Gradient strategii

Gradient strategii

- Dotychczas rozpatrywaliśmy metody akcji-wartości.
  - Zakładaliśmy, że agent najpierw szacuję wartość funkcji Q a dopiero później, na podstawie tego oszacowania wybiera odpowiednią akcję, np. za pomocą strategii ϵ-zachłannej.
- Rozpatrzmy sytuację w której chcemy aby agent uczył się optymalnych zachowań bezpośrednio, z pominięciem szacowania funkcji wartości.

Gradient strategii

- Dotychczas rozpatrywaliśmy metody akcji-wartości.
  - Zakładaliśmy, że agent najpierw szacuję wartość funkcji Q a dopiero później, na podstawie tego oszacowania wybiera odpowiednią akcję, np. za pomocą strategii ϵ-zachłannej.
- Rozpatrzmy sytuację w której chcemy aby agent uczył się optymalnych zachowań bezpośrednio, z pominięciem szacowania funkcji wartości.
- Umożliwia to uczenie agenta nawet w sytuacji w której:
  - Optymalna strategia jest **stochastyczna**
  - Przestrzeń akcji jest **ciągła**

Przykład

- https://www.youtube.com/watch?v=_wjPEMkPJkM

Gradient strategii

- Ponadto w wielu przypadkach okazuje się, że aproksymacja strategii $\pi(a|s, w)$ jest po prostu dużo efektywniejsza niż aproksymacja funkcji wartości $\mathrm{Q}(s, a, \theta)$
  - W przypadku złożonych problemów, gdzie przestrzeń akcji jest duża, metoda gradientu strategii zbiega znacznie szybciej.

Strategia Softmax

- W przypadku gdy przestrzeń akcji a jest dyskretna intuicyjnym sposobem definiowania strategii jest:

$$\pi(a'|s) = \frac{e^{h(s,a',w)}}{\sum_{a \in A(s)} e^{h(s,a,w)}}$$

gdzie miara **preferencji** $h(s, a, w) \in \mathbb{R}$ jest parametryzacją każdej dopuszczalnej pary stan akcja.

- Funkcja h może przyjmować różne postaci:
  - W najprostszym przypadku może to być po prostu kombinacja liniowa wektora cech reprezentującego parę stan-akcja:
  $$h(s, a, w) = x(s, a)^\top w$$
  - albo dowolna inna bardziej złożona metoda aproksymacji (np. **sieć neuronowa**).

Optymalizacja strategii

- Wiedząc jak możemy parametryzować strategię $\pi(a|s)$ w jaki sposób powinniśmy wyznaczyć jej optymalną wartość?
- Naszym celem jest znalezienie wag w, które zapewnią najlepsze oszacowanie $\pi(a|s, w)$.
- Aby tego dokonać musimy wprowadzić miarę jakości oszacowania $J(w)$.
  - Dla przypadków epizodycznych: $J(w) = V_{\pi_w}(s_0)$ (wartość stanu początkowego)
  - Dla przypadków ciągłych: $J(w) = \sum_s d_{\pi_w}(s) V_{\pi_w}(s)$ (przeciętna wartość stanu)
  - Lub: $J(w) = \sum_s d_{\pi_w}(s) \sum_a \pi_w(s|a) R(s, a, s')$ (przeciętna nagroda)

Optymalizacja strategii

- Wtedy jedyne co nam pozostaje to zastosowanie odpowiedniego znajdującego lokalne maksimum funkcji $J(w)$:

$$\Delta w = \alpha \nabla_w J(w)$$

- Wyrażanie $\nabla_w J(w)$ nazywamy **gradientem strategii**:

$$\nabla_w J(w) = \begin{pmatrix} \frac{\partial J(w)}{\partial w_1} \\ \vdots \\ \frac{\partial J(w)}{\partial w_n} \end{pmatrix}$$

:

Twierdzenie o Gradiencie Strategii

- Jak to zrobić?
- Wiemy, że:

$$\nabla_w J(w) = \nabla_w v_{\pi_w}(s) = \nabla_w \left[\sum_{a \in A} \pi(a|s) q_\pi(s,a) \right]$$

Twierdzenie o Gradiencie Strategii

- Zakładając, że funkcja strategii jest różniczkowalna i znamy jej gradient $\nabla_w \pi_w(a|s)$ możemy wyznaczyć:

$$\nabla_w \pi_w(a|s) = \pi_w(a|s) \frac{\nabla_w \pi_w(a|s)}{\pi_w(a|s)} = \pi_w(a|s) \nabla_w \log \pi_w(a|s)$$

- I wtedy definiujemy funkcję wyniku (score function) jako:

$$\nabla_w \log \pi_w(a|s)$$

Twierdzenie o Gradiencie Strategii

- W przypadku jednego kroku w przód i stanu $s \sim d_{\pi_w}(s)$ funkcja celu $J(w)$ wygląda nastepująco:

$$J(w) = \sum_{s \in S} d_{\pi_w}(s) \sum_{a \in A} \pi_w(a|s) r$$

- A jej gradient:

$$\nabla_w J(w) = \sum_{s \in S} d_{\pi_w}(s) \sum_{a \in S} \pi_w(a|s) \nabla_w \log \pi_w(a|s) r$$
$$= \mathrm{E}_{\pi_w} [\nabla_w \log \pi_w(a|s) r]$$

Twierdzenie o Gradiencie Strategii

- W przypadku jednego kroku w przód i stanu $s \sim d_{\pi_w}(s)$ funkcja celu $J(w)$ wygląda nastepująco:

$$J(w) = \sum_{s \in S} d_{\pi_w}(s) \sum_{a \in A} \pi_w(a|s) r$$

- A jej gradient:

$$\nabla_w J(w) = \sum_{s \in S} d_{\pi_w}(s) \sum_{a \in S} \pi_w(a|s) \nabla_w \log \pi_w(a|s) r$$
$$= \mathrm{E}_{\pi_w} [\nabla_w \log \pi_w(a|s) r]$$

- Co jeśli do uaktualnienia chcemy wykorzystać skumulowaną nagrodę R_t? Albo funkcję wartości $Q_{\pi_w}(s, a)$?

Twierdzenie o Gradiencie Strategii

- Na mocy **twierdzenia o gradiencie strategii** okazuje się, że wielkość gradientu $\nabla_w J(w)$ jest proporcjonalna do wyrażenia

$$\nabla_w J(w) \propto \mathrm{E}_{\pi_w}\left[\nabla_w \log \pi_w(a|s) Q_{\pi_w}(s,a)\right]$$

- Lub:

$$\nabla_w J(w) \propto \mathrm{E}_{\pi_w}\left[\nabla_w \log \pi_w(a|s) R_t\right]$$

Twierdzenie o Gradiencie Strategii

Twierdzenie :

Dla dowolnej różniczkowalnej strategii $\pi_w(a|s)$ i dowolnej funkcji celu J gradient strategii jest równy:

$$\nabla_w J(w) = \mathbb{E}_{\pi_w}\left[\nabla_w \log \pi_w(a|s)\, Q_{\pi_w}(s,a)\right]$$

Monte Carlo Policy Gradient (REINFORCE)

1. Zainicjuj algorytm wybierając dowolną strategię $\pi(\cdot \mid \cdot, w)$ parametryzowaną przez wagi w i krok uaktualnienia $\alpha > 0$.
2. W każdej iteracji k:
   - Wygeneruj epizod na podstawie strategii $\pi(\cdot \mid \cdot, w)$: $\mathrm{s}_0, \mathrm{a}_0, \mathrm{r}_1, \ldots, s_{T-1}, a_{T-1}, r_T$.
   - Przyjmij wartość skumulowanej przyszłej nagrody $R = 0$
   - Dla każdego $\mathrm{t} = \mathrm{T} - 1, \mathrm{T} - 2, \ldots, 1, 0$:
     - Przypisz nową wartość $R = \beta R + r_{t+1}$
     - Uaktualnij wagi w: $w \leftarrow w + \eta \beta^t \nabla_w \log \pi\,(a_t | s_t, w) R$

Monte Carlo Policy Gradient (REINFORCE)

- Algorytm REINFORCE nadal ma jednak zasadniczą wadę. Jako metoda Monte Carlo ma bardzo wysoką wariancję.
- Jak ją obniżyć?

Monte Carlo Policy Gradient (REINFORCE)

- Algorytm REINFORCE nadal ma jednak zasadniczą wadę. Jako metoda Monte Carlo ma bardzo wysoką wariancję.
- Jak ją obniżyć?
- Intuicyjną metodą jest wykorzystanie funkcji bazowej (**baseline function**) $b(s)$.
- Wtedy uaktualnienie wag w wygląda następująco:

$$\mathrm{A}(s,a) = R \ - b(s)$$
$$w \leftarrow w + \eta\beta^t \nabla_w \log \pi\,(a_t|s_t, w)\mathrm{A}(s,a)$$

- Funkcje $\mathrm{A}(s)$ nazywamy funkcją korzyści (**advantage function**)
- Aby nie zaburzać wyników, taka funkcja musi być niezależna od akcji a. Naturalnym kandydatem jest **aproksymata funkcji wartości stanu** $\hat{v}(s,\theta)$.

Monte Carlo Policy Gradient z funkcja bazową (REINFORCE)

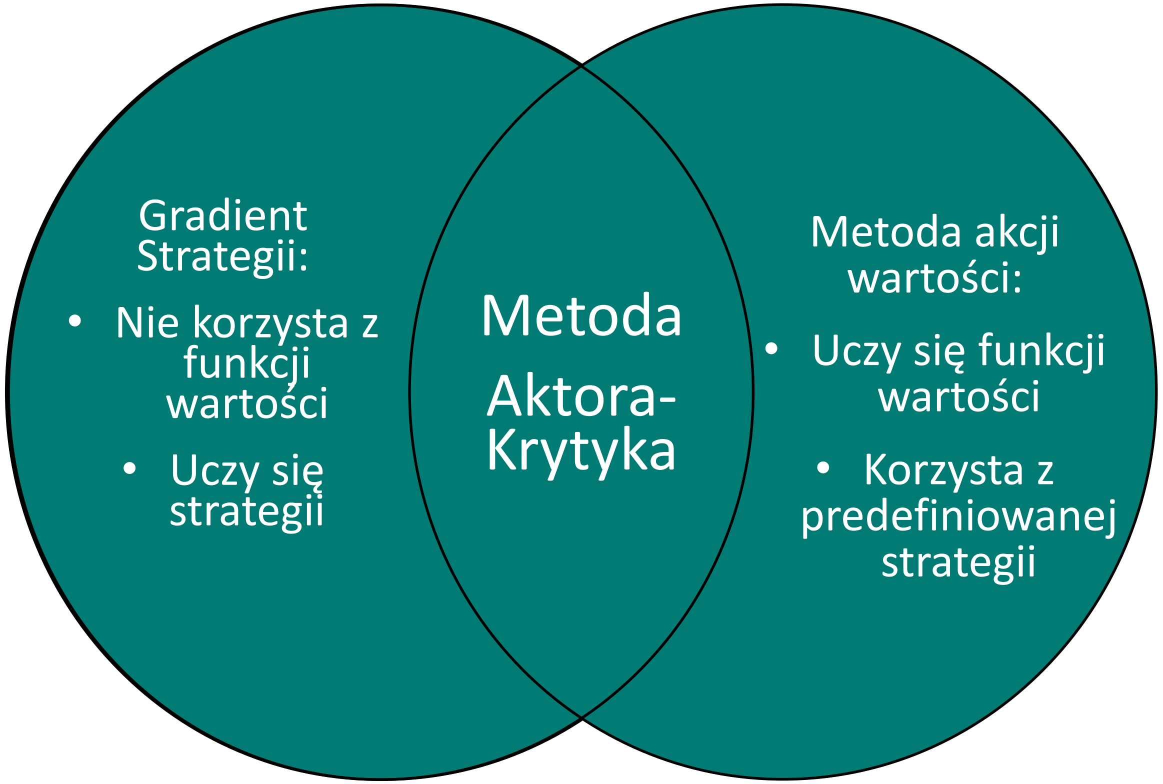
1. Zainicjuj algorytm wybierając dowolną strategię $\pi(\cdot \mid \cdot, w)$ parametryzowaną przez wagi w, aproksymację funkcji wartości $\hat{v}(s, \theta)$ parametryzowaną przez wagi θ i kroki uaktualnienia $\alpha^w, \alpha^\theta > 0$.
2. W każdej iteracji k:
   - Wygeneruj epizod na podstawie strategii $\pi(\cdot \mid \cdot, w)$: $\mathrm{s}_0, \mathrm{a}_0, \mathrm{r}_1, \ldots, s_{T-1}, a_{T-1}, r_T$.
   - Przyjmij wartość skumulowanej przyszłej nagrody $R = 0$
   - Dla każdego $\mathrm{t} = \mathrm{T}-1, \mathrm{T}-2, \ldots, 1, 0$:
     - Przypisz nową wartość $R = \beta R + r_{t+1}$
     - Wyznacz wielkość korzyści $\mathrm{A}(s, a)$: $\mathrm{A}(s, a) = R - \hat{v}(s, \theta)$
     - Uaktualnij wagi θ: $\theta \leftarrow \theta + \alpha^\theta \nabla_\theta \hat{v}(s, \theta) \mathrm{A}(s, a)$
     - Uaktualnij wagi w: $w \leftarrow w + \alpha^w \beta^t \nabla_w \log \pi\,(a_t | s_t, w) \mathrm{A}(s, a)$

Gradient strategii

- Pomimo redukcji wariancji jest to nadal algorytm Monte Carlo.
- Czy jest możliwe wykorzystanie gradientu strategii w przypadku uczenia różnicowego?

Metoda Aktora-Krytyka

- W takim wypadku musimy wprowadzić odpowiednią metodę hybrydową



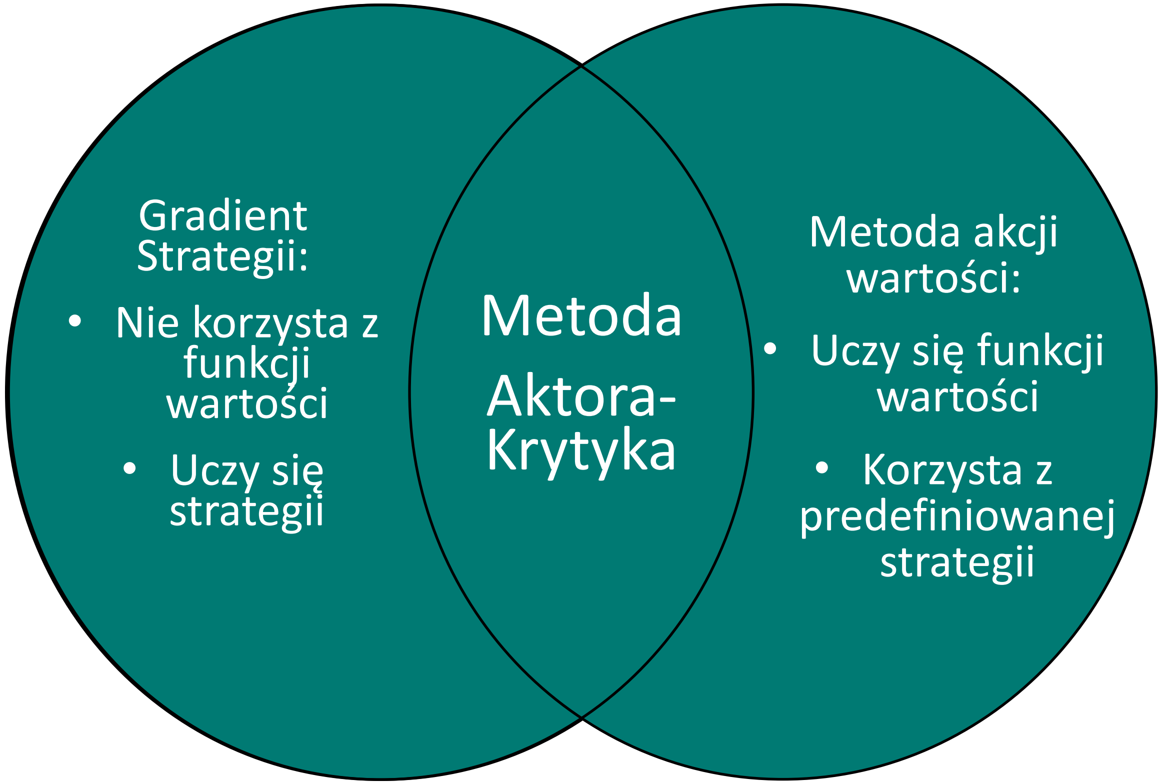
Metoda Aktora-Krytyka

- W takim wypadku musimy wprowadzić odpowiednią metodę hybrydową
- **Krytyk** szacuje wartość funkcji wartości akcji:

$$Q(s, a) \approx \hat{Q}(s, a, \theta)$$

- **Aktor** uaktualnia strategię w kierunku sugerowanym przez krytyka:

$$\Delta w = \eta \nabla_w \log \pi(a|s, w) \hat{Q}(s, a, \theta)$$



Metoda Aktora-Krytyka

1. Zainicjuj algorytm wybierając dowolną strategię $\pi(\,\cdot \mid \cdot, w)$ parametryzowaną przez wagi w, aproksymację funkcji wartości $\hat{Q}(s, a, \theta)$ parametryzowaną przez wagi θ i kroki uaktualnienia $\alpha^w, \alpha^\theta > 0$.
2. W każdej iteracji k:
   - Wyznacz akcję A za pomocą strategii $\pi(\,\cdot \mid \cdot, w)$
   - Podejmij akcję A i zaobserwuj nagrodę R i stan S'
   - Wyznacz akcję A' za pomocą strategii $\pi(\,\cdot \mid \cdot, w)$
   - Wyznacz wielkość przyrostu δ: $\delta = R + \beta\hat{Q}(S', A', \theta) - \hat{Q}(S, A, \theta)$
     - Uaktualnij wagi θ: $\theta \leftarrow \theta + \eta^\theta \nabla_\theta \hat{Q}(S, A, \theta)\delta$
     - Uaktualnij wagi w: $w \leftarrow w + \eta^w \beta^t \nabla_w \log \pi\,(a_t | s_t, w)\hat{Q}(S, A, \theta)$

Metoda Aktora-Krytyka

- W przypadku metody aktora-krytyka podążamy za przybliżonym gradientem:
$$\nabla_w J(w) \approx \mathbb{E}_{\pi_w}\left[\nabla_w \log \pi_w(a|s)\, \hat{Q}\,(s, a, \theta)\right]$$
- W oczywisty sposób wprowadza to obciążenie do modelu, przez co agent może nie znaleźć optymalnego rozwiązania.
- Okazuje się jednak, że spełnienie dwóch prostych warunków gwarantuje że rozwiązanie będzie dokładne:

Metoda Aktora-Krytyka

- Gdy aproksymacja funkcji wartości jest **kompatybilna** ze strategią:
$$\nabla_\theta \hat{Q}\,(s,a,\theta) = \nabla_w \log \pi_w(a|s)$$
- Gdy aproksymacja funkcji wartości minimalizuje błąd średniokwadratowy:
$$MSE = \mathbb{E}_{\pi_w}[\left(\hat{Q}\,(s,a,\theta) - Q_{\pi_w}(s,a)\right)^2]$$
- Wtedy gradient będzie dokładny:
$$\nabla_w J(w) = \mathbb{E}_{\pi_w}\left[\nabla_w \log \pi_w(a|s)\,\hat{Q}\,(s,a,\theta)\right]$$

Na mocy twierdzenia o aproksymacji funkcji kompatybilnych (compatible function approximation theorem).

Metoda Aktora-Krytyka

- Jak ustaliliśmy poprzednio wykorzystanie funkcji korzyści zmniejsza wariancje i znacznie poprawia proces uczenia.
- Jak zastosować ją w przypadku uczenia różnicowego?
- Najprostszy sposób:

$$\mathrm{A}(s,a) = \hat{Q}\,(s,a,\theta) - \hat{v}\,(s,\psi)$$

Metoda Aktora-Krytyka

- Jak ustaliliśmy poprzednio wykorzystanie funkcji korzyści zmniejsza wariancje i znacznie poprawia proces uczenia.
- Jak zastosować ją w przypadku uczenia różnicowego?
- Najprostszy sposób:

$$\mathrm{A}(s,a) = \hat{Q}\,(s,a,\theta)\ - \hat{v}\,(s,\psi)$$

- W takim przypadku konieczna jest jednak aproksymacja trzech funkcji: $\pi(a|s,w), \hat{Q}\,(s,a,\theta)\,, \hat{v}\,(s,\psi)$.
- Okazuje się, że da się ten proces znacznie ułatwić.

Metoda Aktora-Krytyka

- Dla rzeczywistej funkcji wartości v_{π_w} błąd uczenia różnicowego $\delta_{\pi_w} = r + \beta v_{\pi_w}(s') - v_{\pi_w}(s)$ jest nieobciążonym estymatorem funkcji przewagi:

$$\begin{aligned}\mathbb{E}_{\pi_w}\left[\delta_{\pi_w}|s,a\right] &= \mathbb{E}_{\pi_w}\left[r + \beta v_{\pi_w}(s')|s,a\right] - v_{\pi_w}(s)\\ &= Q_{\pi_w}(s,a) - v_{\pi_w}(s)\\ &= \mathrm{A}(s,a)\end{aligned}$$

- Dzięki temu możemy wykorzystać błąd uczenia różnicowego do uaktualniania gradientu strategii i aproksymować tylko jedną funkcję krytyka $\hat{v}\,(s,\psi)$.

Advantage Actor-Critic (A2C)

1. Zainicjuj algorytm wybierając dowolną strategię $\pi(\cdot \mid \cdot, w)$ parametryzowaną przez wagi w, aproksymację funkcji wartości $\hat{v}\ (s, \psi)$ parametryzowaną przez wagi ψ i kroki uaktualnienia $\alpha^w, \alpha^\psi > 0$.
2. W każdej iteracji k:
   - Wyznacz akcję A za pomocą strategii $\pi(\cdot \mid \cdot, w)$
   - Wyznacz wartość $\hat{v}\ (S, \psi)$
   - Podejmij akcję A i zaobserwuj nagrodę R i stan S'
   - Wyznacz wartość $\hat{v}\ (S', \psi)$
   - Wyznacz wielkość korzyści $\mathrm{A}(s, a)$: $\mathrm{A}(s, a) = R + \beta\hat{v}\ (S', \psi) - \hat{v}\ (S, \psi)$
     - Uaktualnij wagi ψ: $\psi \ \leftarrow \psi + \eta^\psi \nabla_\psi \hat{v}\ (S, \psi)\mathrm{A}(s, a)$
     - Uaktualnij wagi w: $w \leftarrow w + \eta^w \beta^t \nabla_w \log \pi\ (a_t | s_t, w)\ \mathrm{A}(s, a)$

Metoda Aktora-Krytyka

- Największą wadą metody Aktora-Krytyka jest to, że jest bardzo wrażliwa na odpowiednią parametryzację.
- Fakt, że musimy kontrolować jednocześnie dwie metody aproksymacji powoduje, że zachowanie modelu może być bardzo niestabilne.
- Niewielkie zmiany aproksymacji funkcji wartości mogą prowadzić do znacznych zmian strategii.
- Kluczowe jest zapewnienie tego żeby zmiany oszacowania strategii $\pi_w(a|s)$ były stabilne w czasie, a ponadto proporcjonalne do zmian oszacowania funkcji wartości $Q_{\pi_w}(s,a)$.

Metoda gradientu naturalnego

- Jak to zrobić?
  - Albo metodą prób i błędów, odpowiednio manipulując parametrami modelu.
  - Albo wykorzystując do uczenia metodę gradientu naturalnego (*Natural Policy Gradient*).

Metoda gradientu naturalnego

- Metoda gradientu naturalnego zakłada, że uaktualnienia strategii agenta nie będą przekraczać pewnej stałej, określonej wartości δ:

$$\max_{w} J_{\pi_w}(\pi_w)$$
$$p.w. \;\; \|\Delta w\|_F < \delta$$

- Co można przedstawić też jako:

$$\pi_{k+1} = \underset{\pi}{\operatorname{argmax}} J_{\pi_w}(\pi_w)$$
$$p.w. \;\; d_{KL}(\pi' || \pi_k) < \delta$$

Gdzie $d_{KL}(\pi' || \pi_k)$ to **dywergencja Kullbacka-Leiblera** pomiędzy π' i π_k.

Metoda gradientu naturalnego

- Po odpowiednich przekształceniach rozwiązanie da się przedstawić jako:

$$\nabla_w^{NAT} \log \pi_w(a|s) = \frac{1}{\gamma} F_w^{-1} \nabla_w \log \pi_w(a|s)$$

Gdzie:

- $F_w = \nabla_w^2 \, d_{kl}(w'|w_k) = \mathrm{E}_{\pi_w}[\nabla_w \log \pi_w(a|s) \, \nabla_w \log \pi_w(a|s)^{\mathrm{T}}]$ to macierz informacji Fishera, która służy do pomiaru odległości pomiędzy dowolnymi parami punktów w przestrzeni strategii.
- $\frac{1}{\gamma} = \sqrt{\frac{2\delta}{\nabla_w J(w)^{\mathrm{T}} F_w^{-1} \nabla_w J(w)}}$

Metoda gradientu naturalnego

- W szczególnym przypadku, na mocy twierdzenia o funkcjach kompatybilnych gradient naturalny upraszcza się do postaci:

$$\nabla_w J(w) = \mathbb{E}_{\pi_w}\left[\nabla_w \log \pi_w(a|s)\, \hat{Q}\,(s, a, \theta)\right]$$
$$= \mathbb{E}_{\pi_w}[\nabla_w \log \pi_w(a|s)\, \nabla_w \log \pi_w(a|s)^{\mathrm{T}}\, \theta\,]$$
$$= F_w \theta$$

- Oznacza to, że wszystkie uaktualnienia strategii odbywają się dokładnie w kierunku zmian aproksymacji funkcji wartości.

Gradient strategii w przypadku ciągłej przestrzeni akcji

- W przypadkach dyskretnych definiowaliśmy aproksymowaną strategię $\pi(a|s)$ jako rozkład prawdopodobieństwa. Analogicznie musimy postąpić w tym wypadku, przyjmując jednak, że rozkład, którego szukamy jest ciągły.
- Musimy jednak przyjąć a priori do jakiej rodziny rozkładów należy strategia $\pi(a|s)$.
- Najpopularniejszym założeniem jest przyjęcie, że $\pi(a|s)$ jest zmienną o rozkładzie normalnym.

Gradient strategii w przypadku ciągłej przestrzeni akcji

- Wtedy $\pi(a|s)$ możemy przedstawić jako:

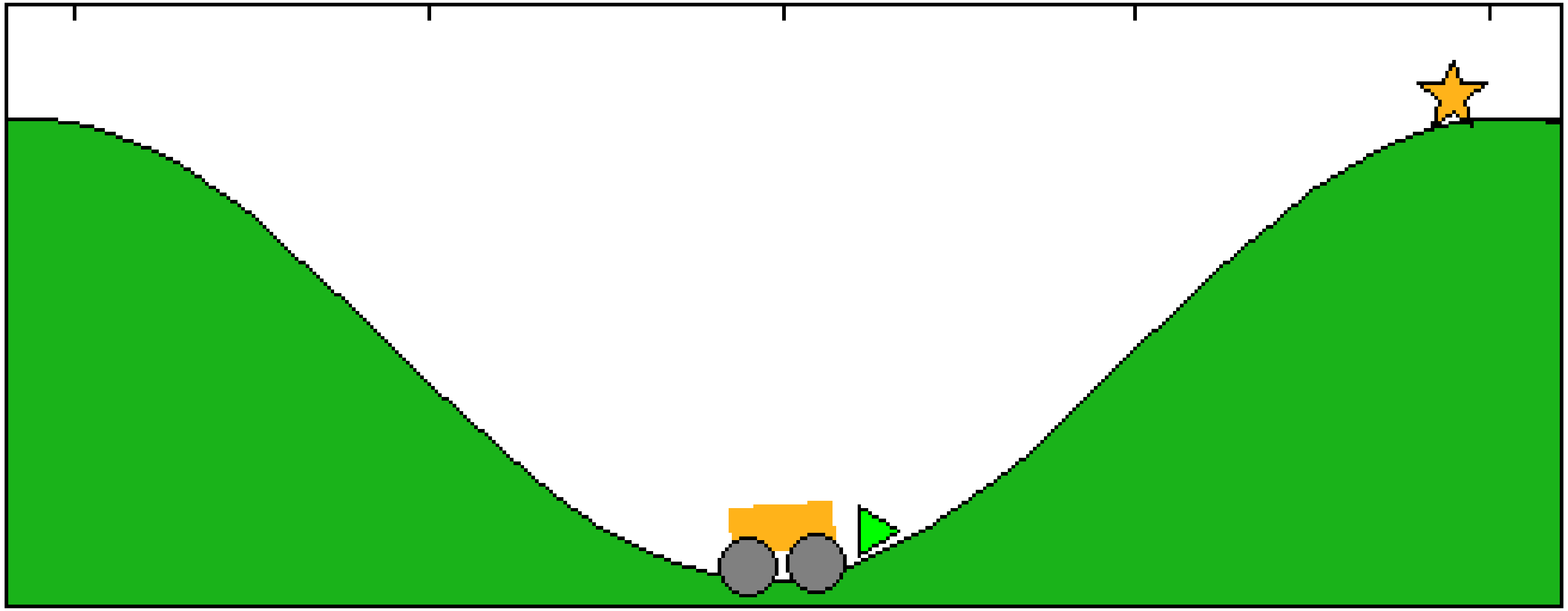
$$\pi(a|s) = \frac{1}{\sigma(s;\theta_\sigma)\sqrt{2\pi}} \exp\left(-\frac{\left(a - \mu(s;\theta_\mu)\right)^2}{2\sigma(s;\theta_\sigma)^2}\right)$$

Gdzie średnia $\mu: \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$ i odchylenie standardowe $\sigma: \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}^+$ są aproksymowanymi funkcjami parametryzowanymi odpowiednio przez $\theta_\mu, \theta_\sigma$

Przykład

Mountain Car

https://en.wikipedia.org/wiki/Mountain_car_problem



Przykład

Cart Pole

https://gym.openai.com/envs/CartPole-v0/