10,3%	48,32	499,04m	95%	715,62	14,91
Var pop	0,006	404682,578	200737,918	200737,918	404719,596	161,89
Desvp pop	0,079	636,147	448,038	448,038	636,176	12,72
Custo-total						T: 745,62

Tabela definitiva de desempenho dos agentes

Aqui podemos observar com mais clareza os resultados. Chama atenção o custo benefício médio do agente não inteligente ser maior que o do inteligente, mas como é possível observar na primeira tabela, o agente DFS tem quase 7x melhor custo-benefício que os demais.

Podemos perceber que, em média, para limpar completamente uma sala de 7m² com 15 obstáculos, o agente Inteligente percorreria 508,9 metros dentro do ambiente, gastaria o equivalente a 8 horas e 27 minutos, quase $\frac{1}{3}$ do dia, e gastaria R\$ 10,76 somente nessa limpeza.

Supondo que uma pessoa deixe o aspirador ligado 4 horas por dia, para tentar limpar só metade da sujeira do ambiente, em um mês ela gastou R\$ 152,50 só com o aspirador, e ainda não limpou toda a sujeira.

Já se essa pessoa usasse o aspirador com DFS, em um ambiente de 7m² ele percorre em média 56 m, em 1 hora e 21 minutos e custa somente R\$ 2,22 limpando completamente o ambiente. Em um mês ele custaria R\$ 66,60, menos da metade do aspirador Inteligente (que limpa só metade), limparia toda a sujeira e em quase quatro vezes menos tempo. Isso mostra o poder da busca em profundidade.

4.2

Para calcular a amostragem, usaremos a fórmula já apresentada e parametrizando ela com o desvio padrão populacional do custo, presente na tabela, o erro amostral tolerável de 15% e um nível de confiança de 95%, segue:

$$Z_c(0.95)=1,96$$

$$e=0,15$$

$$\delta=0.28$$

Assim, temos que:

$$n = \frac{Z_c^2 \times \delta^2}{e^2} = \frac{1,96^2 \times 0,28^2}{0,15^2} = 13,312 \text{ amostras}$$

Ou seja, para garantirmos 95% de confiança e margem de erro de 15%, a nova amostra deve ter por volta de 14 elementos. Isso significa dizer que podemos fazer uma média amostral com 14 custos aleatórios do agente com busca que, com probabilidade de 95%, a média real das 50 rodadas (R\$ 2,20) coincide com a média amostral, dentro de um intervalo de 15% de tolerância para mais e para menos.

4.3

Agora que foi obtido um tamanho de amostra, escolhemos uma porção aleatória de tamanho 14 dos custos por rodada de teste do aspirador DFS, como indica a coluna L na tabela:

	A	B	C	D	E	F	G	H	I	J	K	L
141	Custo-benef:	0,444	Pontos:	664	Percorrido:	72	Bateria:	9928	Tempo(min):	100	Custo(R\$):	2,6
142	Custo-benef:	0,554	Pontos:	674	Percorrido:	60	Bateria:	9940	Tempo(min):	90	Custo(R\$):	2,4
143	Custo-benef:	0,526	Pontos:	684	Percorrido:	59	Bateria:	9941	Tempo(min):	80	Custo(R\$):	2,2
144	Custo-benef:	0,975	Pontos:	696	Percorrido:	49	Bateria:	9951	Tempo(min):	68	Custo(R\$):	1,96
145	Custo-benef:	0,592	Pontos:	678	Percorrido:	58	Bateria:	9942	Tempo(min):	86	Custo(R\$):	2,32
146	Custo-benef:	0,705	Pontos:	692	Percorrido:	50	Bateria:	9950	Tempo(min):	72	Custo(R\$):	2,04
147	Custo-benef:	0,62	Pontos:	679	Percorrido:	59	Bateria:	9941	Tempo(min):	85	Custo(R\$):	2,3
148	Custo-benef:	1,448	Pontos:	714	Percorrido:	34	Bateria:	9966	Tempo(min):	50	Custo(R\$):	1,6
149	Custo-benef:	0,437	Pontos:	660	Percorrido:	71	Bateria:	9929	Tempo(min):	104	Custo(R\$):	2,68
150	Custo-benef:	0,498	Pontos:	667	Percorrido:	70	Bateria:	9930	Tempo(min):	97	Custo(R\$):	2,54
151	Custo-benef:	0,538	Pontos:	671	Percorrido:	63	Bateria:	9937	Tempo(min):	93	Custo(R\$):	2,46
152	Custo-benef:	0,612	Pontos:	685	Percorrido:	55	Bateria:	9945	Tempo(min):	79	Custo(R\$):	2,18
153	Custo-benef:	0,543	Pontos:	674	Percorrido:	64	Bateria:	9936	Tempo(min):	90	Custo(R\$):	2,4
154	Custo-benef:	0,452	Pontos:	660	Percorrido:	71	Bateria:	9929	Tempo(min):	104	Custo(R\$):	2,68
155	Custo-benef:	0,769	Pontos:	680	Percorrido:	56	Bateria:	9944	Tempo(min):	84	Custo(R\$):	2,28
156	Custo-benef:	1,245	Pontos:	719	Percorrido:	32	Bateria:	9968	Tempo(min):	45	Custo(R\$):	1,5
157	Custo-benef:	1,071	Pontos:	710	Percorrido:	35	Bateria:	9965	Tempo(min):	54	Custo(R\$):	1,68
158	Custo-benef:	0,787	Pontos:	689	Percorrido:	52	Bateria:	9948	Tempo(min):	75	Custo(R\$):	2,1
159	Custo-benef:	0,886	Pontos:	698	Percorrido:	47	Bateria:	9953	Tempo(min):	66	Custo(R\$):	1,92
160	Custo-benef:	0,419	Pontos:	663	Percorrido:	73	Bateria:	9927	Tempo(min):	101	Custo(R\$):	2,62
161	Custo-benef:	0,636	Pontos:	689	Percorrido:	55	Bateria:	9945	Tempo(min):	75	Custo(R\$):	2,1
162	Custo-benef:	0,538	Pontos:	675	Percorrido:	63	Bateria:	9937	Tempo(min):	89	Custo(R\$):	2,38
163	Custo-benef:	0,425	Pontos:	658	Percorrido:	75	Bateria:	9925	Tempo(min):	106	Custo(R\$):	2,72
164	Custo-benef:	0,689	Pontos:	685	Percorrido:	54	Bateria:	9946	Tempo(min):	79	Custo(R\$):	2,18
165	Custo-benef:	0,935	Pontos:	687	Percorrido:	51	Bateria:	9949	Tempo(min):	77	Custo(R\$):	2,14
166	Custo-benef:	0,47	Pontos:	673	Percorrido:	68	Bateria:	9932	Tempo(min):	91	Custo(R\$):	2,42
167	Custo-benef:	0,823	Pontos:	692	Percorrido:	51	Bateria:	9949	Tempo(min):	72	Custo(R\$):	2,04
168	Custo-benef:	0,917	Pontos:	697	Percorrido:	48	Bateria:	9952	Tempo(min):	67	Custo(R\$):	1,94
169	Custo-benef:	1,01	Pontos:	688	Percorrido:	52	Bateria:	9948	Tempo(min):	76	Custo(R\$):	2,12
170	Custo-benef:	0,458	Pontos:	663	Percorrido:	71	Bateria:	9929	Tempo(min):	101	Custo(R\$):	2,62
171			D F S									
172	Moda	0,463		674		59		9941		74		2,08
173	Mediana	0,636		685		55		9945		79		2,18
174	Média pop	0,702		682,824		56,627		9943,373		81,176		2,22
175	Var pop	0,052		201,871		107,371		107,371		201,871		0,08
176	Desvp pop	0,229		14,208		10,362		10,362		14,208		0,28

Escolhendo ao acaso 14 valores, temos:

2.06, 2.26, 2.12, 2.60, 2.40, 2.20, 2.04, 2.30, 2.68, 2.54, 2.18, 2.68, 2.28, 2.10

tirando a média e desvio padrão amostrais desses valores, temos $\bar{x}=2,32$ e desvio padrão $s=0,226$

Já é possível observar que a média amostral (2,32) ficou muito próxima da média real da população (2,20), mas isso será confirmado com o cálculo do intervalo de confiança.

Para observar se serão compatíveis as médias real e amostral, procederemos utilizando a distribuição t de student para o intervalo de confiança. Primeiro é preciso calcular a margem de erro máxima. Temos:

$$n=14$$

$$tc(0.95,13)=2,160$$

$$s=0,226$$

Assim, temos que

$$E = tc \times \frac{s}{\sqrt{n}}$$

Agora é só incluir o intervalo de confiança para a média populacional (μ), como segue

$$\bar{x} - E < \mu < \bar{x} + E$$

$$2.32 - 0.13 < \mu < 2.32 + 0.13$$

$$2.19 < \mu < 2.45$$

Assim, é possível garantir com 95% de confiança, considerando a margem de 0,13 em torno da média, que a média amostral de custo para limpar um quarto com DFS coincide com a média real, o que é verdade, pois $\mu=2,20$ está no intervalo de 2,19 até 2,45. Portanto, a nossa amostragem inicial está totalmente coerente e dentro do intervalo de confiança proposto.

Referências

Norvig, P. and Russell, S. (2014). **Artificial Intelligence: a modern approach**.

MARIANO, Mirtes V., LAURICELLA, Christiane M., FRUGOLI, Alexandre D. **Estatística Indutiva: teoria, exercícios resolvidos, exercícios propostos e tarefas**