**Universidade de Fortaleza - UNIFOR**
Programa de Pós-graduação em Computação Aplicada - PPGIA

# Graph Machine Learning aplicado ao Aprendizado Por Reforço para Suporte à Inovação e Desenvolvimento Tecnológico

Antonio Marcos Aires Barbosa

4 de julho de 2023

## Conteúdo

# 1 Introdução

Inteligência Analítica é o campo que engloba o uso de técnicas avançadas de análise de dados, tais como mineração de dados, machine learning, processamento de linguagem natural, e estatísticas, para descobrir padrões, gerar insights, tomar decisões informadas e prever futuros comportamentos e tendências. Essa prática geralmente envolve o uso de tecnologias e software especializados para coletar, processar, limpar, analisar e visualizar dados de várias fontes e em grandes quantidades. O objetivo é ajudar indivíduos, empresas e organizações a tomar decisões baseadas em dados, em vez de apenas na intuição ou na experiência passada, particularmente quando a commplexidade do assunto demanda a construção coletiva de uma decisão que não é acessível a um agente isolado. A inteligência analítica pode ser aplicada em diversas áreas como saúde, finanças, operações, recursos humanos, marketing e muito mais. Na área de finanças, por exemplo, pode ser usada para detectar fraudes, gerenciar riscos, otimizar portfólios de investimentos. Já em marketing, pode-se tratar sobre tarefas como segmentar os clientes, prever a probabilidade de churn, otimizar campanhas publicitárias, entre muitas outras aplicações práticas.

A programação dinâmica é um poderoso método utilizado para resolver problemas de otimização que possuem uma estrutura de sobreposição de subproblemas. Em termos simples, a programação dinâmica divide um problema maior em subproblemas menores, resolve cada um deles apenas uma vez e armazena seus resultados em uma tabela (memória), de onde é possível reconstruir a solução para o problema original. Em grafos, a programação dinâmica tem uma ampla aplicação, especialmente quando se trata de problemas de caminho mais curto, contagem de caminhos, ordenação topológica e muito mais. No contexto de sistemas de recomendação, a aplicação da programação dinâmica pode ter vários ângulos, dependendo do tipo de problema que se está tentando resolver. Dentro do contexto de sistemas de recomendação baseados em conteúdo, a programação dinâmica pode ser usada para comparar sequências de opções ou escolha de itens. Isso poderia ser útil, por exemplo, para comparar diferentes lista de opções, como em opções de alternativas de tratamentos, escolha de medicamentos, delineamento de rotas tecnológicas, ou até questões mais corriqueiras de nossa vida cotidiana como organizar listas de reprodução de músicas ou sequências de filmes/séries de acordo com a preferência ou necessidade específica do usuário.

Para aplicações relacionadas com sistemas de recomendação com uso de estruturas de grafos (por exemplo, sistemas que usam dados de redes sociais, como citações, coautoria, colaborações de qualquer forma), a abordagem com grafos pode ser usada para identificar comunidades, detectar hubs de influência e analisar a estrutura geral da rede. A programação dinâmica também pode ser usada em combinação com técnicas de aprendizado de máquina em grafos, do inglês *Graph Machine Learning* (GML), como os modelos de Markov de ordem superior. Nesses casos, a programação dinâmica pode ser usada para resolver o problema de otimização subjacente ao aprendizado do modelo, que é frequentemente um problema de maximização de verossimilhança.

A aprendizagem por reforço (RL) é um ramo da inteligência artificial que oferece um arcabouço para a aprendizagem autônoma, onde um agente aprende a tomar decisões otimizadas pela interação com seu ambiente [**?**]. A incorporação de RL em GML traz um novo espectro de oportunidades e desafios. A representação em grafo oferece uma maneira natural de modelar a estrutura relacional complexa e os padrões de interação presentes em muitos problemas do mundo real, tornando-a especialmente relevante no campo da saúde e das ciências da vida [**?**].

No domínio das ciências da vida e ciências da saúde, os nós do grafo podem representar entidades como proteínas, genes, pacientes, etc., e as arestas podem representar relações entre essas entidades (por exemplo, interações proteína-proteína, similaridade genética, etc.). Aprendizado de máquina em grafos permite a detecção de padrões complexos dentro desses dados relacionais. A aplicação de RL em GML tem mostrado promessa em várias aplicações, como a descoberta de medicamentos. Por exemplo, um agente de RL poderia aprender a projetar novas moléculas com propriedades desejadas navegando no espaço

de todas as possíveis moléculas como um grafo, onde as ações representam modificações moleculares [**?**]. Outra aplicação possível é a aprendizagem de protocolos de tratamento ótimos para pacientes com base em seu histórico médico e estado atual, modelado como um grafo de paciente-tempo. As ações poderiam representar diferentes intervenções de tratamento, e a recompensa seria baseada no prognóstico do paciente [**?**].

Apesar de seu potencial, a aplicação de RL em GML também apresenta desafios únicos. O espaço de ações em problemas de GML pode ser extremamente grande e a avaliação da recompensa pode ser incerta e ruidosa. Além disso, garantir a explorabilidade suficiente durante a aprendizagem é um desafio crucial para garantir a eficácia do RL em GML [**?**].

Atualmente várias iniciativas relativas à inovação tomam forma no ecossistema nacional e local, que demandam intensamente planejamento que envolvem vários órgão, entidades e especialistas de distintas áreas e especialidades.

# 2 Programação Dinâmica: Algoritmos, Benefícios, Vantagens e Desvantagens

A programação dinâmica é uma abordagem de otimização matemática de grande importância, empregada com sucesso na resolução de uma variedade de problemas complexos. A estratégia central da programação dinâmica é decompor um problema em uma série de subproblemas menores, resolver estes subproblemas, armazenar suas soluções e, por fim, combinar essas soluções para chegar à resposta para o problema original [**?**].

## 2.1 Sistemas de recomendação

Uma classe de aplicações onde o uso desses algoritmos é particularmente útil está em sistemas de recomendação, que tratam do problema de recomendar uma sequência de opções, ou itens distintos, considerada alguma medida de custo, o "custo" pode ser medido por critérios coerentes com o problema em análise, como dissimilaridade em problemas de agrupamento e identificação de comunidades, por exemplo, onde uma estimativa de custo de passar de um item para o próximo é conhecido. Já em problemas de montagem ótima de uma cesta de itens dada uma restrição de recursos, a ideia seria considerar cada item como um vértice e cada "passo" de um item para outro como uma aresta com o "custo" como seu peso.

# 3 Algoritmos em programação dinâmica

## 3.1 Algoritmos clássicos

Alguns algoritmos notáveis incorporam a programação dinâmica, como é o caso do algoritmo de Bellman-Ford, algoritmo de Floyd-Warshall, algoritmo de Dijkstra, algoritmo de Knapsack, algoritmo de corte de hastes, dentre outros. Entre os problemas tratados com esses algoritmos bem conhecidos, que utilizam a programação dinâmica, temos exemplos como o cálculo do caminho mais curto de um vértice a todos os outros vértices em um grafo ponderado com o algoritmo de Bellman-Ford (ABF). Ou de forma semelhante ao algoritmo de Bellman-Ford, também o algoritmo de Floyd-Warshall (AFW) é utilizado para determinar o caminho mais curto em um grafo [**?**]. Porém, enquanto o ABF busca o caminho mais curto de um vértice único para todos os outros vértices, o AFW determina o caminho mais curto entre todos os pares de vértices.

Já com o algoritmo de Dijkstra, embora semelhante ao de Bellman-Ford na busca do caminho mais curto, os problemas passíveis de tratamento mais eficiente são aqueles cujas situações podem ser modeladas

em um grafo que não contêm ciclos de peso negativo [**?**]. Em uma grande quantidade de aplicações possíveis, busca-se soluções ótimas, ou próximas do ótimo, para problemas como o abordado pelo algoritmo de Knapsack (problema da mochila), ou o algoritmo de corte de hastes, ambos sendo utilizados para resolver problemas de otimização de recursos limitados. Porém, enquanto o primeiro se destina a maximizar o valor total que pode ser colocado em uma mochila com capacidade limitada, o segundo visa maximizar o lucro obtido ao cortar uma haste de comprimento fixo em pedaços de comprimentos variados [**?**].

### 3.1.1 Benefícios e Vantagens

Os algoritmos de programação dinâmica proporcionam uma abordagem eficiente para resolver problemas complexos de otimização. Como o tempo de execução desses algoritmos é geralmente polinomial, eles podem lidar com problemas que seriam intratáveis para algoritmos de força bruta. Além disso, graças à memorização, a programação dinâmica evita a redundância de cálculos ao resolver subproblemas sobrepostos [**?**].

### 3.1.2 Desvantagens

Por outro lado, a programação dinâmica possui desvantagens. A principal desvantagem é o alto uso de memória, pois a solução para cada subproblema precisa ser armazenada. Além disso, nem todos os problemas de otimização são adequados para programação dinâmica. Os problemas devem possuir as propriedades de sobreposição de subproblemas e subestrutura ótima para que a programação dinâmica seja efetiva [**?**].

## 3.2 Outros algoritmos em Programação Dinâmica

Enquanto os algoritmos clássicos de programação dinâmica continuam sendo muito eficazes, pesquisas recentes têm levado ao desenvolvimento de novos algoritmos que tentam abordar suas limitações e expandir suas capacidades.

### 3.2.1 Árvore de Junção Dinâmica

Um exemplo notável é a Árvore de Junção Dinâmica, do inglês *Dynamic Junction Tree Algorithm* (DJT), que foi projetada para efetuar inferência em redes bayesianas e gráficos de Markov. A DJT expande as técnicas clássicas de programação dinâmica para permitir atualizações mais eficientes quando o modelo subjacente muda [**?**].

### 3.2.2 Programação Dinâmica Aproximada

Outra abordagem inovadora é a Programação Dinâmica Aproximada *Approximate Dynamic Programming* (ADP), que procura resolver problemas de decisão sequencial sob incerteza, especialmente em contextos em que o espaço de estados é grande ou contínuo. A ADP faz uso de técnicas de aproximação, como a aprendizagem por reforço, para lidar com problemas que seriam impraticáveis para a programação dinâmica clássica [**?**].

### 3.2.3 Algoritmo Fast-DT

Além disso, o algoritmo Fast-DT, um método de programação dinâmica para detecção rápida de comunidades em redes complexas, apresenta uma forma eficaz de identificar comunidades de grande escala em redes com até milhões de nós, superando assim limitações de escalabilidade de métodos anteriores [**?**].

Embora estes algoritmos ainda não sejam tão amplamente utilizados como seus predecessores clássicos, eles representam avanços significativos na área da programação dinâmica e prometem expandir a aplicabilidade desta técnica a uma gama ainda mais ampla de problemas.

# Referências


[Sutton e Barto 1998]SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. 1st edition. ed. [S.l.]: MIT Press, 1998.

[Weng e Ghassemi 2020]WENG, P. S. W.-H.; GHASSEMI, M. Digital health: Challenges and future directions after covid-19. *Nature Medicine*, 2020.

[Wu Shirui Pan e Yu 2020]WU SHIRUI PAN, F. C. G. L. C. Z. Z.; YU, P. S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[You Bowen Liu e Leskovec 2018]YOU BOWEN LIU, Z. Y. V. P. J.; LESKOVEC, J. Graph convolutional policy network for goal-directed molecular graph generation. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2018.