

Atividade 7: Small Grid World Problem

Small Grid World é um problema que pode ser solucionado com aprendizagem por reforço. Este trabalho apresenta uma implementação do algoritmo de política de iteração em linguagem Java 1.8, para a disciplina de tópicos especiais em aprendizagem.

Introdução

Este artigo mostra como a aplicação que implementa aprendizado por reforço por meio da política de iteração foi codificada. Para isto, uma breve explicação é feita sobre cada método desenvolvido, que compõe o exercício. O objetivo do trabalho é implementar um modelo simples que atenda os requisitos da disciplina.

Métodos

O aprendizado de reforço é o problema de conseguir que um agente atue no mundo para maximizar suas recompensas. Por exemplo, considere ensinar um cão a um novo truque: você não pode dizer o que fazer, mas você pode recompensá-lo ou puni-lo se ele fizer a coisa certa ou errada.

Com isso, deve descobrir o que fez que conseguiu obter a recompensa ou punição, que é conhecido como o problema da atribuição de crédito. Podemos usar um método semelhante para treinar computadores para fazer muitas tarefas, utilizando o mesmo princípio.

Podemos formalizar o problema de Aprendizado por Reforço da seguinte forma. O

ambiente é modelado como uma máquina estocástica de estados finitos com entradas (ações enviadas pelo agente) e saídas (observações e recompensas enviadas ao agente):

- Função de transição de estado $P(X(t) | X(t-1), A(t))$
- Função de observação (saída) $P(Y(t) | X(t), A(t))$
- Função de recompensa $E(R(t) | X(t), A(t))$

O objetivo do agente é encontrar uma política e função de atualização do estado de modo a maximizar a soma esperada de recompensas com desconto.

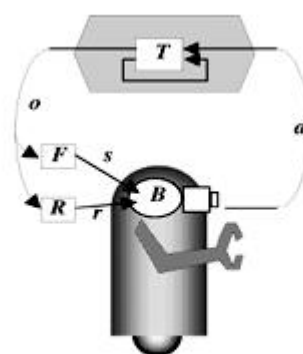


Figura 1: Modelo padrão de Aprendizado por Reforço.

MDP

Um processo de decisão de Markov (MDP) é exatamente como uma Cadeia de Markov,

exceto que a matriz de transição depende da ação tomada pelo tomador de decisão (agente) em cada etapa do tempo. O agente recebe uma recompensa, que depende da ação e do estado. O objetivo é encontrar uma função, chamada política, que especifica qual a ação a seguir em cada estado, de modo a maximizar alguma função (por exemplo, a soma descontada média ou esperada) da seqüência de recompensas. Pode-se formalizar isso em termos da equação de Bellman, que pode ser resolvida iterativamente usando a iteração da política. O único ponto fixo desta equação é a política ótima.

Desenvolvimento

Os métodos foram implementados na linguagem de programação Java. Diferentes métodos foram criados que em conjunto implementam política de iteração.

Policy Iteration

```
while(action != null){

    Vector T = mdp.getTransition(state,
action);

    int s = T.size();

    double nextUtil = 0;

    for(int i=0; i<s; ++i) {

        Transition t=(Transition)T.get(i);
```

```
        double prob=t.probability;

        State sPrime=t.nextState;

        nextUtil += (prob *
mdp.getUtility(sPrime));

    }

    if(action == policyAction) {

        policyUtility = nextUtil;

    }

    if(nextUtil > maxCurrentUtil){

        maxCurrentUtil = nextUtil;

        maxAction = action;

    }

    action = mdp.getNextAction();

}

if(maxCurrentUtil > policyUtility){

    mdp.setAction(state, maxAction);

    changed = true;

}

state = mdp.getNextState();

}
```

Resultados

Foi realizado o small grid

world fornecido em aula. Esse small grid world é uma matriz 4x4, onde o primeiro e o último elemento representam o objetivo e o restante são os possíveis estados que o agente pode tomar.
















As ações do problema são:

- Norte
- Sul
- Leste
- Oeste.

Small Grid World utilizado:

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Política resultante:

Conclusão

A partir do algoritmo implementado foi possível executar o exemplo indicado com sucesso, absorvendo os conceitos teóricos e as aplicações. Seus resultados mostraram a possibilidade de resolver o problema, conseguindo chegar em uma política que solucionou o problema.

Referências

1. Reinforcement learning - Disponível em https://en.wikipedia.org/wiki/Reinforcement_learning. Acesso em: 19 de Dez. de 2017
2. A brief introduction to reinforcement learning - Disponível em (<https://www.cs.ubc.ca/~murphyk/Bayes/pomdp.html>). Acesso em: 19 de Dez. de 2017