Giới thiệu chung về khám phá (Exploration) trong Reinforcement Learning với Function Approximation

Khám phá (exploration) là thành phần then chốt trong Reinforcement Learning (RL), giúp agent thu thập thông tin về môi trường trước khi chọn chính sách tối ưu. Khi không gian trạng thái hoặc hành động lớn/continuous, ta thường dùng function approximation để xấp xỉ hàm giá trị hoặc Q-function, nhưng điều này cũng ảnh hưởng đến cơ chế khám phá. Slide "Exploration under Function Approximation" trình bày hai phương pháp phổ biến: **giá trị khởi tạo lạc quan (optimistic initial values)** và **ε-greedy**. Dưới đây là phân tích chi tiết từng khái niệm, cách áp dụng, ưu/nhược điểm và ví dụ minh họa kèm dẫn chứng từ tài liệu.

---

1. Optimistic Initial Values trong Function Approximation

1.1. Khái niệm Optimistic Initial Values (OIV)

- **Định nghĩa cơ bản**: Trong tabular RL, optimistic initial values nghĩa là khởi tạo các giá trị Q(s,a) hoặc V(s) ở mức cao hơn giá trị thật kỳ vọng. Mục tiêu là khuyến khích agent thử từng hành động ít nhất một lần vì agent tin tưởng ban đầu rằng mỗi hành động có tiềm năng thu được phần thưởng lớn hơn thực tế. Khi đã thử, nếu giá trị thực kém hơn, bản ước lượng sẽ bị giảm xuống, nhưng những hành động chưa thử luôn duy trì giá trị khởi tạo cao, tạo tính hệ thống trong khám phá ban đầu medium.com stanford.edu .
- **Áp dụng Tabular**: Dễ thực hiện do mỗi state-action có giá trị riêng biệt. Ví dụ nếu biết reward tối đa có thể đạt, khởi Q(s,a) = giá trị này, agent sẽ ưu tiên thử các hành động chưa thử ở mỗi state. Khi Q(s,a) cập nhật qua trải nghiệm, giá trị giảm dần nếu thực sự kém.

1.2. Áp dụng OIV với Function Approximation

- **Khó khăn chung**: Với function approximation, ta lưu trọng số w và feature $\phi(s)$ hoặc $\phi(s,a)$. Giá trị ước lượng $\hat{Q}(s,a;w)$ là hàm của w và feature. Để khởi tạo lạc quan, cần khởi w sao cho $\hat{Q}(s,a)$ "đều" cao hơn giá trị thật cho mọi (s,a) chưa biết. Việc này tùy thuộc cấu trúc feature và hàm xấp xỉ (linear hay phi tuyến).
- **Trường hợp dễ**:
  - Với feature nhị phân (binary features) trong linear approximation: nếu mỗi feature $\phi_i(s,a) \in \{0,1\}$, ta có thể khởi mỗi trọng số w_i = một giá trị lớn bằng hoặc lớn hơn phần thưởng tối đa có thể đạt. Khi đó với mỗi (s,a) có ít nhất một feature active, tổng $w^\top \phi(s,a)$ sẽ lớn, khiến $\hat{Q}(s,a)$ lạc quan. Ví dụ tile coding hoặc coarse coding với feature binary: khởi mọi weight bằng giá trị lớn nhất kỳ vọng return, do vậy bất kỳ state-action nào chứa feature active đều cho giá trị khởi tạo cao medium.com .
  - Với linear trên feature liên tục đã normalized: có thể khởi weights bằng giá trị sao cho $w^\top \phi(s,a)$ lớn ví dụ chọn bias term cao.
- **Trường hợp phức tạp**:

    - **Non-linear function approximation (neural networks)**: Mối quan hệ giữa trọng số và giá trị ước lượng phi tuyến thường phức tạp và không dễ điều khiển. Ví dụ khởi bias layer cuối ở giá trị cao có thể bị normalization (batch norm) hoặc activation phi tuyến làm giảm hiệu quả, hoặc cập nhật gradient sẽ nhanh chóng "học mất" tính lạc quan ban đầu. Do đó OIV với mạng nơ-ron đòi hỏi cân nhắc kỹ: có thể khởi bias đầu ra cao, nhưng dễ gây instability hoặc bị undo bởi cơ chế training (ví dụ layer normalization). Nhiều nghiên cứu thử hack bias, nhưng không bền trong deep RL reddit.com .
    - **Feature generalization**: Nếu feature generalize mạnh, cập nhật một (s,a) có thể làm thay đổi giá trị ước lượng cho nhiều state-action khác, khiến giá trị lạc quan ban đầu bị hạ cho các state-action chưa từng thử. Ví dụ extreme: chỉ có một feature luôn =1 cho mọi state-action; khởi w để $\hat{Q}$ cao, nhưng khi agent trải nghiệm một (s,a) thực tế kém, cập nhật weight giảm, và do feature chung, mọi state-action đều giảm, mất đi tính lạc quan cho những state-action khác. Kết quả: không tạo ra khám phá hệ thống như tabular ai.stackexchange.com .
    - **Localized updates**: Nếu function approximation hỗ trợ cập nhật cục bộ – tức mỗi trải nghiệm chỉ ảnh hưởng giá trị những state-action gần hoặc chứa cùng feature – có thể giữ tính lạc quan cho phần còn lại. Ví dụ tile coding: mỗi trạng thái-action được đại diện bởi một bộ feature sparse; cập nhật qua TD sẽ giảm weight chỉ tại các tile active, không làm giảm giá trị của state-action không chia sẻ tile, do đó vẫn giữ lạc quan cho chúng. Neural network đôi khi có khả tự học feature cục bộ, nhưng thường generalize mạnh, khiến cập nhật làm thay đổi rộng rãi. Vì vậy tile coding hoặc coarse coding với vùng chồng lấn vừa phải có thể hỗ trợ OIV hiệu quả hơn deep nets thô. medium.com .
- **Thiết kế OIV với function approximation**:
    - **Linear binary features**: Khởi weight cao; đảm bảo feature active tồn tại cho mọi state-action.
    - **Tile coding**: Khởi weight tile ở giá trị lạc quan; do sparse và cập nhật cục bộ, state-action chưa gặp vẫn giữ lạc quan cho đến khi được thử.
    - **Coarse coding**: Khởi weight receptive fields lạc quan, nếu vùng chồng lấn đủ cục bộ.
    - **Neural networks**: Khó hơn; có thể thử khởi bias đầu ra cao, hoặc dùng thêm bonus intrinsic reward thay vì OIV trực tiếp.
- **Giới hạn**:
    - Khi feature generalize quá rộng, OIV không đảm bảo khám phá hệ thống.
    - Khi feature quá cục bộ (ví dụ không overlap), OIV giống tabular, nhưng cần nhiều parameter/trải nghiệm.
    - Non-linear: OIV có thể nhanh chóng bị "quên" qua cập nhật, không hiệu quả lâu dài.
- **Ví dụ minh họa**:
    - Giả sử trạng thái 1D chia thành 4 vùng tile coding, mỗi tile binary. Khởi weight mỗi tile = R_max (ví dụ 1.0). Khi agent ở state s thử action a, cập nhật giảm weight của các tile active cho (s,a). Các state-action khác dùng tile khác hoặc chỉ share một phần, vẫn giữ giá trị lạc quan, nên agent sẽ thử các vùng khác tiếp tục.
    - Nếu chỉ một feature mọi nơi active, cập nhật đầu tiên làm giảm cho mọi state-action, mất lạc quan, khám phá ngẫu nhiên không hệ thống.

1.3. Hệ quả đối với quá trình khám phá

- **Khám phá hệ thống (systematic exploration)**: Nhờ OIV cục bộ, agent ưu tiên thử các state-action chưa gặp. Khi thử xong, weight giảm, chuyển khám phá sang phần khác.
- **Khám phá hiệu quả vs sample efficiency**: OIV có thể cải thiện tốc độ khám phá ban đầu khi feature hỗ trợ cục bộ. Tuy nhiên, khi môi trường phức tạp, cần kết hợp thêm các phương pháp khác (intrinsic motivation, bonus reward, UCB, Thompson sampling) để khám phá hiệu quả hơn thay vì chỉ OIV.
- **Thời gian hiệu lực**: OIV chỉ hữu ích cho giai đoạn ban đầu; khi agent đã trải nghiệm đa dạng, OIV hết tác dụng. Nếu môi trường non-stationary, OIV không giúp tái khám phá sau khi reward phân bố thay đổi.
- **Khi không gian lớn**: Với feature cục bộ, OIV hỗ trợ khám phá từng vùng; nhưng nếu feature không phủ hết không gian hoặc phân phối trải nghiệm skewed, có thể vẫn bỏ sót vùng quan trọng. Cần đảm bảo feature/tilings bao phủ mọi vùng.

---

2. Epsilon-Greedy với Function Approximation

2.1. Cơ chế ε-greedy

- **Định nghĩa**: Với giá trị ước lượng $\hat{Q}(s,a)$, ε-greedy chọn hành động: với xác suất ε, chọn ngẫu nhiên một hành động (exploration); với xác suất 1−ε, chọn hành động greedy (max $\hat{Q}(s,a)$) (exploitation).
- **Áp dụng chung**: Không phụ thuộc cách tính $\hat{Q}(s,a)$; chỉ cần có ước lượng giá trị cho mọi action tại state hiện tại, nên dễ áp dụng dù dùng tabular, linear hay phi tuyến.
- **Khám phá ngẫu nhiên**: ε-greedy dựa trên randomness để khám phá, không có chỉ dẫn hệ thống dựa trên độ tin cậy hoặc độ bất định của ước lượng. Điều này dẫn đến khả năng bỏ sót các action tốt nếu ước lượng ban đầu kém và random không may không chọn, hoặc chọn quá ít.

2.2. Áp dụng trong Function Approximation

- **Linear hoặc Neural**: Ta vẫn dùng ε-greedy: tại mỗi bước, tính $\hat{Q}(s,a;w)$ cho mọi a; chọn theo ε-greedy. Không cần thay đổi gì đặc biệt cho function approximation.
- **Khám phá không có định hướng (undirected exploration)**: ε-greedy không tận dụng thông tin về độ bất định (uncertainty) của ước lượng; luôn chọn random với xác suất cố định hoặc decay theo schedule.
- **Decay ε**: Thường bắt đầu với ε cao (ví dụ 1.0) để khám phá rộng, sau đó giảm dần đến giá trị nhỏ (ví dụ 0.1 hoặc 0.01) nhằm tập trung exploitation sau khi ước lượng tốt hơn. Cách decay phải cân bằng: nếu giảm quá nhanh, có thể chưa khám phá đủ; giảm quá chậm, quá nhiều random khiến học chậm.
- **Non-stationary và sticky ε**: Với môi trường thay đổi, cần tái tăng ε định kỳ để khám phá lại.

- **Tương tác với OIV**: OIV cung cấp khởi đầu lạc quan cho giá trị ước lượng; kết hợp ε-greedy giúp agent khám phá ban đầu đa dạng hơn. Tuy nhiên OIV có thể giảm reliance vào ε-greedy ban đầu, nhưng về lâu dài ε-greedy vẫn cần để khám phá nhỏ lẻ hoặc escape local optimum.

2.3. Ưu và nhược điểm

- **Ưu điểm**:
  - Dễ triển khai, không cần tính toán thêm; áp dụng với mọi function approximation.
  - Có thể điều chỉnh ε schedule linh hoạt.
- **Nhược điểm**:
  - **Không định hướng**: Khám phá hoàn toàn ngẫu nhiên, không tập trung vào khu vực có nhiều tiềm năng hoặc nơi ước lượng chưa chắc chắn.
  - **Hiệu quả thấp trong không gian lớn**: Nếu số action lớn hoặc state-action space rộng, random exploration hiếm khi chọn đúng action quan trọng.
  - **Không tận dụng uncertainty**: Không dùng thông tin về độ tin cậy của $\hat{Q}$; các phương pháp UCB, Bayesian, Thompson sampling, hoặc intrinsic reward (curiosity) có thể khám phá hiệu quả hơn.
  - **Chính sách chattering**: Với function approximation, ε-greedy có thể gây dao động policy (chattering) do ước lượng thay đổi và random. Một số nghiên cứu chỉ ra Q-learning hoặc Sarsa với ε-greedy phi tuyến có thể hội tụ đến các attractors không mong muốn openreview.net .
- **Khi nên dùng**: Khi môi trường tương đối đơn giản hoặc khi không có khả năng tính toán độ bất định; làm baseline nhanh chóng; khi muốn kết hợp OIV và ε-greedy để khám phá ban đầu; trong Deep RL (ví dụ DQN) thường vẫn dùng ε-greedy kết hợp replay buffer, target network để ổn định.

---

3. So sánh OIV và ε-greedy, và các phương pháp khám phá khác

3.1. OIV vs ε-greedy

- **OIV**: Tạo ưu thế ban đầu cho mọi action chưa thử, khuyến khích khám phá hệ thống cho đến khi ước lượng giảm. Hoạt động hiệu quả khi feature approximation cục bộ. Tuy nhiên chỉ áp dụng giai đoạn đầu, và phụ thuộc cấu trúc feature.
- **ε-greedy**: Khám phá ngẫu nhiên liên tục theo xác suất ε. Không định hướng, không phụ thuộc feature, nhưng dễ bỏ sót action quan trọng nếu ε thấp hoặc random không may.
- **Kết hợp**: Thường dùng OIV để khởi đầu lạc quan, sau đó dùng ε-greedy với decay ε. Điều này giúp agent trong giai đoạn đầu thử nhiều action mới do OIV, sau đó vẫn dùng ε-greedy random khám phá thêm.

3.2. Nhược điểm chung và hướng mở rộng

- **Khám phá thiếu định hướng**: Cả OIV và ε-greedy không tận dụng explicit uncertainty hoặc novelty. Trong function approximation, nhất là deep RL, có thể dùng:
  - **Upper Confidence Bound (UCB)** dựa trên ước lượng độ bất định (ví dụ sử dụng phương sai hoặc mạng huyền nội bộ) để chọn action có tiềm năng cao chưa chắc chắn.
  - **Thompson Sampling**: Mẫu từ phân phối posterior của Q-function; chọn action theo sample để cân bằng khám phá-khai thác.
  - **Intrinsic Motivation / Curiosity**: Tạo bonus reward cho trạng thái mới, nghịch thường hoặc high prediction error, khuyến khích khám phá vùng ít thăm.
  - **Count-based exploration**: Với MDP lớn, dùng approximated counts qua hash hoặc density model để khuyến khích đến trạng thái ít thăm.
  - **Optimism under Uncertainty**: Tính toán bound trên giá trị chưa biết, chọn action đạt upper bound cao. Với function approximation, ước tính bound phức tạp, nhưng có nghiên cứu dùng Bayesian linear models, kernel methods để approximate UCB camallen.net .
  - **Bootstrapped DQN**: Dùng mạng bootstrap để ước tính uncertainty và chọn policy ngẫu nhiên từ ensemble.
- **Khó khăn lý thuyết**: Nhiều nghiên cứu chỉ ra khó khăn khám phá sample-efficient trong môi trường lớn với function approximation nếu không có cấu trúc đặc biệt nanjiang.cs.illinois.edu .
- **Khi áp dụng slide**: Slide chỉ đề cập OIV và ε-greedy làm phương pháp cơ bản, phù hợp trong nhiều bài toán đơn giản hoặc khi feature cục bộ. Nhưng để giải bài toán RL phức tạp, cần xem xét phương pháp nâng cao.

---

4. Kết luận và khuyến nghị

- **Optimistic Initial Values**:
  - Hiệu quả nếu feature approximation cho phép cập nhật cục bộ (tile coding, coarse coding) để duy trì lạc quan cho các state-action chưa thử.
  - Với function approximation phi tuyến (deep nets), OIV khó duy trì và có thể bị "dịch chuyển" nhanh do generalization. Cần thận trọng hoặc dùng các hình thức intrinsic reward hoặc khai triển mô hình uncertainty khác.
  - Chỉ hỗ trợ giai đoạn đầu, khi agent chưa trải nghiệm nhiều.
- **ε-greedy**:
  - Dễ triển khai với mọi function approximation; tuy nhiên không định hướng, có thể kém hiệu quả trong không gian lớn.
  - Nên dùng decay ε hợp lý (GLIE: decaying ε đảm bảo đủ khám phá nhưng tập trung sau). Với môi trường non-stationary, cần điều chỉnh lại ε hoặc dùng adaptive exploration.
- **Kết hợp**:
  - Thường khởi OIV cho giai đoạn đầu nếu feature cục bộ hỗ trợ, sau đó dùng ε-greedy với decay.
  - Dùng ε-greedy xuyên suốt khi không có khả tính toán độ bất định; nếu cần hiệu quả hơn, cân nhắc UCB, Thompson, intrinsic reward.

- **Lời khuyên**:
  - Khi dùng linear approximation với tile coding hoặc coarse coding: thiết kế feature sao cho cập nhật cục bộ, cho phép OIV hiệu quả.
  - Khi dùng neural network: ưu tiên các phương pháp exploration dựa trên uncertainty/intrinsic bonus thay vì OIV trực tiếp.
  - Theo dõi learning curves và phân phối trải nghiệm để điều chỉnh exploration: nếu agent "quên" vùng nào, tăng ε hoặc thêm bonus.
  - Xem xét GLIE (Greedy in the Limit with Infinite Exploration) để đảm bảo hội tụ policy tốt theo lý thuyết.