Apunts del Taller de Nous Usos de la Informàtica

Jordi Vitrià

Universitat de Barcelona

10 de setembre de 2019

UNIVERSITAT DE BARCELONA

Lliçó 7: Aprenentatge interactiu: Explotació versus Exploració

L'escenari

- Imaginem el següent escenari:
  - Suposem que cada vegada que arriba un usuari al nostre *web site* podem oferir-li una opció que triarem entre les n opcions que tenim disponibles en aquell moment (anuncis que pot seleccionar, circuits de vacances, itenerari d'avió, etc.).
  - Un cop oferta l'opció, i en funció de l'acció de l'usuari, rebem una *recompensa* que es pot representar numèricament (que pot ser simplement si ha estat seleccionada o no, el preu que paga o no l'usuari per aquella opció, etc.).
  - Suposem que la recompensa es pot modelar com una funció estacionària de probabilitats que depen de l'acció seleccionada.
- Un objectiu interessant en aquest escenari és maximitzar la recompensa total esperada sobe un periode de temps (per exemple, sobre 1000 accions). A cada una de les accions l'anomenem *jugada*.

El problema

- Aquesta formulació es coneix com el problema dels *n-armed bandits*, o de les màquines escurabutxaques amb múltiples actuadors, per analogia amb les màquines escurabutxaques o *one-armed bandit* (que tenen només un actuador i en les que només hem de decidir si volem continuar jugant).
- En el nostre cas, cada acció té una *recompensa esperada* o *recompensa mitja* donada l'acció que es selecciona. A aquest concepte l'anomenem *valor* de l'acció.
- Si sabéssim el valor de cada acció la solució del problema seria trivial i consistiria en seleccionar sempre l'acció amb més valor.
- En el nostre cas no saben el valor de cada acció, però en podem tenir estimacions.

El problema

- En una aplicació concreta, la tria entre **explotar o explorar** depèn d'una forma complexa dels valors específics de les estimacions, de les incerteses sobre aquestes estimacions i del nombre de jugades que ens queden per fer.
- De totes maneres, veurem que és possible definir polítiques que trobin un bon balanç entre les dues possibilitats i millorin l'explotació pura en la immensa majoria dels casos.

Mètodes d'acció-valor

- Primer hem de veure com estimar els valors de les accions i com usar aquestes estimacions per decidir quina acció seleccionem.
- Anomenem $Q^*(a)$ al valor real de l'acció a, que és desconegut. Recordem que aquest valor és la mitja de les recompenses rebudes quan seleccionem aquesta acció.
- Anomenem $Q_t(a)$ al seu valor estimat després de t jugades.
- La forma natural d'estimar $Q_t(a)$, suposant que a la jugada t-èssima l'acció a ha estat seleccionada k_a vegades abans de t i hem obtingut unes recompenses $r_1 + r_2 + \cdots + r_{k_a}$, és:

$$Q_t(a) = \frac{r_1 + r_2 + \cdots + r_{k_a}}{k_a} \tag{1}$$

Mètodes d'acció-valor: explotació

- En el cas que $k_a = 0$ definim $Q_t(a)$ amb un valor inicial tal com $Q_0(a) = 0$. Llavors, a mesura que $k_a \to \infty$, per la llei dels grans nombres $Q_t(a)$ tendirà a $Q^*(a)$.
- Llavors la regla de selecció més simple després de t jugades és triar l'acció amb el valor estimat més alt, o dit d'una altra manera, seleccionar a la jugada $t+1$ una de les accions voraces, a^*, per la qual $Q_t(a^*) = max_a Q_t(a)$.
- Aquest mètode sempre explota el coneixement actual per maximitzar la recompensa immediata.

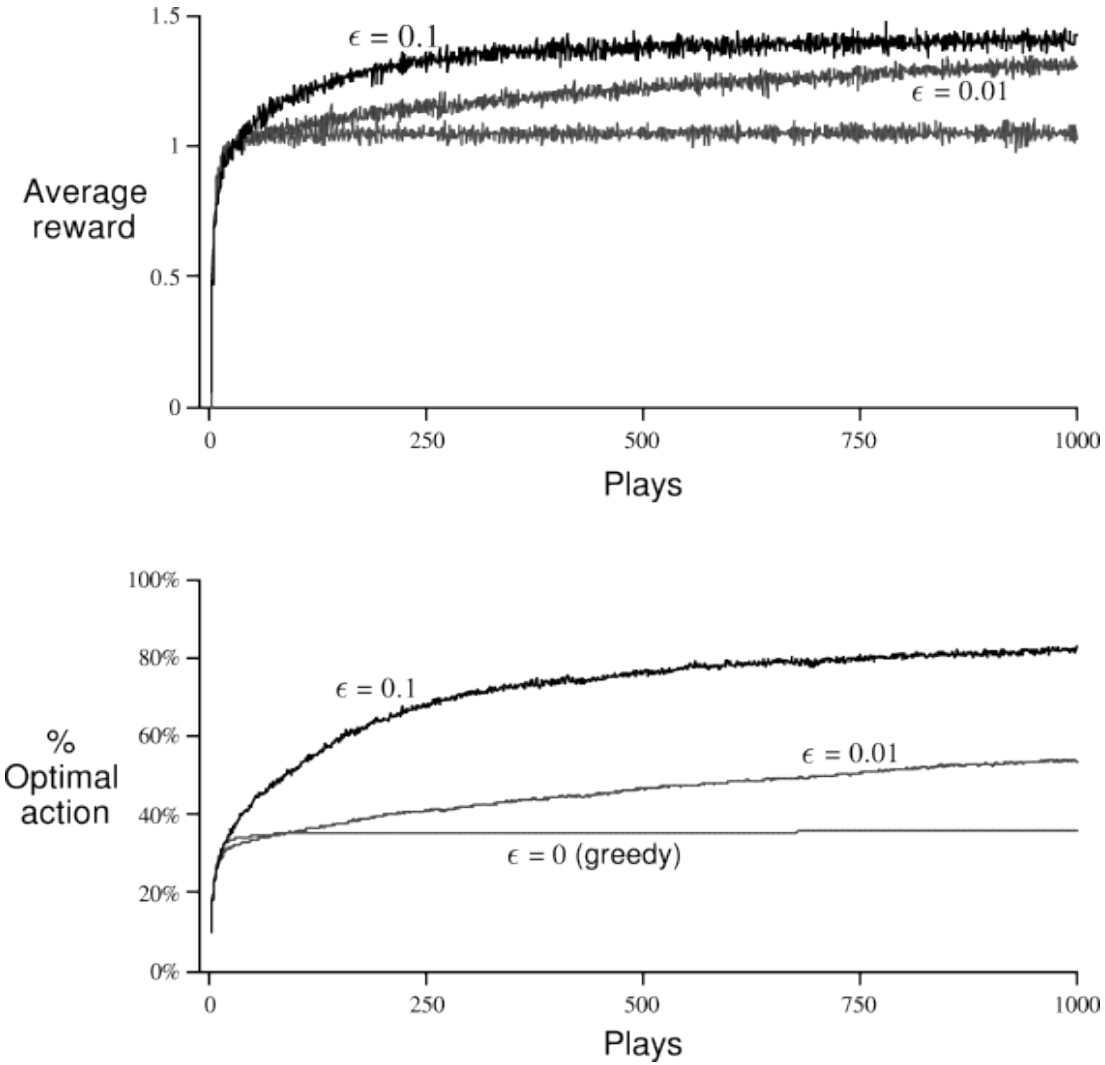
Mètodes d'acció-valor: exploració

- Una possible alternativa és actuar de forma voraç la majoria de vegades però de tant en tant, amb una petita probabilitat ϵ, seleccionar una altra acció de forma uniformement aleatòria, independentment de les estimacions d'acció-valor.
- Aquests darrers mètodes s'anomenen ϵ-voraços.
- Una avantatja d'aquests mètodes és que a mesura que augmenta el nombre de jugades cada una de les accions serà mostrejada un nombre infinit de vegades, garantint que $k_a \rightarrow \infty$ per totes les a i per tant assegurant que $Q_t(a)$ convergeix a $Q^*(a)$.

Mètodes d'acció-valor

- A la gràfica de la pàgina següent podem veure el resultat del següent experiment:
  - S'han generat 2000 problemes de forma aleatòria amb $n = 10$.
  - Els valors de les accions dels 2000 problemes s'han generat seleccionant les $Q^*(a)$ segons una distribució normal amb mitja 0 i variança 1.
  - Per cada acció a que triem la recompensa es genera d'una distribució de probabilitats Gaussiana amb mitja $Q^*(a)$ i variança 1.
  - Si fem la mitja sobre les tasques, podem visualitzar l'eficiència i comportament dels mètodes a mesura que van estimants durant les primeres 1000 jugades.

Mètodes d'acció-valor



Mètode Softmax

- Tot i que els mètodes ϵ-voraços són simples i efectius per balancejar l'exploració i l'explotació, podem tenir un petit defecte: quan exploren trien entre totes les accions per igual. És a dir, té les mateixes probabilitats la pitjor que la més semblant a la millor.
- La sol·lució obvia a aquest problema és triar les accions segons una funció que les distribueixi en funció del seu valor estimat.
- Aquesta estratègia s'anomena *mètode del softmax* i fa servir la distribució de Boltzmann: escull una acció a a la jugada t amb probabilitat

$$\frac{e^{-Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{-Q_t(b)/\tau}}$$

on τ és un paràmetre positiu anomenat *temperatura*.

Mètode Softmax

- Temperatures altes fan que les accions siguin equiprobables i temperatures baixes fan que les probabilitats de selecció de les accions es faci màximament diferent en funció dels valors estimats.
- Quan $\tau \to 0$ tenim el cas voraç pur.

Implementació incremental

- Fins ara hem estimat els valors de les accions com a mitjes de les recompenses observades amb l'expressió (1).
- El problema amb aquesta fòrmula és que els requeriments de memòria i càlcul creixen amb el temps, tot i que no és necessari.
- Donada una acció, sigui Q_k la mitja de les seves primeres k recompenses (que no és el mateix que $Q_k(a)$ o mitja de l'acció a després de la jugada k). Donada aquesta mitja i la recompensa $k+1$, r_{k+1}, podem calcular la mitja de les $k+1$ recompenses com:

Implementació incremental

$$Q_{k+1} = \frac{1}{k+1} \sum_{i=1}^{k+1} r_i$$

Implementació incremental

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{k+1} \sum_{i=1}^{k+1} r_i \\
&= \frac{1}{k+1} \left(r_{k+1} + \sum_{i=1}^{k} r_i \right)
\end{aligned}
$$

Implementació incremental

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{k+1}\sum_{i=1}^{k+1} r_i \\
&= \frac{1}{k+1}\left(r_{k+1} + \sum_{i=1}^{k} r_i\right) \\
&= \frac{1}{k+1}\left(r_{k+1} + kQ_k + Q_k - Q_k\right)
\end{aligned}
$$

Implementació incremental

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{k+1}\sum_{i=1}^{k+1} r_i \\
&= \frac{1}{k+1}\left(r_{k+1} + \sum_{i=1}^{k} r_i\right) \\
&= \frac{1}{k+1}\left(r_{k+1} + kQ_k + Q_k - Q_k\right) \\
&= \frac{1}{k+1}\left(r_{k+1} + (k+1)Q_k - Q_k\right)
\end{aligned}
$$

Implementació incremental

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{k+1}\sum_{i=1}^{k+1} r_i \\
&= \frac{1}{k+1}\left(r_{k+1} + \sum_{i=1}^{k} r_i\right) \\
&= \frac{1}{k+1}\left(r_{k+1} + kQ_k + Q_k - Q_k\right) \\
&= \frac{1}{k+1}\left(r_{k+1} + (k+1)Q_k - Q_k\right) \\
&= Q_k + \frac{1}{k+1}\left(r_{k+1} - Q_k\right)
\end{aligned}
$$

Aquesta expressió serveix fins i tot per $k = 0$ i dona $Q_1 = r_1$ per un Q_0 arbitrari.

Implementació incremental

- Aquesta formulació incremental segueix una forma general interessant:

$$NovaEst \leftarrow EstimacioAnt + MidaPas[ValorActual - EstimacioAnt]$$

- La part $[ValorActual - EstimacioAnt]$ es pot interpretar com l'error de l'estimació i per tant el mètode el que fa és fer un pas en la direcció del valor actual (en el nostre cas, el valor actual és la recompensa r_{k+1}).
- Tal i com ho hem formulat, la mida del pas, que es pot escriure com $\alpha_k(a)$, canvia a cada pas: quan processem la recompensa k de l'acció a el mètode usa una mida $\frac{1}{k}$ (i així totes tenen el mateix pes).

El cas dels problemes no estacionaris

- Si la distribució de recompenses va variant amb el temps (cas no estacionari) no té sentit estimar les recompenses com ho fem: hem de donar més pes a les més recents.
- Una manera d'implementar això és usar una mida de pas constant a l'equació. Per exemple:

$$Q_{k+1} = Q_k + \alpha\,(r_{k+1} - Q_k)$$

on la mida del pas, α, $0 < \alpha \leq 1$, és constant.

El cas dels problemes no estacionaris

- Això resulta en una fórmula on Q_k és una mitja ponderada de les recompenses passades i de l'estimació inicial Q_0:

$$Q_k = Q_{k-1} + \alpha [r_k - Q_{k-1}]$$

El cas dels problemes no estacionaris

- Això resulta en una fórmula on Q_k és una mitja ponderada de les recompenses passades i de l'estimació inicial Q_0:

$$\begin{aligned} Q_k &= Q_{k-1} + \alpha\left[r_k - Q_{k-1}\right] \\ &= \alpha r_k + (1-\alpha) Q_{k-1} \end{aligned}$$

El cas dels problemes no estacionaris

- Això resulta en una fórmula on Q_k és una mitja ponderada de les recompenses passades i de l'estimació inicial Q_0:

$$\begin{aligned} Q_k &= Q_{k-1} + \alpha\left[r_k - Q_{k-1}\right] \\ &= \alpha r_k + (1-\alpha)Q_{k-1} \\ &= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 Q_{k-2} \end{aligned}$$

El cas dels problemes no estacionaris

- Això resulta en una fórmula on Q_k és una mitja ponderada de les recompenses passades i de l'estimació inicial Q_0:

$$\begin{aligned} Q_k &= Q_{k-1} + \alpha\left[r_k - Q_{k-1}\right] \\ &= \alpha r_k + (1-\alpha)Q_{k-1} \\ &= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 Q_{k-2} \\ &= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 \alpha r_{k-2} + \ldots \\ &\quad + (1-\alpha)^{k-1}\alpha r_1 + (1-\alpha)^k \alpha Q_0 \end{aligned}$$

El cas dels problemes no estacionaris

- Això resulta en una fórmula on Q_k és una mitja ponderada de les recompenses passades i de l'estimació inicial Q_0:

$$\begin{aligned} Q_k &= Q_{k-1} + \alpha\left[r_k - Q_{k-1}\right] \\ &= \alpha r_k + (1-\alpha)Q_{k-1} \\ &= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 Q_{k-2} \\ &= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 \alpha r_{k-2} + \ldots \\ &\quad +(1-\alpha)^{k-1}\alpha r_1 + (1-\alpha)^k \alpha Q_0 \\ &= (1-\alpha)^k Q_0 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} r_i \end{aligned}$$

El cas dels problemes no estacionaris

- Això és una mitja ponderada perquè la suma dels pesos és $(1-\alpha^k)+\sum_{i=1}^{k}\alpha(1-\alpha)^{k-i}=1$, tal i com es pot comprovar.
- El pes de la recompensa r_i, $\alpha(1-\alpha)^{k-i}$, va decreixent *exponencialment* en funció a mesura que el nombre de recompenses augmenta (que és el que voliem!).

Comparació de reforçament

- Una alternativa als valors de base que hem fet servir amb el *softmax*, que usa directament els valors de les accions, és fer servir una mesura de la seva distància a algun nivell de referència global, que anomenarem *recompensa de referència*.
- Una tria natural d'aquest valor és la mitja de totes les recompenses rebudes. Els mètodes que fan servir aquest concepte s'anomenen mètodes de *comparació de reforçament*. Aquests mètodes, en alguns escenaris, són més efectius que els mètodes basats en el valor de l'acció.

Comparació de reforçament

- Usualment aquests mètodes no mantenen estimacions dels valors de les accions sinó únicament un nivell de recompensa general. Per escollir una acció mantenen una mesura de la seva *preferència per cada una de les accions*.
- Sigui $p_t(a)$ la preferència per l'acció a a la jugada t. Les preferències poden ser usades per determinar les probabilitats de selecció de les accions segons una regla softmax:

$$\pi_t(a) = \frac{e^{p_t(a)}}{\sum_{b=1}^{n} e^{p_t(b)}}$$

on $\pi_t(a)$ denota la probabilitat de seleccionar una acció a en la jugada t.

Comparació de reforçament

- La idea de comparació de reforçament s'usa quan actualitzem les preferències de les accions. Després de cada jugada, la preferència de l'acció seleccionada a la jugada, a_t, s'incrementa per la diferència entre la recompensa, r_t, i la recompensa de referència, $\overline{r}_t$:

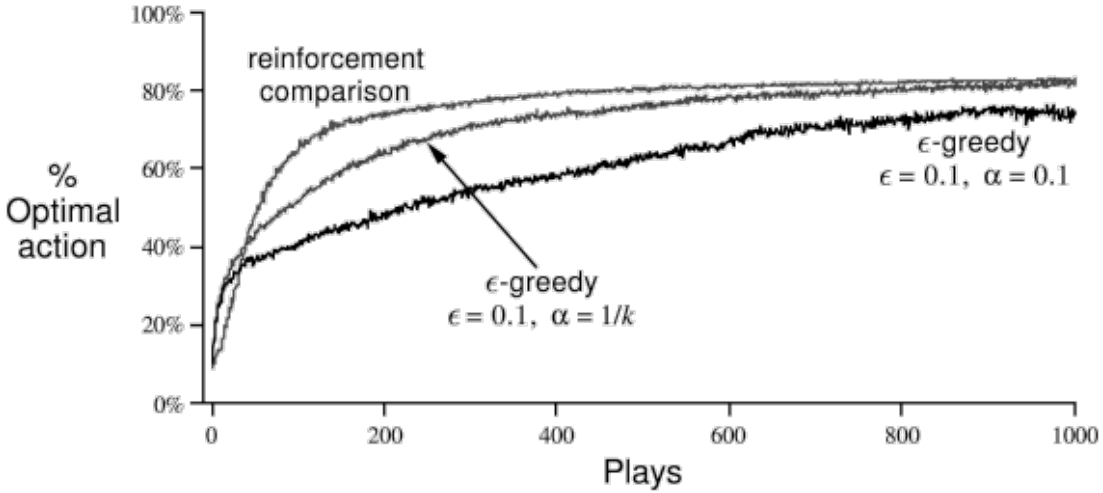
$$p_{t+1}(a_t) = p_t(a_t) + \beta\,[r_t - \overline{r}_t] \tag{2}$$

  on β és un paràmetre positiu de la mida del pas.
- La recompensa de referència és una mitja incremental de totes les recompenses rebudes, independentment de quines accions les han generades. Després d'aplicar (2) actualitzem la recompensa de referència:

$$\overline{r}_{t+1} = \overline{r}_t + \alpha\,[r_t - \overline{r}_t]$$

  on α, $0 < \alpha \leq 1$, és el paràmetre de mida de pas habitual.

Comparació de reforçament



El gràfic mostra l'eficiència d'aquest algorisme ($\alpha = 0.1$) en comparació amb els mètodes ϵ-voraços.

Mètodes de persecució

- Una darrera classe d'algorismes pel problema dels *n-armed bandits* són els *mètodes de persecució*.
- Aquests mètodes mantenen tant les estimacions dels valors de les accions com les preferències per les accions i van fent que les preferències vagin *perseguint* contínuament l'acció més voraç segons l'estimació actual dels valors de les accions.
- La preferència d'una acció es representa amb la probabilitat $\pi_t(a)$, que descriu la probabilitat amb que a la jugada t es seleccioni l'acció a.
- Després de cada jugada, les probabilitats s'actualitzen de manera que l'acció més voraç incrementi la seva probabilitat d'ésser seleccionada.

Mètodes de persecució

- Després de la jugada t, sigui $a^*_{t+1} = argmax_a Q_{t+1}(a)$ l'acció voraç per la jugada $t+1$. Llavors, la probabilitat de seleccionar $a_{t+1} = a^*_{t+1}$ s'incrementa en una fracció β que l'aproxima a 1:

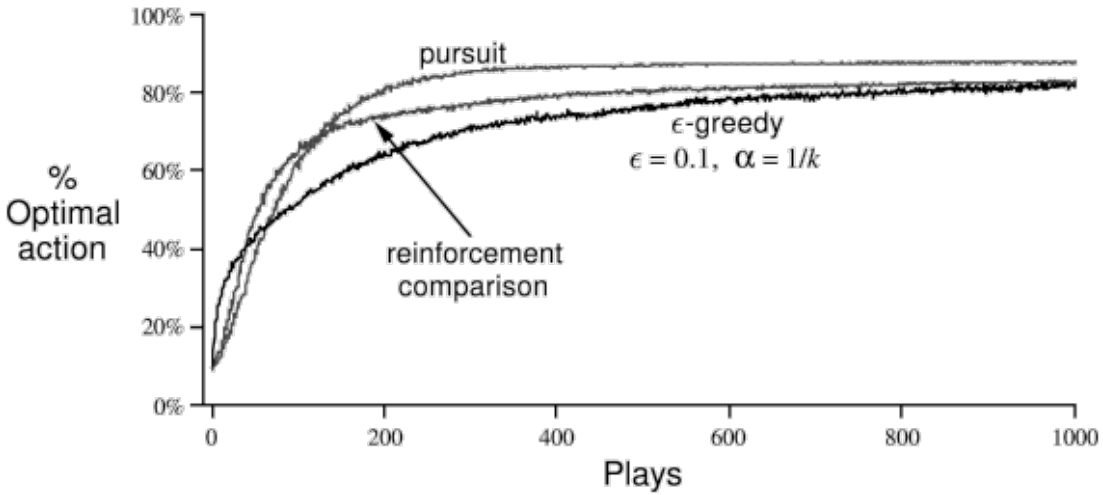
$$\pi_{t+1}(a^*_{t+1}) = \pi_t(a^*_{t+1}) + \beta \left[1 - \pi_t(a^*_{t+1})\right]$$

i les probabilitats de seleccionar les altres accions es redueixen cap a zero:

$$\pi_{t+1}(a) = \pi_t(a) + \beta \left[0 - \pi_t(a)\right]$$

Els valors de les accions, $Q_{t+1}(a)$, s'actualitzen seguint algun dels mètodes vist anteriorment (per exemple, fent que siguin els promitjos de les mostres de les recompenses observades).

Mètodes de persecució



El gràfic mostra l'eficiència d'aquest algorisme quan els valors de les accions s'estimen com mitjes de les mostres, $\pi_0(a) = \frac{1}{n}$ i $\beta = 0.01$.