

Aprendizado por Reforço
Prof. Domingos Napolitano
Aula 1: Introdução aos K-armed Bandits

Agenda da Aula

01

Problema do K-armed Bandit

Definição, origens e formulação matemática (30 min)

03

Exploração vs. Aproveitamento

O dilema fundamental do aprendizado (25 min)

05

Bandits Não-Estacionários

Quando distribuições mudam (20 min)

07

Aplicações no Mundo Real

Casos de uso práticos (15 min)

09

Resumo

Principais conclusões (5 min)

02

Valor da Ação e Métodos

Estimativas, cálculos e implementações (40 min)

04

Implementação Incremental

Técnicas eficientes de atualização (15 min)

06

Testbed e Experimentos

Avaliação de algoritmos (20 min)

08

Conceitos Adicionais

Métodos avançados (10 min)



Objetivos da Aula



Definir o problema dos K-armed bandits



Explicar o dilema exploração vs. aproveitamento



Implementar métodos de seleção de ação



Aplicar conceitos em problemas reais



Avaliar algoritmos usando testbeds



Conectar com RL completo

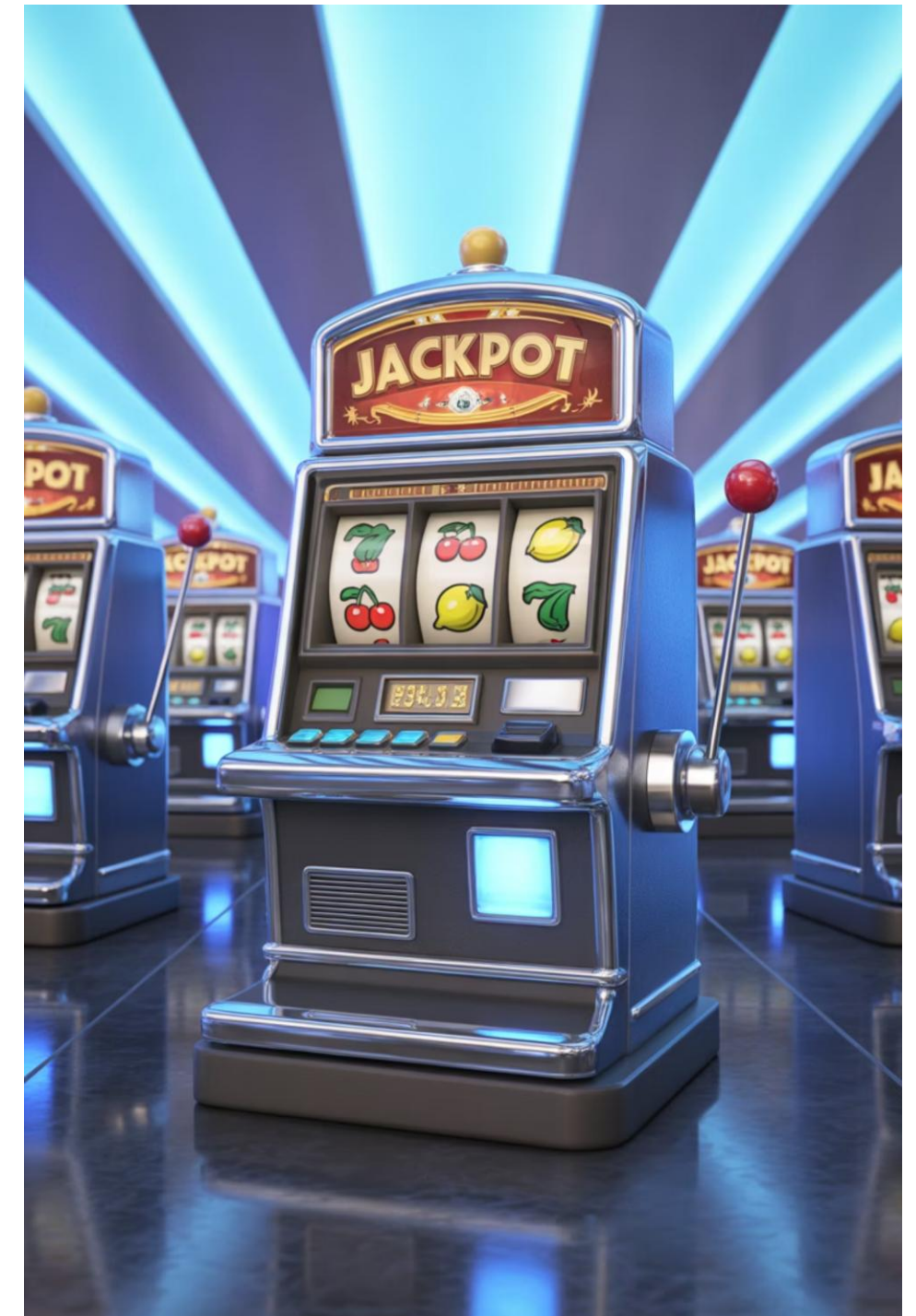
De Onde Vem o Nome?

"ONE-ARMED BANDIT" → Máquina caça-níquel que "rouba" seu dinheiro

"K-ARMED BANDIT" → Conceito estendido:

- Máquina com k alavancas diferentes
- Cada alavanca tem distribuição de pagamento diferente
- Você quer maximizar seus ganhos ao longo do tempo

❓ **PERGUNTA PARA REFLEXÃO:** Como você decidiria qual alavanca puxar se não soubesse qual tem o melhor pagamento médio?



K-Armed Bandit: Definição Formal

Elementos Básicos

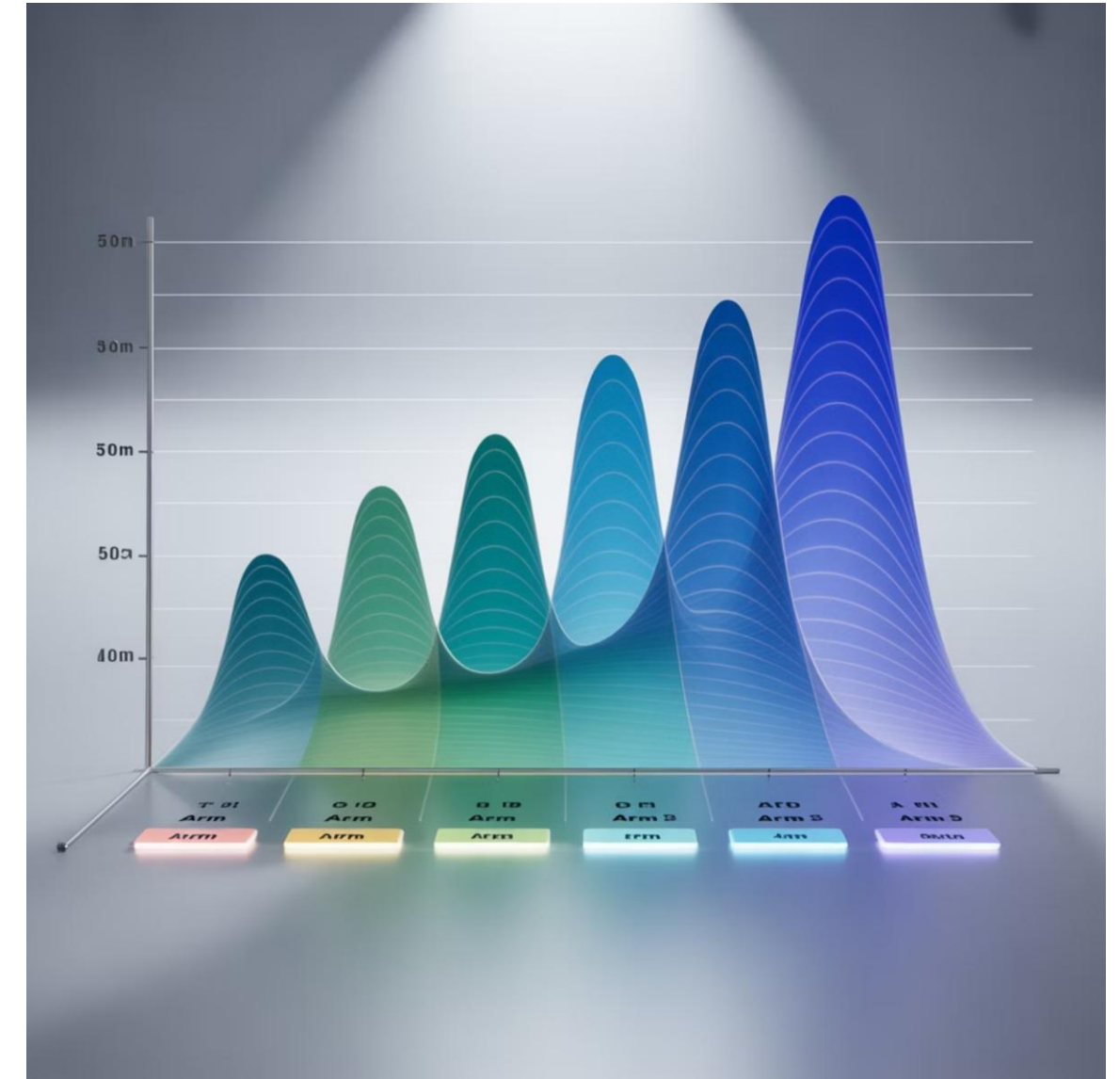
- Conjunto de k ações: $A = \{1, 2, \dots, k\}$
- Ação selecionada no tempo t : A_t
- Recompensa recebida: R_t

Valores de Ação

- Valor verdadeiro: $q^*(a) = E[R_t \mid A_t = a]$
- Valor estimado: $Q_t(a) \approx q^*(a)$

OBJETIVO: Maximizar a soma de recompensas $\sum R_t$ ao longo do tempo

⊗ **DESAFIO FUNDAMENTAL:** $q^*(a)$ é desconhecido!
Precisamos descobrir quais ações são melhores através de tentativa e erro.



Exemplo Prático

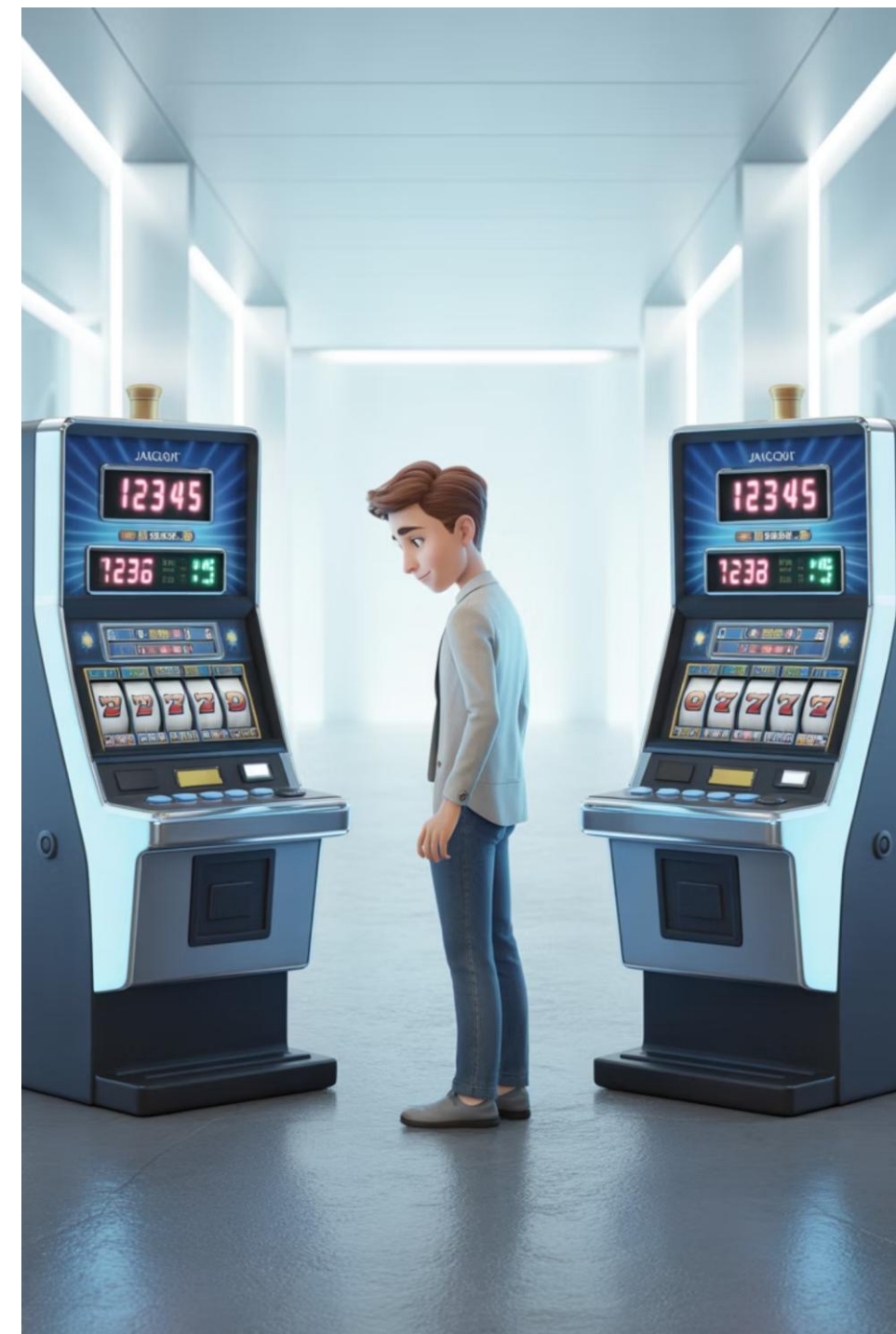
Imagine que você está testando duas máquinas caça-níquel e registra os seguintes resultados:

Tempo	Máquina A	Máquina B
1	+0.2	--
2	--	+0.8
3	+0.1	--
4	--	+0.6
5	-0.1	--

❓ **PERGUNTA PARA A TURMA:** Qual máquina parece melhor? Por quê?

Máquina A: média = +0.07

Máquina B: média = +0.70



Valor da Ação

Definição Formal

VALOR VERDADEIRO (desconhecido):

$$q(a) = E[R_t \mid A_t = a]$$

VALOR ESTIMADO (calculamos):

$$Q_t(a) = \frac{\text{soma das recompensas ao escolher } a}{\text{número de vezes que } a \text{ foi escolhida}}$$

Método de Média Amostral

$$Q_t(a) = \frac{R_1 + R_2 + \dots + R_n}{n}$$

onde n é o número de vezes que a ação a foi escolhida antes do tempo t .

OBJETIVO: À medida que coletamos mais dados, queremos que nossa estimativa $Q_t(a)$ se aproxime do valor verdadeiro $q^*(a)$.



Método Greedy vs. ϵ -Greedy

Método Greedy

REGRA: $A_t = \operatorname{argmax}_a Q_t(a)$

VANTAGENS:

- Simples de implementar
- Maximiza recompensa imediata
- Usa conhecimento atual

PROBLEMAS:

- Pode ficar "preso" em ações subótimas
- Zero exploração
- Dependente de estimativas iniciais

Método ϵ -Greedy

REGRA:

- Se $\text{random}() < \epsilon$: A_t = ação aleatória (EXPLORAÇÃO)
- Senão: $A_t = \operatorname{argmax}_a Q_t(a)$ (APROVEITAMENTO)

PROBABILIDADES:

- Ação greedy: $P = 1 - \epsilon + \epsilon/k$
- Cada ação não-greedy: $P = \epsilon/k$

VALORES TÍPICOS: $\epsilon = 0.01, 0.1, 0.3$





Exploração vs. Aproveitamento

O dilema fundamental no aprendizado por reforço:



Aproveitamento (Exploitation)

- Escolhe a melhor ação conhecida
- Maximiza recompensa imediata
- Usa conhecimento atual
- Seguro, porém limitado



Exploração (Exploration)

- Testa ações menos conhecidas
- Pode descobrir opções melhores
- Sacrifica recompensa imediata
- Arriscado, porém potencialmente melhor



IMPOSSÍVEL FAZER AMBOS SIMULTANEAMENTE! Cada ação individual é ou exploração ou aproveitamento.

Analogias do Mundo Real

O dilema exploração vs. aproveitamento aparece em muitas situações cotidianas:



Restaurante

Aproveitamento: Ir ao seu restaurante favorito onde você sempre gosta da comida

Exploração: Experimentar um restaurante novo que pode ser melhor (ou pior)



Investimentos

Aproveitamento: Manter investimentos em ações conhecidas com retorno estável

Exploração: Arriscar em novos ativos com potencial de maior retorno



Carreira

Aproveitamento: Continuar no emprego atual onde você já tem experiência

Exploração: Mudar para uma nova posição ou área com potencial de crescimento

❓ Em qual situação você exploraria mais? Quando a exploração vale mais a pena que o aproveitamento?

Implementação Eficiente

Método Ingênuo (ineficiente)

- Armazenar todas as recompensas
- Recalcular média a cada passo

Método Incremental (eficiente)

$$Q_{n+1} = Q_n + \frac{1}{n} \times [R_n - Q_n]$$

Forma Geral:

$$\text{Nova} = \text{Antiga} + \text{TamanhoPasso} \times [\text{Alvo} - \text{Antiga}]$$



Quando as Coisas Mudam

Problema Estacionário

- Distribuições de recompensa fixas
- Dados antigos sempre relevantes
- Média amostral funciona bem
- Convergência garantida com infinitas amostras

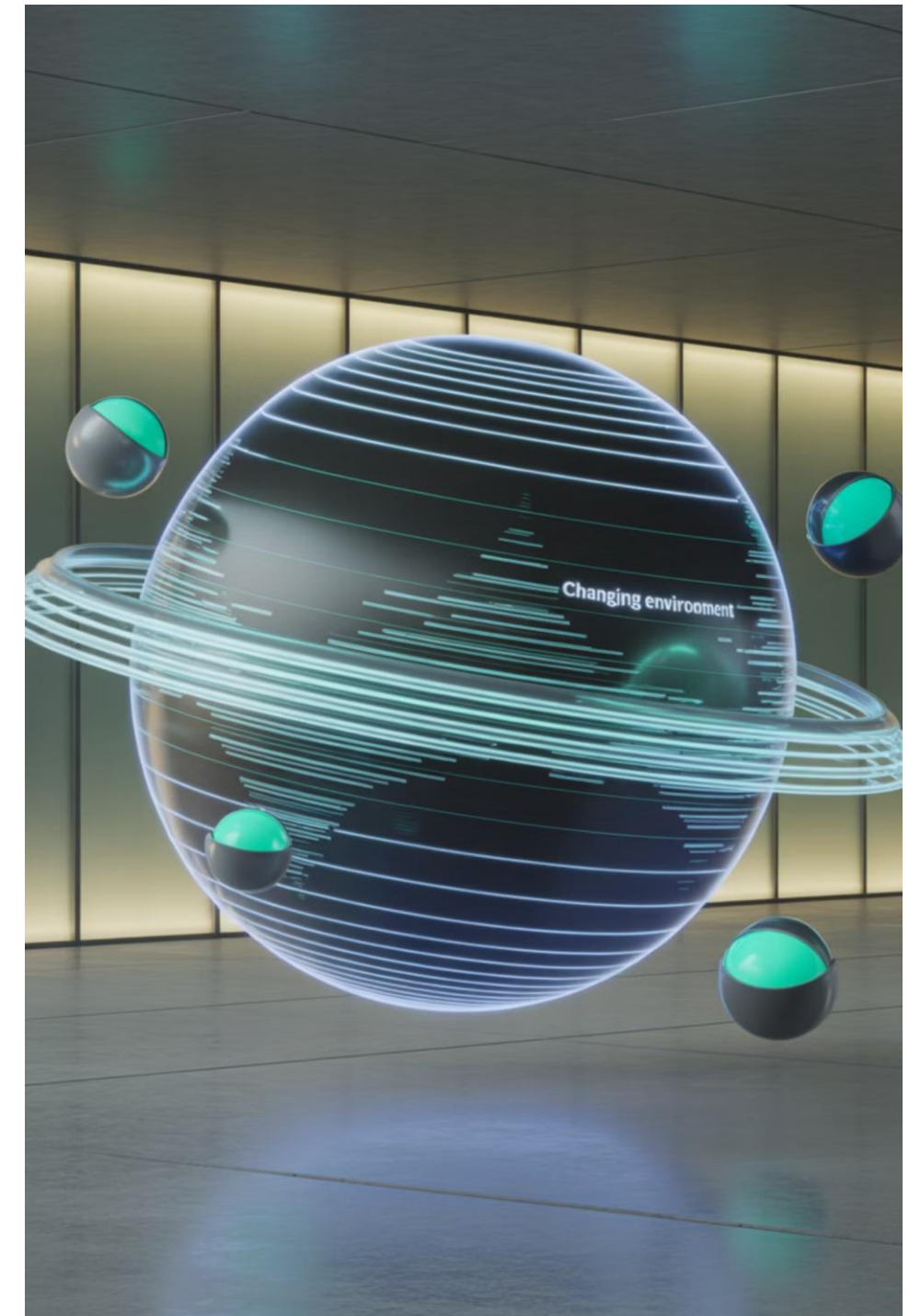
Problema Não-Estacionário

- Distribuições mudam ao longo do tempo
- Dados antigos podem ser irrelevantes
- Precisamos nos adaptar rapidamente
- Média amostral dá peso excessivo ao passado

Solução: Tamanho de Passo Constante

$$Q_{n+1} = Q_n + \alpha \times [R_n - Q_n], \quad \alpha \in (0, 1]$$

Quando $\alpha = 1$: considera apenas a observação mais recente




Avaliação Experimental

Configuração do Testbed

- $k = 10$ ações
- $q^*(a) \sim N(0, 1)$ para cada ação
- $R_t \sim N(q^*(A_t), 1)$ para recompensas
- 2000 problemas independentes
- 1000 passos por problema

Métricas de Avaliação

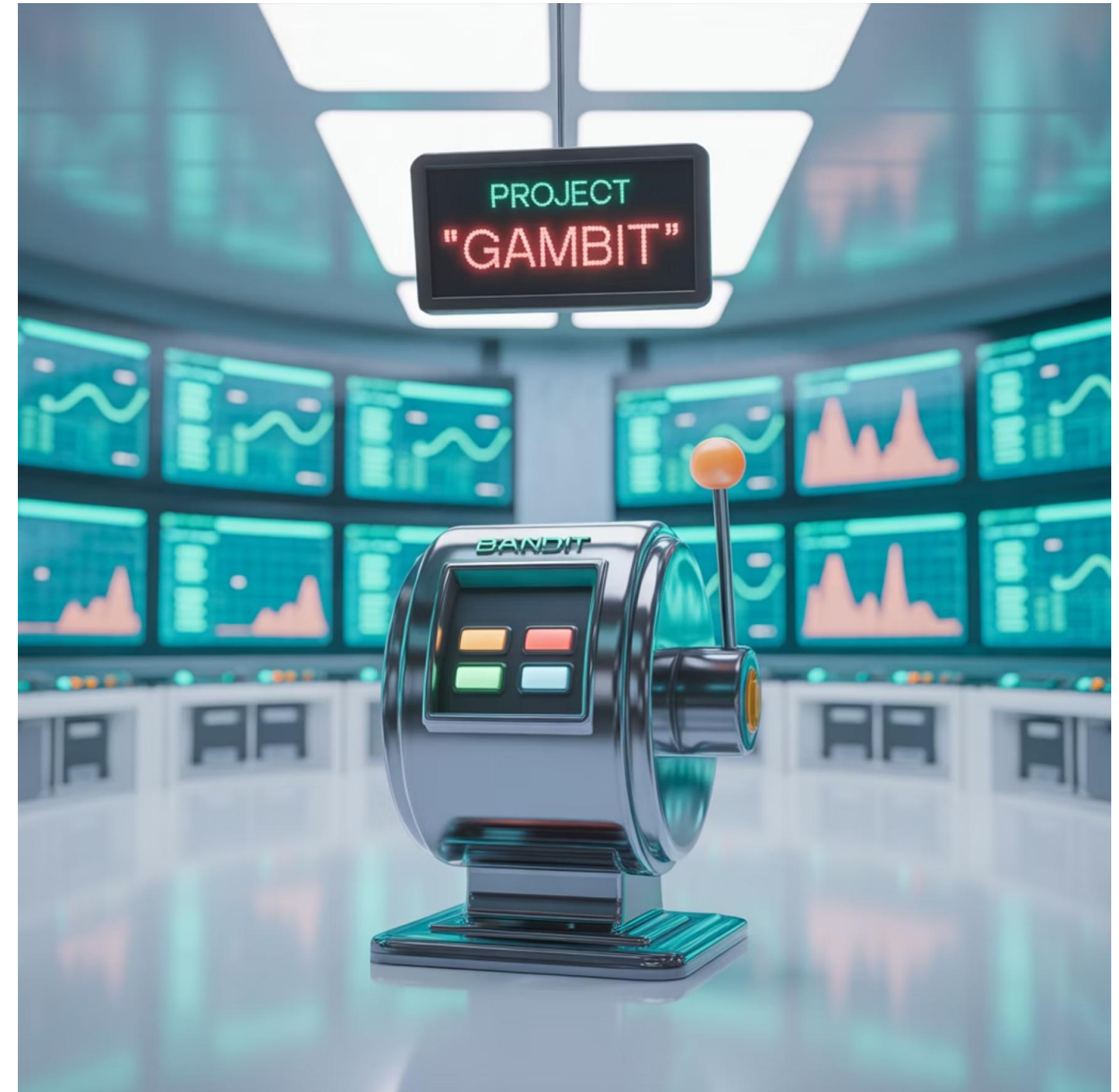
- Recompensa média ao longo do tempo
- Porcentagem de escolhas da ação ótima

 Um testbed padronizado permite a comparação justa entre diferentes algoritmos sob as mesmas condições.

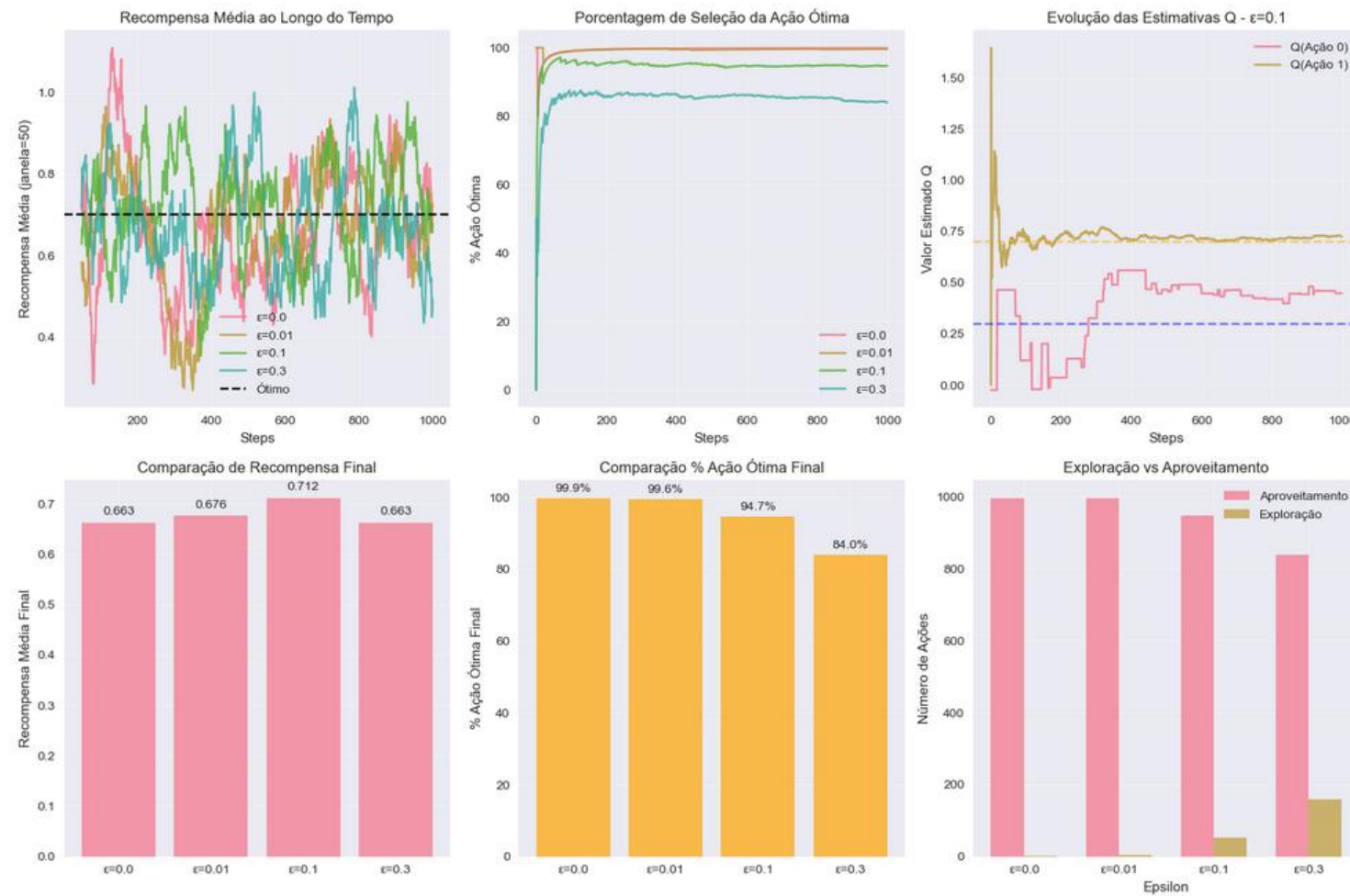
Para cada problema, geramos aleatoriamente:

- Os valores verdadeiros $q^*(a)$ de cada ação
- As recompensas específicas de cada ação escolhida

Resultado esperado: ϵ -greedy supera greedy no desempenho de longo prazo!



Greedy vs. ϵ -Greedy: Resultados



Greedy

- ✗ Melhora rápido mas estagna
- ✓ Encontra ação ótima em ~99% dos casos
- ✗ Recompensa final $\approx 0,6$
- ✓ Melhor no início (primeiros 200 passos)

ϵ -Greedy ($\epsilon=0.1$)

- ✓ Melhora mais lento inicialmente
- ✗ Encontra ação ótima em ~95% dos casos
- ✓ Recompensa final $\approx 0,71$
- ✓ Superior a longo prazo



LIÇÃO IMPORTANTE: Exploração compensa a longo prazo! O sacrifício inicial de recompensa imediata resulta em desempenho superior posteriormente.

UPPER CONFIDENCE BOUND (UCB)

Problema com ϵ -greedy:

- Exploração é aleatória e indiscriminada.
- Não considera a incerteza nas estimativas das ações.
- Trata ações com "quase ótimas" igual às "claramente ruins".

Solução UCB:

- Considera tanto o valor estimado da recompensa quanto a incerteza associada.
- Favorece ações com alta estimativa de recompensa OU alta incerteza.
- Garante que ações menos exploradas, mas potencialmente melhores, tenham sua chance.

Fórmula UCB:

$$A_t = \underset{a}{\operatorname{argmax}} [Q_t(a) + c \sqrt{\frac{\ln(t)}{N_t(a)}}]$$

$Q_t(a)$

Valor estimado da recompensa para a ação **a** no passo **t**.
Representa o quanto já sabemos sobre a ação (aproveitamento).

$c\sqrt{\ln(t)/N_t(a)}$

Termo de confiança que quantifica a incerteza sobre a ação **a**.
Incentiva a exploração de ações menos visitadas. O parâmetro **c** > **0** controla o grau de exploração.

O UCB equilibra a exploração e o aproveitamento de forma mais sofisticada, focando em ações que têm maior potencial de serem ótimas, mesmo que ainda não tenhamos certeza.



UCB EM AÇÃO - EXEMPLO

Para ilustrar como o UCB equilibra exploração e aproveitamento, considere o seguinte cenário após 20 passos ($t = 20$), com um parâmetro de exploração de $c = 2$:

Ação 1

Valor Estimado (Q): 0.8

Vezez Escolhida (N): 10

$$\begin{aligned}\text{Cálculo UCB: } & 0.8 + 2\sqrt{(\ln(20)/10)} = 0.8 \\ & + 0.96 = 1.76\end{aligned}$$

Ação 2

Valor Estimado (Q): 0.6

Vezez Escolhida (N): 2

$$\begin{aligned}\text{Cálculo UCB: } & 0.6 + 2\sqrt{(\ln(20)/2)} = 0.6 \\ & + 2.45 = 3.05\end{aligned}$$

Ação 3

Valor Estimado (Q): 0.4

Vezez Escolhida (N): 1

$$\begin{aligned}\text{Cálculo UCB: } & 0.4 + 2\sqrt{(\ln(20)/1)} = 0.4 \\ & + 2.99 = 3.39\end{aligned}$$

Neste exemplo, o UCB escolhe a **Ação 3**! Embora ela tenha o menor valor de recompensa estimado ($Q = 0.4$), seu termo de incerteza é o maior, impulsionando sua pontuação UCB para o valor máximo.

Em contraste, um algoritmo puramente ϵ -greedy (com ϵ pequeno) provavelmente continuaria a explorar a Ação 1 na maioria das vezes, por ter o maior Q estimado.

"Melhor testar uma ação com grande incerteza e potencial, do que continuar com uma ação que já conheço bem e talvez não seja a ótima."



UCB EM AÇÃO - EXEMPLO

Para ilustrar como o UCB equilibra exploração e aproveitamento, considere o seguinte cenário após 20 passos ($t = 20$), com um parâmetro de exploração de $c = 2$:

Ação 1

Valor Estimado (Q): 0.8 Vezes Escolhida (N): 10

$$\begin{aligned} \text{Cálculo UCB: } 0.8 + 2\sqrt{(\ln(20)/10)} &= 0.8 \\ + 0.96 &= 1.76 \end{aligned}$$

Ação 2

Valor Estimado (Q): 0.6 Vezes Escolhida (N): 2

$$\begin{aligned} \text{Cálculo UCB: } 0.6 + 2\sqrt{(\ln(20)/2)} &= 0.6 \\ + 2.45 &= 3.05 \end{aligned}$$

Ação 3

Valor Estimado (Q): 0.4 Vezes Escolhida (N): 1

$$\begin{aligned} \text{Cálculo UCB: } 0.4 + 2\sqrt{(\ln(20)/1)} &= 0.4 \\ + 2.99 &= 3.39 \end{aligned}$$

Neste exemplo, o UCB escolhe a **Ação 3**! Embora ela tenha o menor valor de recompensa estimado ($Q = 0.4$), seu termo de incerteza é o maior, impulsionando sua pontuação UCB para o valor máximo.

Em contraste, um algoritmo puramente ϵ -greedy (com ϵ pequeno) provavelmente continuaria a explorar a Ação 1 na maioria das vezes, por ter o maior Q estimado.

"Melhor testar uma ação com grande incerteza e potencial, do que continuar com uma ação que já conheço bem e talvez não seja a ótima."

Sistemas de Recomendação

Os algoritmos de bandits são amplamente utilizados em plataformas digitais para personalizar recomendações:

Netflix

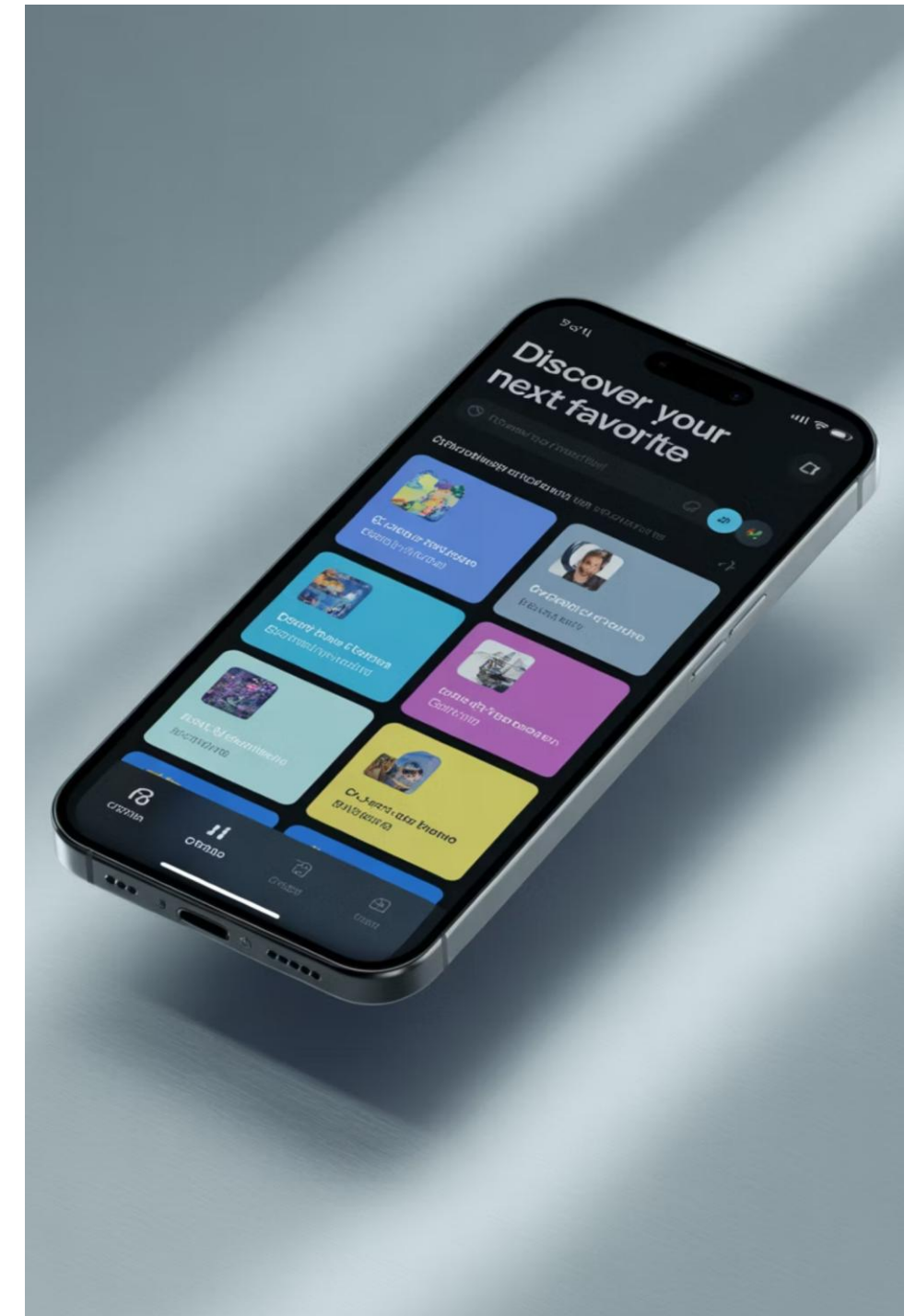
- **Ações:** Filmes e séries diferentes
- **Recompensa:** Tempo assistido, conclusão do conteúdo
- **Dilema:** Recomendar populares vs. descobrir preferências
- **Desafio:** Preferências mudam com o tempo (não-estacionário)

Spotify

- **Ações:** Músicas e playlists
- **Recompensa:** Músicas ouvidas completamente, adições à biblioteca
- **Dilema:** Hits conhecidos vs. descoberta musical
- **Estratégia:** Playlists "Descobertas da Semana" (exploração)

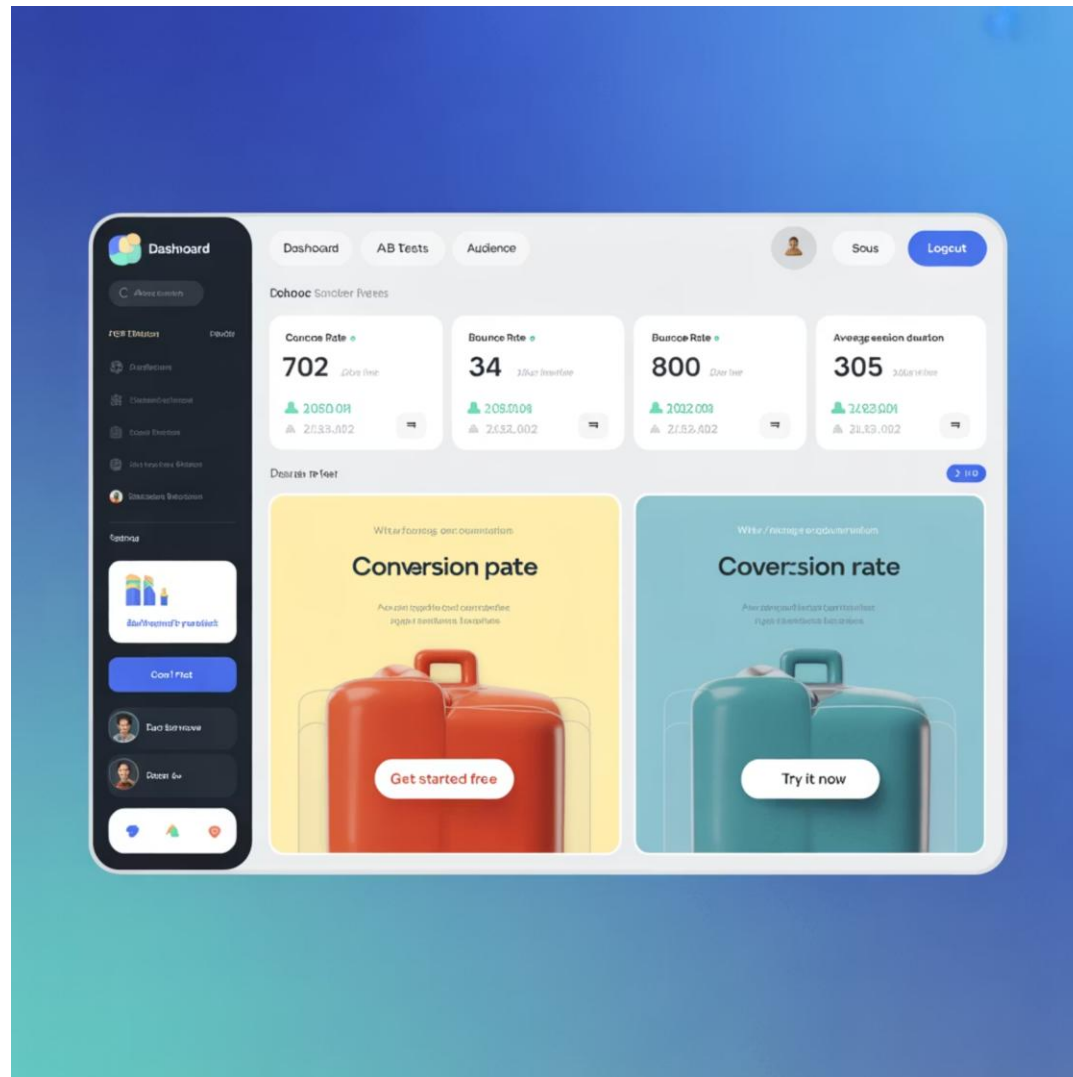
YouTube

- **Ações:** Vídeos diferentes
- **Recompensa:** Tempo de visualização, engajamento (curtidas, comentários)
- **Complexidade:** Milhões de ações possíveis
- **Implementação:** Bandits contextuais com informações do usuário



Testes A/B

Técnicas de bandit podem superar testes A/B tradicionais ao equilibrar melhor exploração e aproveitamento:



Problema: Qual versão é melhor?

Testes A/B tradicionais:

- Divisão fixa (50/50) entre versões
- Espera até o final do teste para tomar decisão
- Muitos usuários expostos à versão inferior

Bandits para Testes A/B

- Adaptação dinâmica da distribuição de tráfego
- Mais tráfego para versões com melhor desempenho
- Equilibra teste (exploração) e otimização (aproveitamento)

Aplicações Comuns:

- **Websites:** Layout, posição de botões, cores
- **Apps:** Diferentes interfaces, fluxos de usuário
- **Email Marketing:** Assuntos, conteúdos, horários



Empresas como Booking.com, Microsoft e Google relatam ganhos significativos ao substituir testes A/B tradicionais por algoritmos de bandits.



Ensaio Clínicos e Considerações Éticas

Problema: Qual tratamento é mais eficaz?

Configuração tradicional:

- Divisão aleatória de pacientes entre tratamentos
- Análise somente após conclusão do estudo
- Metade dos pacientes recebe tratamento inferior

Abordagem com bandits:

- Alocação adaptativa de pacientes
- Mais pacientes recebem tratamentos promissores
- Análise contínua durante o estudo

Considerações Éticas Especiais

- Minimizar danos potenciais aos pacientes
- Descobrir tratamentos eficazes rapidamente
- Balancear avanço científico com bem-estar individual

⚠ Questão Ética Fundamental: Como equilibrar a necessidade de dados confiáveis (exploração) com o dever de fornecer o melhor tratamento conhecido (aproveitamento)?

Os algoritmos de bandit oferecem uma abordagem mais ética ao reduzir o número de pacientes expostos a tratamentos inferiores, mantendo a validade científica.

Valor da Ação (Value Action)

- O Valor de uma ação é a recompensa esperada quando a ação é tomada

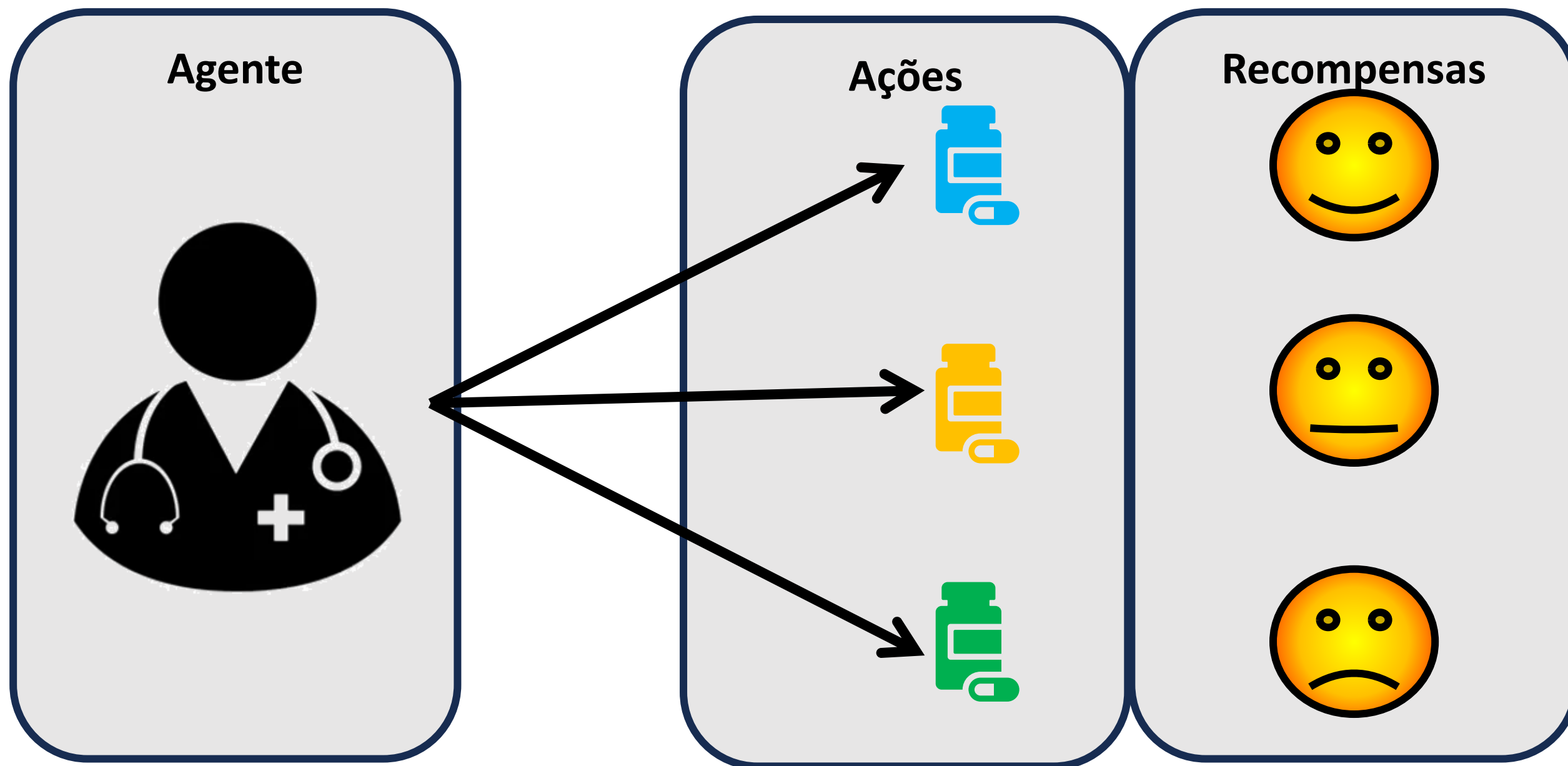
$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a]$$

- $q_*(a)$ não é conhecido assim nós o estimamos

Imagine o caso de um medico que precisa decidir entre três medicamentos






Imagine o caso de um medico que precisa decidir entre três medicamentos



Digamos que sabemos os valores das ações



Valor das Ações		
		
0,50	0,75	0,25

Primeiro o Médico Escolhe Aleatoriamente o Remédio

A recompense é igual a 1 se o resultado é positivo do contrário é 0



1	✓		
2	✗		

Vamos estimar o valor da Ação usando o Método Sample Weigh (Média Ponderada)

$$Q_t(a) = \frac{\text{soma das recompensas quando a ação } a \text{ é tomada}}{\text{número de vezes em que a ação } a \text{ é tomada}}$$

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t}$$

A recompensa é igual a 1 se o resultado é positivo do contrário é 0



1	✓		
2	✗		
$Q_t(a)$?		

Aplique a formula para calcular os dois primeiros testes




$$Q_t(a) = \frac{\text{soma das recompensas quando a ação } a \text{ é tomada}}{\text{número de vezes em que a ação } a \text{ é tomada}}$$

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

A recompensa é igual a 1 se o resultado é positivo do contrário é 0

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t = 1}$$



			
1	✓		
2	✗		
3		✓	
4			✗
5		✗	
6			✓
7	✗		
8			✓
9		✓	
10			✗
11		✓	
12	✗		
$Q_t(a)$?	?	?

A recompensa é igual a 1 se o resultado é positivo do contrário é 0

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t = 1}$$



1	✓		
2	✗		
3		✓	
4			✗
5		✗	
6			✓
7	✗		
8			✓
9		✓	
10			✗
11		✓	
12	✗		
$Q_t(a)$	0,25	0,75	0,50

A recompensa é igual a 1 se o resultado é positivo do contrário é 0

