

Detectando Padrões Comportamentais de Agentes Através de Árvores de Decisão

Josimara de Á. Silveira, Thiago L. Paes, Leonardo R. Emmendorfer e Diana F. Adamatti

Centro de Ciências Computacionais – C3
Universidade Federal do Rio Grande (FURG)
Av. Itália, km 08, Carreiros
96201-090 – Rio Grande -RS - BRASIL

josi_silver@hotmail.com, thiagopaes@gmail.com, leonardo.emmendorfer@gmail.com, dianaadamatti@furg.br

Abstract

This paper provides a description of an alternative for detecting the behavior of multiagent systems. Decision trees are learned from behavioral data collected from simulations of a predator-prey ecosystem. The behavior inferred from the decision trees is compared to the programmed behavior of the agents, and the differences between both are analyzed.

Keywords: Artificial Intelligence, Decision Trees, Multiagent Systems, Predator-Prey.

1. INTRODUÇÃO

Técnicas de aprendizado artificial, ou aprendizado de máquina, são adotadas em diversas situações que envolvem a necessidade de adaptação, mudança de comportamento, extração de conhecimento, detecção de padrões, entre outras. O aprendizado de máquina está relacionado com a obtenção de melhora de desempenho automática, através de experiência. No estudo em questão, utilizamos árvores de decisão. A aprendizagem de árvores de decisão é um método para aproximar funções alvo de valor discreto, onde a função aprendida é representada através de uma estrutura de condicionamento em árvore [1][2]. Árvores de decisão são treinadas por meio de conjuntos de dados que simbolizam eventos já ocorridos e já classificados. Existem diversos algoritmos para a geração de árvores de decisão. Neste

trabalho, utilizamos o J48, uma variante do C 4.5. [3]

O segundo conceito importante de ser enfatizado antes de prosseguirmos é o de Sistemas Multiagentes (SMA). Os Sistemas Multiagentes (SMA) formam uma subárea da Inteligência Artificial Distribuída e concentram-se no estudo de agentes autônomos em um universo multiagente. Para os SMA, o termo autônomo designa o fato de que os agentes têm uma existência própria, independente da existência de outros agentes. Usualmente, cada agente possui um conjunto de capacidades comportamentais que definem sua competência, um conjunto de objetivos, e a autonomia necessária para utilizar suas capacidades comportamentais a fim de alcançar seus objetivos. Um agente é uma entidade computacional com um comportamento autônomo que lhe permite decidir suas próprias ações [4].

O objetivo do estudo em questão é reconstruir regras que regem dinâmicas de agentes a partir do comportamento observado dos mesmos. Inicialmente estamos trabalhando com um sistema multiagente baseado em regras comportamentais bastante simples.

Uma questão muito freqüente em aprendizado artificial diz respeito aos limites associados a cada uma das técnicas. Em outras palavras, qual o escopo de aplicação de cada técnica de aprendizado artificial? Com isso em mente, surge a idéia de conduzir experimentos de modo a respeitar esta questão e procurar descobrir até que ponto as árvores de decisão são suficientes para detectar padrões de comportamentos de agentes e gerar, para diferentes classes de indivíduo, descrições aproximadas da sua

personalidade.

O trabalho tomou como base um modelo multiagente do tipo presa-predador[5][8]. Foram escolhidos empiricamente conjuntos de parâmetros para os quais o modelo é estável, representando a manutenção das duas espécies concorrentes ao longo do tempo.

2. METODOLOGIA

Foi utilizada uma simulação multiagentes de um ecossistema presa-predador, no caso lobo-ovelha. Tal simulação foi implementada a partir de um modelo já pronto disponível na versão 4.04 do NetLogo[6]. Foi criada uma versão estendida da original, na qual foram adicionados comportamentos de fuga e busca entre os agentes, que antes se moviam aleatoriamente.

A movimentação e o alcance local dos indivíduos utilizam o conceito de vizinhança de Moore, a qual consiste das células mais próximas. Existe a possibilidade de movimentação para 8 direções distintas, correspondentes às direções Norte (N), Sul(S), Leste(L), Oeste (O), Noroeste (NO), Nordeste (NE), Sudoeste (SO), Sudeste(SE).

O modelo é formado por 3 tipos de agentes: Grama, Lobos e Ovelhas. A grama morre sempre que uma ovelha a atinge. Quando isso acontece, um contador é incrementado com um valor randômico relacionado a um valor de *grass-regrowth-time* estipulado na simulação. Esse valor é decrementado até que zere e a grama nasce novamente naquele lugar. Cada vez que uma ovelha come grama a energia da ovelha aumenta.

Os lobos interagem com as ovelhas em uma relação de predação, caçando-as (Figura 1). Quando o lobo caça uma ovelha sua energia aumenta. Para isso, o lobo percebe as ovelhas (somente as que estão a sua frente). A visão do lobo é em forma de cone com 200 graus de abertura.

```

ifelse (ovelhas_na_volta >= 1)
[
  Escolhe_uma_das_ovelhas_com_200_graus_de_visao()

  if alvo != ninguem
  [
    Mata_ovelha()
    Aumenta_energia_lobo()
  ]
  Diminui_energia_lobo
  Atualiza_coordenadas()
]

```

Figura 1: Pseudocódigo - Comportamento dos lobos

Já as ovelhas sobrevivem fugindo dos lobos e

alimentando-se de grama. A fuga das ovelhas está relacionada com a percepção ou não de lobos à sua volta. O método utilizado para fugir baseia-se em cada ovelha perceber a presença ou não de algum lobo e também a posição destes (Figura 2). Caso não existam lobos à sua volta, não há fuga e a ovelha movimenta-se aleatoriamente. Caso exista lobo, o comportamento da ovelha é delineado conforme regras de precedência. Quando em perigo, a ovelha olha primeiro para o noroeste (NO) e, caso não exista lobo, dirige-se para lá. Se existe lobo em NO, ela olha para o norte (N) e dirige-se até lá caso não haja predador, e assim por diante seguindo a ordem de procura: NO, N, NE, O, E, SO, S, SE. Tanto lobos quanto ovelhas perdem energia ao se mover.

```

ifelse (lobos_na_vizinhanca >1) [
  ifelse (lobo_em_NO) = 0 [
    vai_para_NO
  ] ;
  ifelse (lobo_em_N) = 0 [
    vai_para_N
  ] ;
  ifelse (lobo_em_NE) = 0 [
    vai_para_NE
  ] ;
  ifelse (lobo_em_O) = 0 [
    vai_para_O
  ] ;
  ifelse (lobo_em_E) = 0 [
    vai_para_E
  ] ;
  ifelse (lobo_em_SO) = 0 [
    vai_para_SO
  ] ;
  ifelse (lobo_em_S) = 0 [
    vai_para_S
  ] ;
  ifelse (lobo_em_SE) = 0 [
    vai_para_SE
  ] ;
]
Anda_randomico() // sem lobo na
vizinhanca
]
Perde_energia() // se movimentou,
perdeu energia
Atualiza_coordenadas()

```

Figura 2: Pseudocódigo - Como as ovelhas fogem?

Os dados de comportamento gerados pelo ambiente NetLogo são os seguintes, para os agentes do tipo “lobo”:

- a energia de cada indivíduo,
- um vetor de nove posições que representa a existência ou não de ovelhas nos seus oito vizinhos ou junto dele,

- a soma do número de ovelhas na volta,
- um booleano representador do fato do lobo estar vivo ou não (um lobo não estará vivo quando sua energia for igual a zero) e
- um booleano representador do fato do lobo ter matado ou não uma ovelha.

O registro dos indivíduos do tipo ovelha contém:

- um inteiro representando a energia de cada indivíduo,
- um booleano representador do fato do indivíduo ter comido grama ou não,
- um vetor de oito posições que representa a presença ou não de lobos em cada uma das direções da vizinhança,
- um inteiro contendo a soma do número de lobos na vizinhança,
- um booleano representador do fato do indivíduo estar morrendo (quando sua energia chega a zero) ou vivendo e,
- que movimento o indivíduo realizou (são 9 valores possíveis, os oito espaços a sua volta ou um movimento aleatório caso não haja um lobo na vizinhança).

Os registros foram tratados para servir de entrada para o software WEKA [7], versão 3.7, o qual utiliza uma coleção de algoritmos de aprendizado de máquina para realizar mineração de dados. Neste software os registros dos agentes foram classificados utilizando o algoritmo de árvore de decisão J48, o mais popular deste software, o qual é uma variante do algoritmo C4.5 [3].

3. RESULTADOS

3.1 COMPORTAMENTO DOS AGENTES

A Figura 3 apresenta a árvore de decisão que foi aprendida utilizando dados comportamentais dos agentes do tipo ovelha, a partir de simulações.

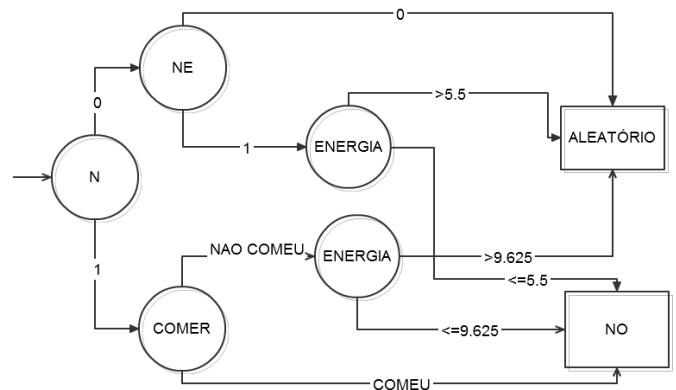


Figura 3: Árvore de decisão dos indivíduos do tipo ovelha

Nesta árvore percebe-se que os movimentos realizados pelos indivíduos do tipo ovelha foram, preferencialmente, o movimento aleatório e o movimento para NO.

De acordo com a árvore de decisão, a ovelha irá se mover aleatoriamente em 3 ocasiões:

- quando não há lobo a norte ou a nordeste da posição atual da ovelha;
- quando não existe lobo a norte, mas há a nordeste e a energia da ovelha é maior que 5,5;
- ou ainda, caso exista lobo a norte, a ovelha não comeu grama e a energia dela é maior ou igual a 9,625.

Também existem três comportamentos detectados pela árvore para que a decisão da ovelha seja mover para NO:

- quando há lobo a norte e a ovelha comeu grama;
- quando há lobo a norte, a ovelha não comeu, mas tem energia maior que 9,625;
- ou quando não há lobo a norte, mas há a nordeste e a energia é menor ou igual a 5,5.

A Figura 4 apresenta a árvore de classificação indivíduos do tipo lobo.

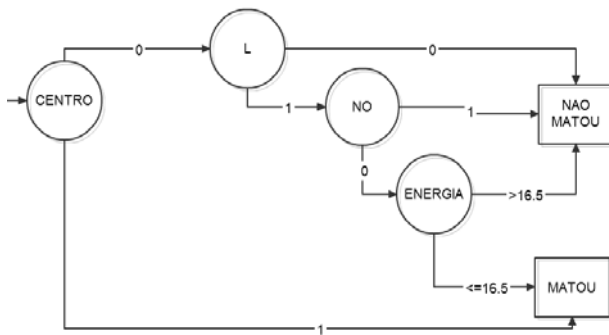


Figura 4: Árvore de decisão dos indivíduos do tipo lobo

Nesta Figura nota-se quando o um indivíduo do tipo lobo consegue matar um do tipo ovelha. Há duas formas que o lobo consegue matar a ovelha:

- quando há uma ovelha no centro, ou seja, ocupando a mesma posição que a sua;
- ou quando não há ovelha no centro, mas há uma a leste, nenhuma a noroeste e sua energia é menor ou igual a 16,5.

E há três formas onde não consegue matar a ovelha:

- quando não há ovelha no centro ou a leste;
- quando não há ovelha no centro e há a leste e a noroeste;
- ou ainda, quando não há ovelha no centro, mas há uma a leste, nenhuma a noroeste e sua energia é maior que 16,5.

3.2 VALIDAÇÕES DOS DADOS EXTRAÍDOS

Os dados extraídos tanto sobre as ovelhas quanto sobre os lobos mostraram-se altamente competentes em representar o modelo presa-predador em questão, confirme ilustrado na Tabela 1.

tipo	instâncias classificadas corretamente	
	Total	%
ovelha	941	94,19%
lobo	949	94,90%

Tabela 1: Eficiência das árvores de decisão como classificadores

4. DISCUSSÃO E CONCLUSÃO

A classificação dos dados no WEKA mostrou que as árvores de decisão tiveram poder suficiente para capturar e explicitar comportamentos dos agentes que estavam implícitos no código.

Como resultado, as técnicas utilizadas mostraram que, para o modelo em questão, o comportamento efetivo dos agentes (detectado pelas árvores) pode ser representado por um conjunto pequeno de variáveis se comparado com o comportamento programado (explícito no código) e o erro é baixo (5,1% e 5,8% para lobos e ovelhas, respectivamente). Para ambos os agentes, tendo em vista as árvores geradas, 4 é o número máximo de variáveis que precisam ser analisadas para sabermos qual a ação de cada agente. Em contrapartida, ao olhar o código referente à simulação, nos depararemos com um número consideravelmente maior de variáveis.

Como trabalhos futuros, pretende-se utilizar a mesma abordagem para ambientes complexos onde os agentes são reais (humanos, por exemplo) a partir de dados de observação do comportamento destes agentes. Existem vários possíveis cenários que podem ser observados, que vão de controle de tráfego e personalidade de condutores até flutuações de mercados de investimento, como bolsas de valores.

REFERÊNCIAS

- [1] R. Duda., P. Hart, D. Stork. PatternClassification2ed. Willey Interscience, Capítulo 8, 2002
- [2] T. Mitchell Machine Learning. WCB McGraw-Hill, Capítulo 3, 1997.
- [3] JR. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
- [4] L. O. Alvares, J. Sichman. Introdução aos Sistemas Multiagentes. In: Jornada de Atualização em Informática, 16.; Congresso da SBC, 17., 1997, Brasília. Anais... Brasília: SBC, p.1-38, 1997
- [5] F. Hoppensteadt, "Predator-prey model", Scholarpedia, 1(10), 1563, 2006
- [6] U. Wilensky. NetLogo Wolf Shee Predation model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. 1997
- [7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten . The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1, 2009
- [8] A. J. Lotka, Lotka on population study, ecology, and evolution. Population and Development Review, 15(3), 539-550, 1989