Computação Natural - GridWorld Benchmark

Diogo Braga

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a82547}@alunos.uminho.pt

1 Introdução e conceção

Neste *benchmark* do jogo *GridWorld*, muito utilizado como primeiro *approach* com o *Reinforcement Learning*, vão estar em foco as implementações **Q-Learning** e **SARSA**. Ambos são algoritmos que funcionam no sentido de aprender uma política, de forma a comunica-la a um agente, para que este saiba que ação executar em determinadas circunstâncias.

O **Q-Learning** segue uma abordagem *off-policy*, o que significa que a sua aprendizagem é realizada através do caminho ótimo em cada iteração. Devido a tal, esta implementação é considerada de *exploitation*, pois obtendo o primeiro melhor caminho, executa-o repetidamente nas seguintes iterações. É definido como sendo um algoritmo *greedy*.

O **SARSA**, por outro lado, segue uma abordagem *on-policy*, demorando um pouco mais a aprender o melhor caminho mas de uma forma mais segura, pois tenta sempre explorar outros caminhos que podem ter melhor recompensa que o primeiro melhor caminho encontrado. Esta implementação é, por isso, considerada de *exploration*. Este algoritmo, em comparação com o *Q-Learning*, pode demorar um pouco mais a convergir.

Neste documento, vai ser possível visualizar estes termos mais teóricos, mas aplicados de uma forma mais prática, comparando as duas técnicas. Este *benchmark* vai analisar dois principais fatores, são eles: **quantidade da recompensa** e **número de ações**.

2 Realização e resultados

Numa primeira fase, a análise dos resultados é realizada com os valores pré-definidos, de forma a ter uma ideia geral no ponto de partida. Deste modo, vai ser possível avaliar a importância das variáveis no cálculo dos *Q-Values* atribuídos às células do tabuleiro do jogo.

Os valores pré-definidos são:

- **episodes:** 20;
- **alpha (learning rate):** 0.7;
- **gamma (discount reward of future decisions):** 0.7;
- **epsilon (greedy action selection):** 0.3;
- **epsilon_degradation (more exploitation over episode):** 0.03.

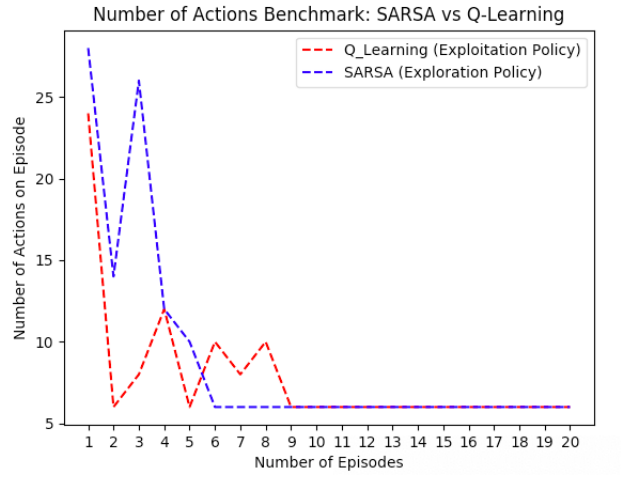

Fig. 1. Benchmarking das ações

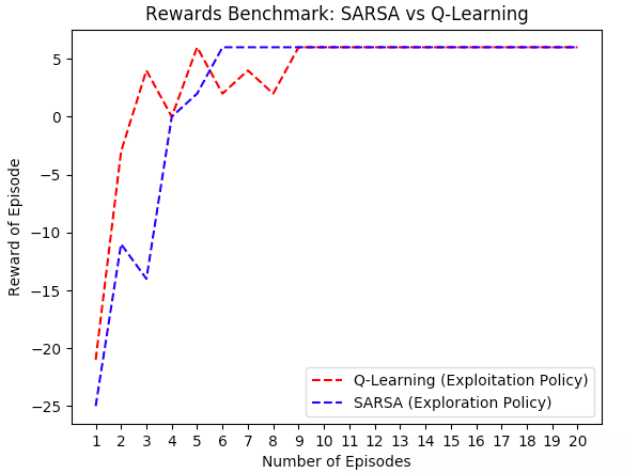

Fig. 2. Benchmarking das recompensas

Destes gráficos podemos retirar algumas conclusões:

- No **Q-Learning**, é possível verificar uma aproximação mais rápida aos valores próximos dos valores ideais, tanto nas ações como nas recompensas, pois este algoritmo segue uma abordagem de *exploitation*, optando sempre pelo caminho que apresenta o melhor *Q-Value* em todas as iterações (*greedy*).
- No **SARSA**, é possível verificar, nos episódios iniciais, variações mais acentuadas que se devem à abordagem de *exploration* que procura, desta forma, variados caminhos além do escolhido numa primeira fase. Verifica-se, na mesma, uma aproximação ao valor ótimo mas, neste caso, com influência da degradação aplicada no valor de *epsilon*, que vai aumentando os parâmetros de *exploitation*.
- Apesar das diferentes abordagens, ambas as implementações atingem a solução ótima.

2.1 Alteração do *alpha*

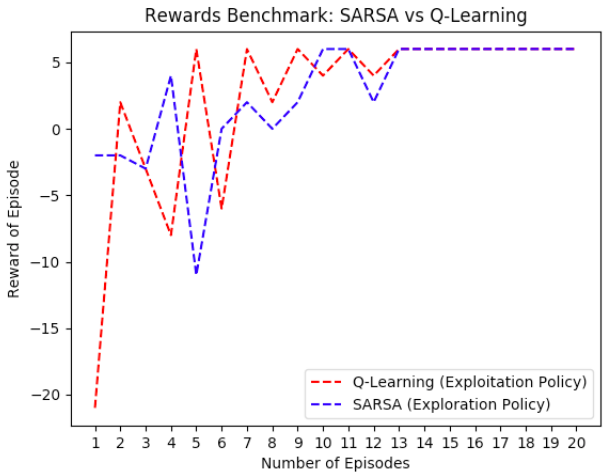

Fig. 3. Benchmarking das recompensas com *alpha* = 0.2

Deste gráfico é possível concluir:

- Em comparação com o gráfico inicial (*alpha* = 0.7, convergência no episódio 9), a convergência para a solução ótima acontece mais tarde, neste caso, no episódio 13.
- Reduzindo o ritmo de aprendizagem, a convergência dos algoritmos para a solução ótima demora, de facto, mais tempo a acontecer. Tal faz sentido acontecer pois, com o *learning rate* mais baixo, a propagação dos resultados para os *Q-Values* das células acontece com menos influência.
- No entanto, apesar de uma grande variação no valor do *alpha*, o número de episódios não sofreu um atraso tão considerado, pelo que, em alguns casos, diminuir este valor pode ser vantajoso para privilegiar uma exploração mais abrangente.

2.2 Alteração do *gamma*

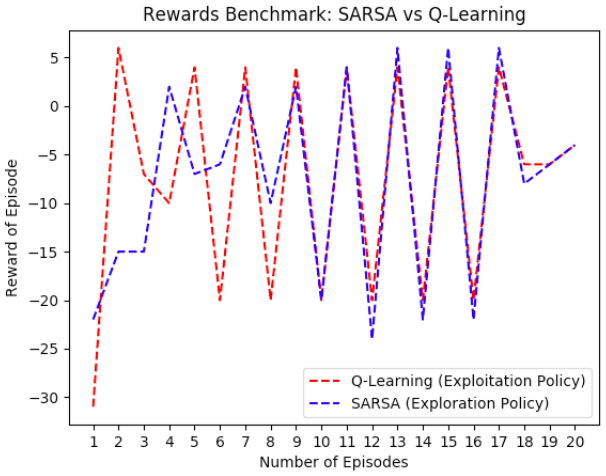

Fig. 4. Benchmarking das recompensas com *gamma* = 0.0

Deste gráfico é possível concluir:

- Em comparação com o gráfico inicial (*gamma* = 0.7, convergência no episódio 9), a convergência para a solução ótima não acontece com *gamma* = 0.0.
- Colocando o valor das recompensas com influência nas decisões futuras igual a 0, o agente não vai conseguir tirar vantagens do conhecimento adquirido em cada iteração e vai sempre procurar um caminho novo, considerando apenas as recompensas atuais que adquire.

2.3 Alteração do *epsilon*

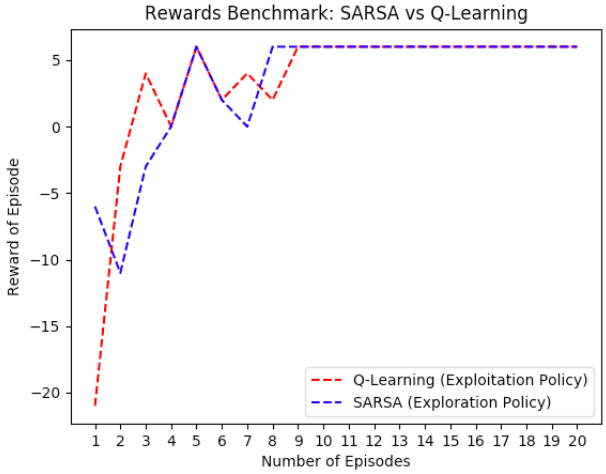

Fig. 5. Benchmarking das recompensas com *epsilon* = 0.1

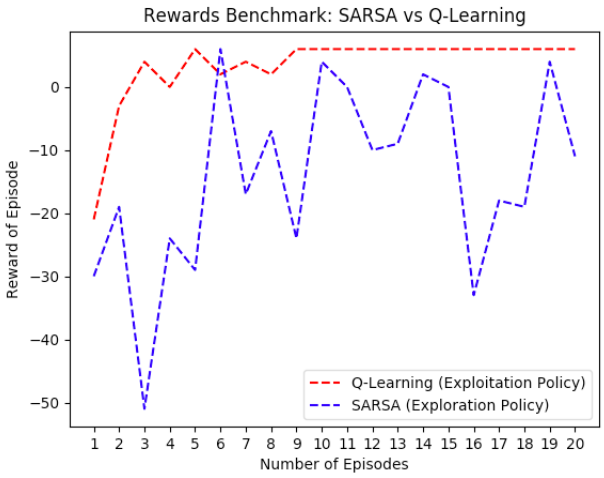

Fig. 6. Benchmarking das recompensas com *epsilon* = 0.8 e sem degradação

Deste gráfico é possível concluir:

– No primeiro gráfico, em ambas as implementações é possível verificar uma semelhança do comportamento, e tal acontece devido ao baixo *epsilon* que torna ambas as abordagens *greedy*.
– No segundo gráfico, no **Q-Learning** é possível verificar uma normal convergência, pois a sua implementação por si é sem exploração. No entanto, no **SARSA** verificam-se grandes variações nas recompensas devido ao elevado *epsilon*, o que aumenta ainda mais a exploração que já é implementada no método.
– Portanto um *epsilon* elevado leva a mais exploração, enquanto um *epsilon* baixo leva o algoritmo para uma tendência *greedy*.

3 Conclusão e principais dificuldades

Com este *benchmark* foi possível verificar os diferentes fins a que se podem chegar com variadas definições das variáveis. Nesta lógica, os valores devem ser ajustados de acordo com a finalidade do problema, no sentido de tentar tirar o melhor proveito destas duas implementações de *Reinforcement Learning*.

Com uma boa fundamentação teórica nas duas abordagens, a realização e análise deste *benchmark* foi acessível e bastante proveitosa para constatar os conceitos de uma forma prática.