Învățare Automată - Tema 2

Problema explorării în învățarea prin recompensă

Tudor Berariu

Facultatea de Automatică și Calculatoare

Scopul temei

- Scopul temei îl reprezintă înțelegerea problemei explorării în medii cu recompense rare, precum și implementarea unor strategii eficiente de explorare pentru algoritmul SARSA.

- Pentru rezolvarea acestei teme veți:
  1. implementa algoritmul SARSA pentru învățare on-policy;
  2. implementa strategiile de explorare: ϵ-greedy, UCB (upper confidence bound) și Softmax (explorare Boltzmann).

Problema explorării

Explorare versus exploatare
Pentru a descoperi politici eficiente un agent trebuie să îmbine inteligent exploatarea informațiilor pe care le are și explorarea unor stări și acțiuni noi sau despre care știe mai puțin. Mediile în care recompensele sunt rare agravează dilema și au nevoie de strategii bune de explorare a spațiului stărilor.

Algoritmul SARSA

- Agentul învață prin interacțiune cu mediul observând consecințele acțiunilor pe care le ia.

procedure $\mathrm{SARSA}(\langle \mathcal{S}, \mathcal{A}, \gamma \rangle, \pi)$
  for all $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $q(s, a) \leftarrow q_0$ ▷ Valoarea inițială pentru o pereche s, a
    $N(s) \leftarrow 0$
  end for
  for all episodes **do**
    $s \leftarrow$ stare inițială
    alege acțiunea a conform $\pi(s, q)$
    while s nu este stare finală **do**
      $N(s) \leftarrow N(s) + 1$
      execută a și observă recompensa r și noua stare s'
      alege acțiunea a' conform $\pi(s, q)$
      $q(s, a) \leftarrow q(s, a) + \alpha\left(r + \gamma q(s', a') - q(s, a)\right)$
      $s \leftarrow s'$
      $a \leftarrow a'$
    end while
  end for
end procedure

Algoritmul SARSA

- Agentul învață prin interacțiune cu mediul observând consecințele acțiunilor pe care le ia.
- Valorile q sunt ajustate prin diferențe temporale.

procedure $\mathrm{SARSA}(\langle \mathcal{S}, \mathcal{A}, \gamma \rangle, \pi)$
  for all $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $q(s, a) \leftarrow q_0$ ▷ Valoarea inițială pentru o pereche s, a
    $N(s) \leftarrow 0$
  end for
  for all episodes **do**
    $s \leftarrow$ stare inițială
    alege acțiunea a conform $\pi(s, q)$
    while s nu este stare finală **do**
      $N(s) \leftarrow N(s) + 1$
      execută a și observă recompensa r și noua stare s'
      alege acțiunea a' conform $\pi(s, q)$
      $q(s, a) \leftarrow q(s, a) + \alpha\left(r + \gamma q(s', a') - q(s, a)\right)$
      $s \leftarrow s'$
      $a \leftarrow a'$
    end while
  end for
end procedure

Algoritmul SARSA

- Agentul învață prin interacțiune cu mediul observând consecințele acțiunilor pe care le ia.
- Valorile q sunt ajustate prin diferențe temporale.
- Spre deosebire de algoritmul Q-Learning, în SARSA învățarea este on-policy (politica π cu care se iau acțiuni în mediu este aceeași cu cea ale cărei valori q sunt învățate)

procedure $\mathrm{SARSA}(\langle \mathcal{S}, \mathcal{A}, \gamma \rangle, \pi)$
  for all $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $q(s, a) \leftarrow q_0$ ▷ Valoarea inițială pentru o pereche s, a
    $N(s) \leftarrow 0$
  end for
  for all episodes **do**
    $s \leftarrow$ stare inițială
    alege acțiunea a conform $\pi(s, q)$
    while s nu este stare finală **do**
      $N(s) \leftarrow N(s) + 1$
      execută a și observă recompensa r și noua stare s'
      alege acțiunea a' conform $\pi(s, q)$
      $q(s, a) \leftarrow q(s, a) + \alpha\left(r + \gamma q(s', a') - q(s, a)\right)$
      $s \leftarrow s'$
      $a \leftarrow a'$
    end while
  end for
end procedure

Algoritmul SARSA

- Agentul învață prin interacțiune cu mediul observând consecințele acțiunilor pe care le ia.
- Valorile q sunt ajustate prin diferențe temporale.
- Spre deosebire de algoritmul Q-Learning, în SARSA învățarea este on-policy (politica π cu care se iau acțiuni în mediu este aceeași cu cea ale cărei valori q sunt învățate)
- Multe strategii de explorare se bazează pe numărul de vizite.

procedure $\mathrm{SARSA}(\langle \mathcal{S}, \mathcal{A}, \gamma \rangle, \pi)$
  for all $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $q(s, a) \leftarrow q_0$ ▷ Valoarea inițială pentru o pereche s, a
    $N(s) \leftarrow 0$
  end for
  for all episodes **do**
    $s \leftarrow$ stare inițială
    alege acțiunea a conform $\pi(s, q)$
    while s nu este stare finală **do**
      $N(s) \leftarrow N(s) + 1$
      execută a și observă recompensa r și noua stare s'
      alege acțiunea a' conform $\pi(s, q)$
      $q(s, a) \leftarrow q(s, a) + \alpha\left(r + \gamma q(s', a') - q(s, a)\right)$
      $s \leftarrow s'$
      $a \leftarrow a'$
    end while
  end for
end procedure

Strategii de explorare: ϵ-greedy

- O poltică ϵ-greedy este una care alege în $1-\epsilon$ din cazuri cea mai bună acțiune, iar în restul cazurilor alege acțiunea aleator (uniform).
- O strategie practică de a reduce explorarea cu timpul este aceea de a varia ϵ invers porporțional cu numărul de vizite în starea respectivă $N(s)$ (c este o constantă):

$$\epsilon(s) = \frac{c}{N(s)}$$

procedure ϵ-GREEDY(s, q, ϵ)

$\mathcal{A}^* \leftarrow \{a \mid q(s,a) = \max_{a'} q(s,a')\}$

$$p(s,a,q) \leftarrow \begin{cases} \frac{\epsilon(s)}{|\mathcal{A}|} + \frac{1-\epsilon(s)}{|\mathcal{A}^*|} & , a \in \mathcal{A}^* \\ \frac{\epsilon(s)}{|\mathcal{A}|} & , a \notin \mathcal{A}^* \end{cases}$$

return $a \sim p(s, \cdot, q)$

end procedure

Strategii de explorare: Softmax (Boltzmann)

- O altă metodă de a defini o politică stocastică este explorarea Boltzmann:

$$p(a \mid s, q) = \frac{e^{\beta(s) q(s,a)}}{\sum_{a'} e^{\beta(s) q(s,a')}}$$

unde

$$\beta(s) = \frac{\log(N(s))}{\max\limits_{a_1, a_2 \in \mathcal{A} \times \mathcal{A}} |q(s, a_1) - q(s, a_2)|}$$

- Explorarea Boltzmann alege cu o probabilitate mai mare o acțune ce promite un câștig mediu mai mare.
- La fel ca în cazul degradării lui ϵ în cazul ϵ-greedy, în explorarea Boltzmann $\beta(s)$ tinde la zero atunci când numărul de vizite tinde la infinit, iar politica devine lacomă.

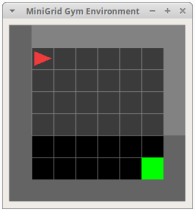
Strategii de explorare: Upper Confidence Bound

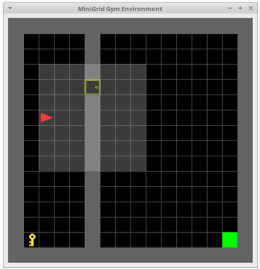
- Strategia UCB adaugă valorilor q un bonus de explorare bazat pe contorizarea dăților în care o acțiune a fost luată într-o stare.
- Termenul de explorare încurajează alegerea acțiunilor luate mai puțin în trecut.
- Bonusul de explorare se degradează în timp.
- Termenul c controlează *nivelul* de explorare.

procedure $\mathrm{UCB}(s, q, c)$
  return $\underset{a \in \mathcal{A}}{\operatorname{argmax}} \left[q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}} \right]$
end procedure

Mediul de test

- Se va testa eficiența celor trei metode de explorare pe medii[1] în care agentul observă parțial mediul și are una din două misiuni:
  - Se deplasează în colțul opus al camerei pentru a primi o recompensă (`Empty`).
  - Caută o cheie, o ridică, deschide o ușă și apoi culege recompensa (`DoorKey`).





[1] `https://github.com/maximecb/gym-minigrid`

Mediul de test

- Se va testa eficiența celor trei metode de explorare pe medii[1] în care agentul observă parțial mediul și are una din două misiuni:
  - Se deplasează în colțul opus al camerei pentru a primi o recompensă (`Empty`).
  - Caută o cheie, o ridică, deschide o ușă și apoi culege recompensa (`DoorKey`).
- Se vor testa strategiile de explorare pe medii de trei dimensiuni:
  - `MiniGrid-Empty-6x6-v0`
  - `MiniGrid-Empty-8x8-v0`
  - `MiniGrid-Empty-16x16-v0`
  - `MiniGrid-DoorKey-6x6-v0`
  - `MiniGrid-DoorKey-8x8-v0`
  - `MiniGrid-DoorKey-16x16-v0`

[1] `https://github.com/maximecb/gym-minigrid`

Cerințe

Cerința 1 Implementați algoritmul SARSA.

Cerința 2 Implementați explorare ϵ-greedy și explorare Boltzmann. Variați rata de învățare, valoarea lui ϵ și cea a constantei c și comparați eficiența celor două strategii pe cele șase medii.

Cerința 3 Încercați o valoare $q_0 > 0$ (o inițializare optimistă) și vedeți impactul asupra eficienței algoritmului.

Cerința 4 Faceți grafice în care să comparați metodele și sumarizați concluziile într-un fișier README. Indicați valorile testate pentru hiper-parametri.

BONUS Implementați metoda de explorare Upper Confidence Bound, căutați o valoare potrivită pentru constanta c și comparați strategia cu cele două implementare anterior.]

Observații

- Faceți grafice cu lungimea episoadelor și câștigul episodic mediu la număr de pași egal (pentru a compara algoritmii în funcție de numărul de interacțiuni cu mediul).
- Lăsați algoritmul să învețe pentru suficient de mulți pași (de ordinul milioanelor de pași pentru hărțile mari).
- Deoarece algoritmii de învățare prin recompensă au în general variantă mare, faceți media a 5-10 experimente identice cu seed-uri diferite.