Trabalho I Aprendizado por Reforço – Roberto Martins

2025/05/07

Aplicabilidade do Aprendizado por Reforço: Uma Investigação Teórico-Prática

1. Introdução

Este projeto investiga e aplica conceitos de Aprendizado por Reforço (AR) com base em estudos científicos atuais. Inclui a análise crítica de três artigos sobre aplicabilidade de AR e uma simulação prática usando Python e Gymnasium para explorar Processos de Decisão de Markov (PDM), Programação Dinâmica (PD) e Busca em Árvore de Monte Carlo (MCTS).

2. Resumo e Análise Crítica dos Artigos

Artigo 1: Predição precisa da estrutura de interações biomoleculares com AlphaFold 3

- **Identificação:** Abramson, J. et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, *630*, 493–500. Qualis: A1. Link: https://www.nature.com/articles/s41586-024-07487-w
- **Problema:** Prever estruturas 3D de interações biomoleculares complexas (proteínas com ligantes, DNA/RNA, modificações químicas), superando a limitação do AlphaFold 2 a proteínas.
- **Método:** Utiliza modelagem de difusão, refinando ruído aleatório até estruturas moleculares precisas, com arquitetura simplificada e processamento unificado de moléculas.
- **Resultados:** Melhorias significativas na predição de diversas interações, incluindo 65% em interações anticorpo-antígeno. Cria um sistema unificado para todos os tipos moleculares no Protein Data Bank.
- **Conexão com AR:** Embora não seja AR explícito, o processo iterativo de refinamento e o módulo de confiança (similar à função de valor) ecoam a tomada de decisão sequencial e a estimação de valor do AR.

Artigo 2: Descobrindo algoritmos de multiplicação de matrizes mais rápidos com aprendizado por reforço

- **Identificação:** Fawzi, A. et al. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, *610*, 47–53. Qualis: A1. Link: https://www.nature.com/articles/s41586-022-05172-4
- **Problema:** Encontrar algoritmos de multiplicação de matrizes mais eficientes, um problema NP-difícil fundamental.
- **Método:** AlphaTensor, que formula a descoberta como um jogo (TensorGame) resolvido com o framework AlphaZero. Uma rede neural guia uma Busca em Árvore Monte Carlo (MCTS), aprendendo por auto-jogo.
- **Resultados:** Descobriu um algoritmo para matrizes 4×4 com 47 multiplicações (superando o recorde de 49 de Strassen). Melhorou algoritmos para mais de 70 tamanhos de matriz e otimizou para hardware específico.
- **Conexão com AR:** Aplicação direta de AR: TensorGame como um Processo de Decisão de Markov (PDM), MCTS para busca no espaço de ações e aprendizado de função de valor para guiar a estratégia.

Artigo 3: DeepSeek-R1: Incentivando a Capacidade de Raciocínio em LLMs via Aprendizado por Reforço

- **Identificação:** Guo, D. et al. (2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*. Qualis: N/A (preprint). Link: https://arxiv.org/abs/2501.12948
- **Problema:** Desenvolver capacidades de raciocínio em Modelos de Linguagem Grandes (LLMs) com mínima supervisão, usando AR como motor principal.
- **Método:** Apresenta DeepSeek-R1-Zero (AR puro) e DeepSeek-R1 (AR com dados mínimos). Usa o algoritmo "Group Relative Policy Optimization" (GRPO), que compara múltiplas respostas geradas para calcular recompensas.
- **Resultados:** DeepSeek-R1-Zero melhorou de 15.6% para 71.0% no benchmark AIME 2024. DeepSeek-R1 igualou ou superou modelos de ponta em tarefas de raciocínio e transferiu conhecimento para modelos menores.

- **Conexão com AR:** Raciocínio como um PDM (geração de token = ação). GRPO como forma de aproximação de função de valor. Demonstra o AR treinando raciocínio complexo com incentivos.

3. Aplicação Prática: FrozenLake-v1 com PD e MCTS

Utilizamos o ambiente FrozenLake-v1 (Gymnasium) – um agente navegando em gelo escorregadio (4x4) para alcançar um objetivo (G) evitando buracos (H) – para demonstrar conceitos de AR.**3.1. Fundamentos Teóricos Concisos**

- **PDMs:** Modelam o problema com estados (células da grade), ações (movimentos), probabilidades de transição (estocásticas) e recompensas (+1 no objetivo). O objetivo é achar a política ótima π.
- **PD (Iteração de Valor):** Resolve PDMs com modelo conhecido, atualizando iterativamente o valor V(s) de cada estado até a convergência, usando a equação de Bellman.
- **MCTS:** Algoritmo de busca heurística que constrói uma árvore de decisão a partir do estado atual, simulando episódios (rollouts) para estimar valores de ações, balanceando exploração e explotação (UCB1).

3.2. Metodologia Resumida Implementação em Python:

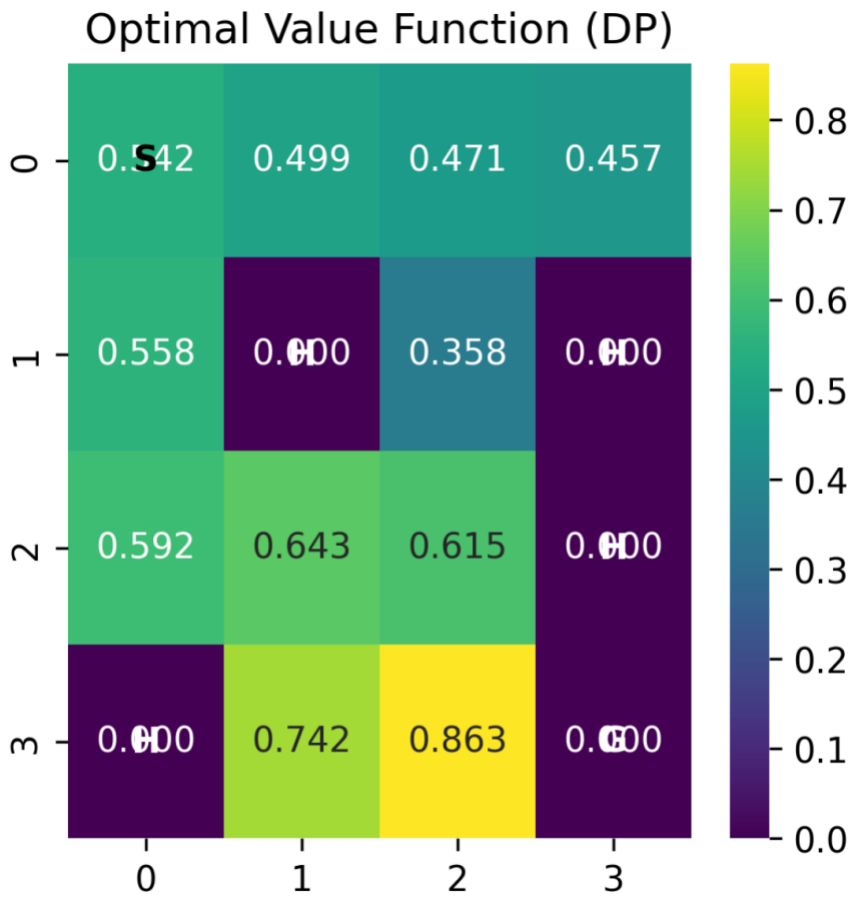
- **PD:** Iteração de Valor com γ=0.99 e θ=1e-9.
- **MCTS:** 2000 iterações/estado, constante de exploração (UCB1) = 1.414, γ=0.99. A política é derivada da ação mais visitada na raiz da árvore de busca de cada estado.

3.3. Resultados e Análise Objetiva Layout: SFFF / FHFH / FFFH / HFFG

Análise visual de um episódio utilizando DP

- **PD (Ótimo):**
  - Política típica: ←↑↑↑ / ←H←H / ↑↓←H / H→↓G
  - Valores V(s) (ex: [0.542 ...]) indicam a recompensa futura esperada, mais altos perto de G.

Análise visual do mapa de calor da iteração ótima do DP



MCTS (2000 iterações/estado):

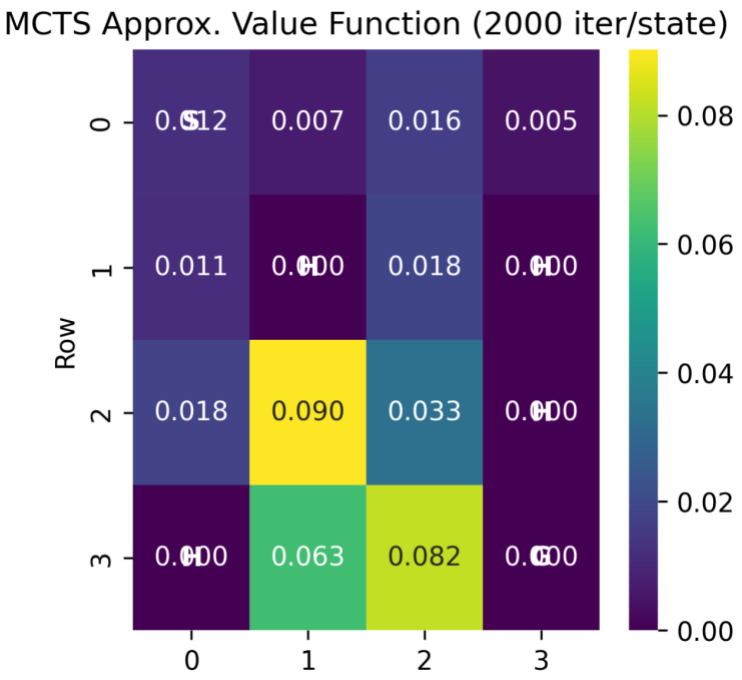
- Política típica: ←↓←← / ↓H→H / ↑↑←H / H←→G (próxima da ótima).
- Valores V(s) aproximados (ex: [0.017 ...]) tendem aos valores da PD com mais simulações.

Análise Visual de uma performance do MCTS

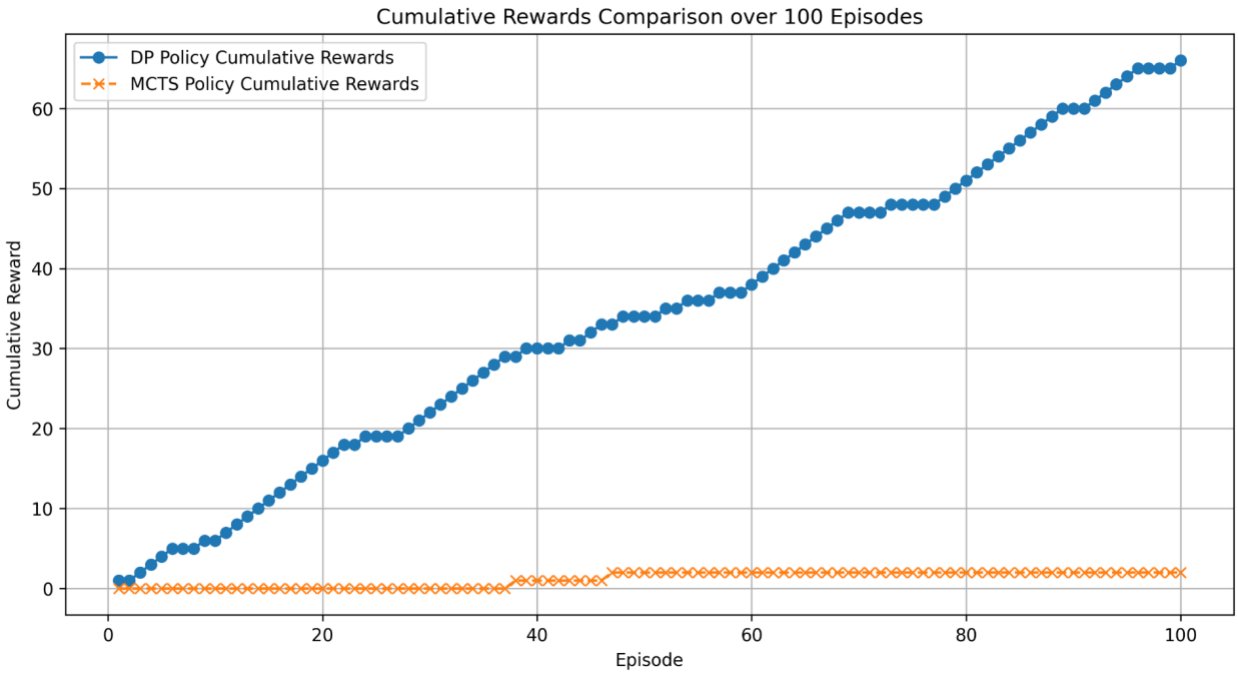
Comparativo:

- Políticas: MCTS (com iterações suficientes) aproxima-se da política ótima da PD.
- Valores: PD é exata; MCTS é uma estimativa.
- Custo: PD resolve tudo; MCTS é sob demanda (bom para espaços grandes ou quando só alguns estados importam).
- Modelo: PD requer modelo; MCTS aqui usou o modelo para simulação, mas pode ser model-free.

Análise visual do mapa de calor do da iteração do MCTS



Análise Comparativa entre a função de recompensa das abordagens



4. Conclusão Crítica: Ligando Pesquisa e Prática

O exercício com FrozenLake-v1 (PD e MCTS) ilustra a mecânica central de AR (PDMs, valor, política, exploração/explotação) que é escalada e adaptada nos artigos:

1. **AlphaFold 3:** Refinamento iterativo e escores de confiança espelham a busca por políticas e funções de valor do AR.
2. **AlphaTensor:** Usa MCTS diretamente em um PDM (TensorGame) para descobrir algoritmos, mostrando a aplicação direta dos princípios de busca e aprendizado de valor.
3. **DeepSeek-R1:** Formula o raciocínio em LLMs como um PDM, usando AR e design de recompensas para treinar estratégias complexas, similar ao FrozenLake, mas em um domínio muito mais vasto.

Esta prática demonstra que os fundamentos do AR são cruciais para as aplicações avançadas vistas na pesquisa, permitindo resolver problemas complexos em diversas áreas.

5. Referências

ABRAMSON, J. et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. **Nature**, v. 630, p. 493–500, 2024. Qualis: A1. Disponível em: https://www.nature.com/articles/s41586-024-07487-w.

2. FAWZI, A. et al. Discovering faster matrix multiplication algorithms with reinforcement learning. **Nature**, v. 610, p. 47–53, 2022. Qualis: A1. Disponível em: https://www.nature.com/articles/s41586-022-05172-4.
3. GUO, D. et al. **DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning**. 2025. Preprint arXiv:2501.12948 (cs.CL, cs.AI, cs.LG). Qualis: Não aplicável (preprint). Disponível em: https://arxiv.org/abs/2501.12948.