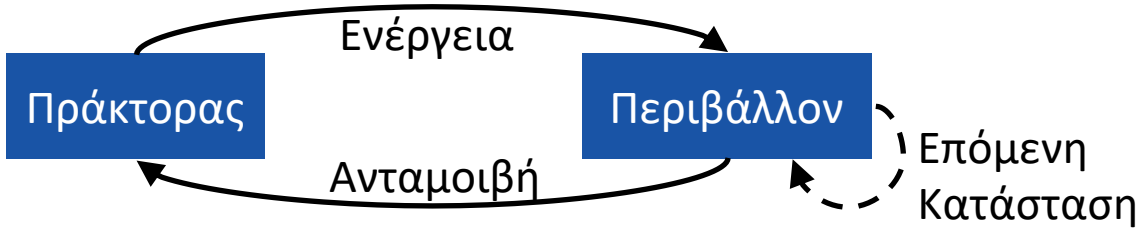
Μηχανική μάθηση

Ενισχυτική μάθηση

Τι είναι η ενισχυτική μάθηση;

- Ένα *σύστημα* (*πράκτορας*) αλληλοεπιδρά με το *περιβάλλον* εκτελώντας **ενέργειες** (**actions**) και λαμβάνοντας **ανταμοιβές** (**rewards**).



- Η ανταμοιβή μπορεί να μην είναι άμεσα διαθέσιμη και να προκύπτει στο τέλος κάποιας ακολουθίας ενεργειών, πχ. στο τέλος μιας παρτίδας παιχνιδιού (νίκη, ισοπαλία, ήττα).
- Το σύστημα μπορεί να είναι δυναμικό.
- Στόχοι μάθησης:
  - Να εκτιμηθεί η βέλτιστη πολιτική ενεργειών ώστε να μεγιστοποιηθεί η ανταμοιβή
  - Να εκτιμηθεί η κατανομή πιθανότητας των ανταμοιβών
  - Αν το σύστημα είναι δυναμικό να αποτιμηθεί η κατάστασή του ή να εκτιμηθεί η πιθανότητα μετάβασης στην επόμενη κατάσταση

Κίνητρο και εφαρμογές

- Νέα κλάση μεθόδων μάθησης διαφορετική από μάθηση με επίβλεψη ή τη μάθηση χωρίς επίβλεψη.
- Κεντρικές έννοιες:
  - **Εξερεύνηση (Exploration)**: δοκιμή πολλών διαφορετικών ενεργειών ώστε να καταγραφεί η αντίδραση του περιβάλλοντος
  - **Εκμετάλλευση (Exploitation)**: χρήση των εκτιμήσεών μας έτσι ώστε να επιλέξουμε τις καλύτερες ενέργειες
  - Δίλημμα εξερεύνησης/εκμετάλλευσης
- *Εφαρμογές:*
  - Ρομποτική
  - Λήψη αποφάσεων
  - Παιχνίδια

Περιγραφή προβλήματος μονόχειρων ληστών

- Το απλούστερο πρόβλημα ενισχυτικής μάθησης.
- Στατικό περιβάλλον χωρίς μνήμη
- Ορισμός: Κάθε χρονική στιγμή t, σε μια χρονική ακολουθία $t = 1, \dots, T$, πρέπει να επιλέξουμε 1 ανάμεσα από k πιθανές ***ενέργειες***. Έστω $A(t)$ η ενέργεια που επιλέγουμε τη στιγμή t.
  - Κάθε ενέργεια a δίνει μια ανταμοιβή q που εξαρτάται μόνο από το a (και όχι από το t). Έτσι έχουμε μια ακολουθία ανταμοιβών $R(t)$:

    $$A(t) = a \Rightarrow R(t) = q$$

  - Η ανταμοιβή q είναι μια τυχαία μεταβλητή με άγνωστη κατανομή.
  - Ζητάμε την **αναμενόμενη ανταμοιβή** με δεδομένο το a:

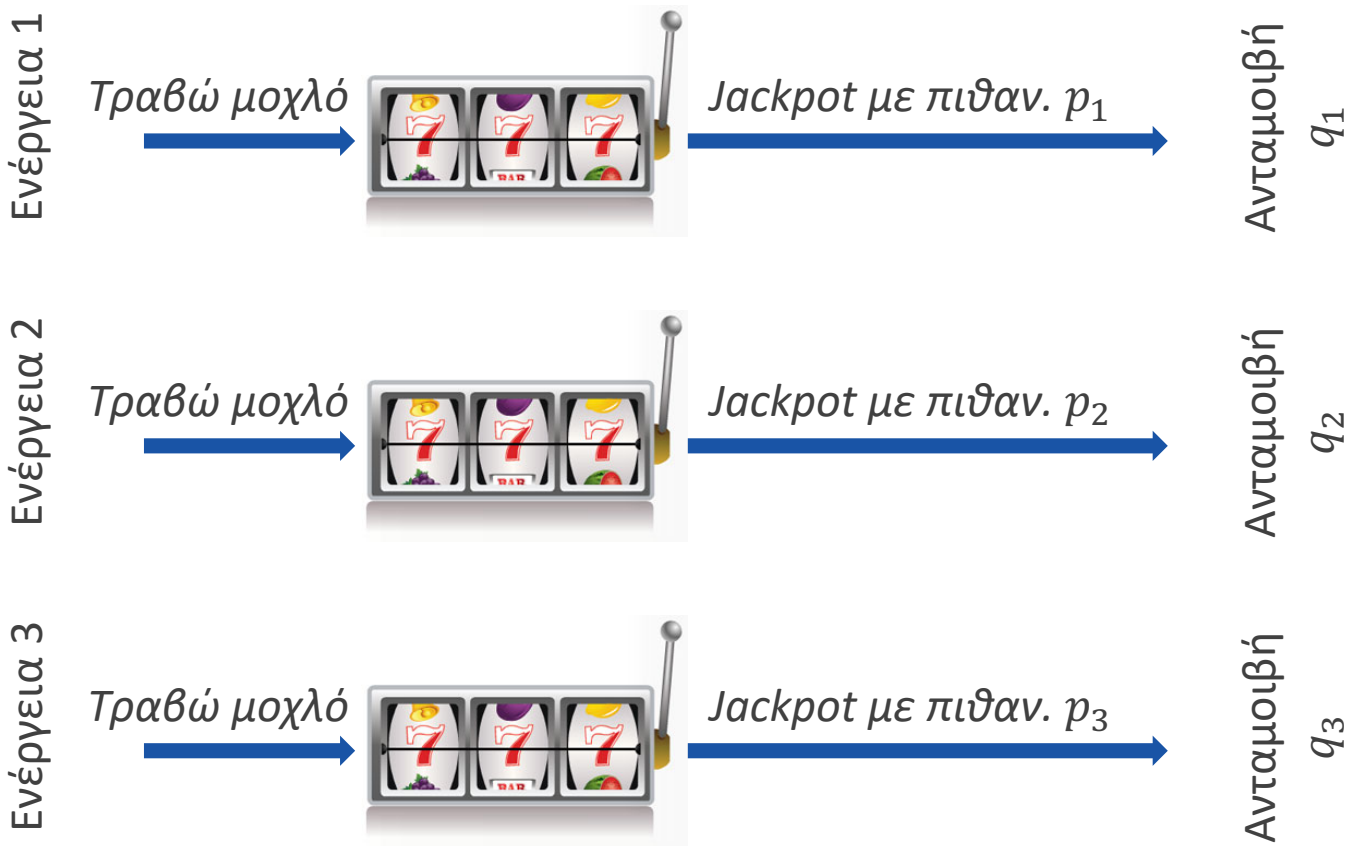
    $$\mu(a) = E\{R(t)|A(t) = a\} = \int_q P(q|a)dq$$

  - Στόχος είναι να επιλέξουμε μια ακολουθία ενεργειών ώστε να μεγιστοποιηθεί η συνολική αναμενόμενη ανταμοιβή για τη χρονική περίοδο T.

Παραδείγματα εφαρμογών

- Ιατρικές εφαρμογές: Έστω ότι έχουμε k θεραπείες για την ίδια ασθένεια.
  - Θέλουμε να βρούμε την καλύτερη θεραπεία για το μέσο πληθυσμό
  - Δεν θέλουμε να δώσουμε κακές ή όχι βέλτιστες θεραπείες σε πολλούς ασθενείς
- Συστήματα συστάσεων: Θέλουμε να συστήσουμε k προϊόντα σε χρήστες.
  - Θέλουμε να βρούμε την καλύτερη σύσταση για το μέσο πληθυσμό
  - Δεν θέλουμε να κάνουμε πολλές κακές συστάσεις στους χρήστες
- Δρομολόγηση (Δίκτυα υπολογιστών): Έχουμε k δυνατές διαδρομές για ένα μήνυμα
  - Θέλουμε να βρούμε την καλύτερη διαδρομή κατά μέσο όρο
  - Δεν θέλουμε να κάνουμε πολλές κακές δοκιμές

Ενέργειες και ανταμοιβές



Αν ξέρουμε την αναμενόμενη ανταμοιβή για κάθε ενέργεια τότε θα επιλέγουμε πάντα την ενέργεια με την μεγαλύτερη ανταμοιβή

Εξερεύνηση εναντίον εκμετάλλευσης

- **Εξερεύνηση**: Καθώς αρχικά δεν ξέρουμε την κατανομή πιθανότητας για την ανταμοιβή καμίας ενέργειας, πρέπει να την εκτιμήσουμε κάνοντας δοκιμές
- **Εκμετάλλευση**: Αφού εκτιμήσουμε τις πιθανότητες ανταμοιβών εκμεταλλευόμαστε τη γνώση αυτή για να κάνουμε την καλύτερη επιλογή ενέργειας.
- Η Εξερεύνηση είναι απαραίτητη ώστε να συλλεχθούν στατιστικά για τις ανταμοιβές των ενεργειών και να εκτιμηθεί η ενέργεια με την μεγαλύτερη ανταμοιβή. Από την άλλη μεριά, κατά τη διάρκεια της εξερεύνησης μπορεί να δοκιμάζουμε μη βέλτιστες ενέργειες.
- Το δίλημμα Εξερεύνηση/Εκμετάλλευση: χωρίς εξερεύνηση κάνουμε ενέργειες στα τυφλά. Εφαρμόζοντας πολλή εξερεύνηση κάνουμε πολλές κακές ή μη βέλτιστες ενέργειες.

Απλοϊκή προσέγγιση

- Εξερευνούμε κάθε ενέργεια a ένα συγκεκριμένο πλήθος φορών έτσι ώστε να εκτιμήσουμε την αναμενόμενη ανταμοιβή $\hat{\mu}(a)$. Κατόπιν, θα επιλέγουμε συνεχώς την ενέργεια με την μεγαλύτερη αναμενόμενη ανταμοιβή:
  - Διάλεξε μια ενέργεια a επανειλημμένα N φορές
  - Σύλλεξε στατιστικά και εκτίμησε την κατανομή της ανταμοιβής γι' αυτή την ενέργεια
  - Εκτίμησε την αναμενόμενη ανταμοιβή για την ενέργεια a:

$$\hat{\mu}(a) = \frac{\text{Άθροισμα ανταμοιβών } R(i) \text{ όταν επιλέγουμε } a}{\text{Πλήθος φορών που επιλέξαμε } a}$$

  - Αφού υπολογίσουμε όλα τα $\hat{\mu}(1), \dots, \hat{\mu}(k)$, επιλέγουμε πάντα την ενέργεια a^* με τη μεγαλύτερη αναμενόμενη ανταμοιβή $\hat{\mu}(a^*)$.
- Σημαντική παρατήρηση: έχουμε υποθέσει ότι η στατιστική συμπεριφορά των ανταμοιβών δεν εξαρτάται από το χρόνο t αλλά μόνο από την ενέργεια a.

Παράδειγμα: 3 bandits

- Έστω ότι έχουμε 3 δυνατές ενέργειες $a = 0$, $a = 1$, $a = 2$, με ανταμοιβές που ακολουθούν την Γκαουσσιανή κατανομή

- Ανταμοιβή για $a = 0$:

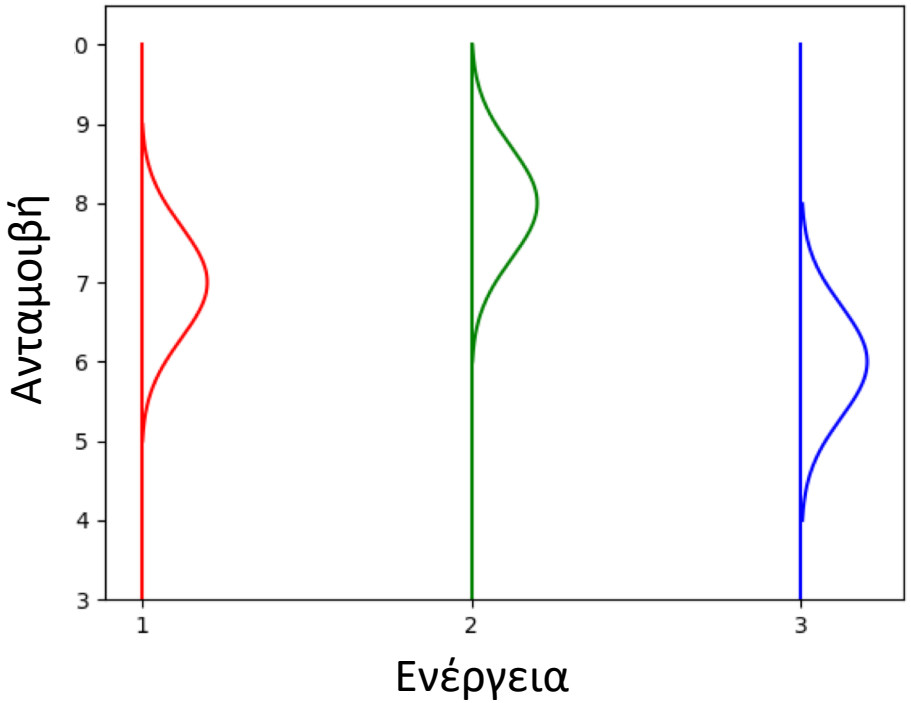
$q(0) \sim N(\mu = 7, \sigma^2 = 1)$

- Ανταμοιβή για $a = 1$:

$q(1) \sim N(\mu = 8, \sigma^2 = 1)$

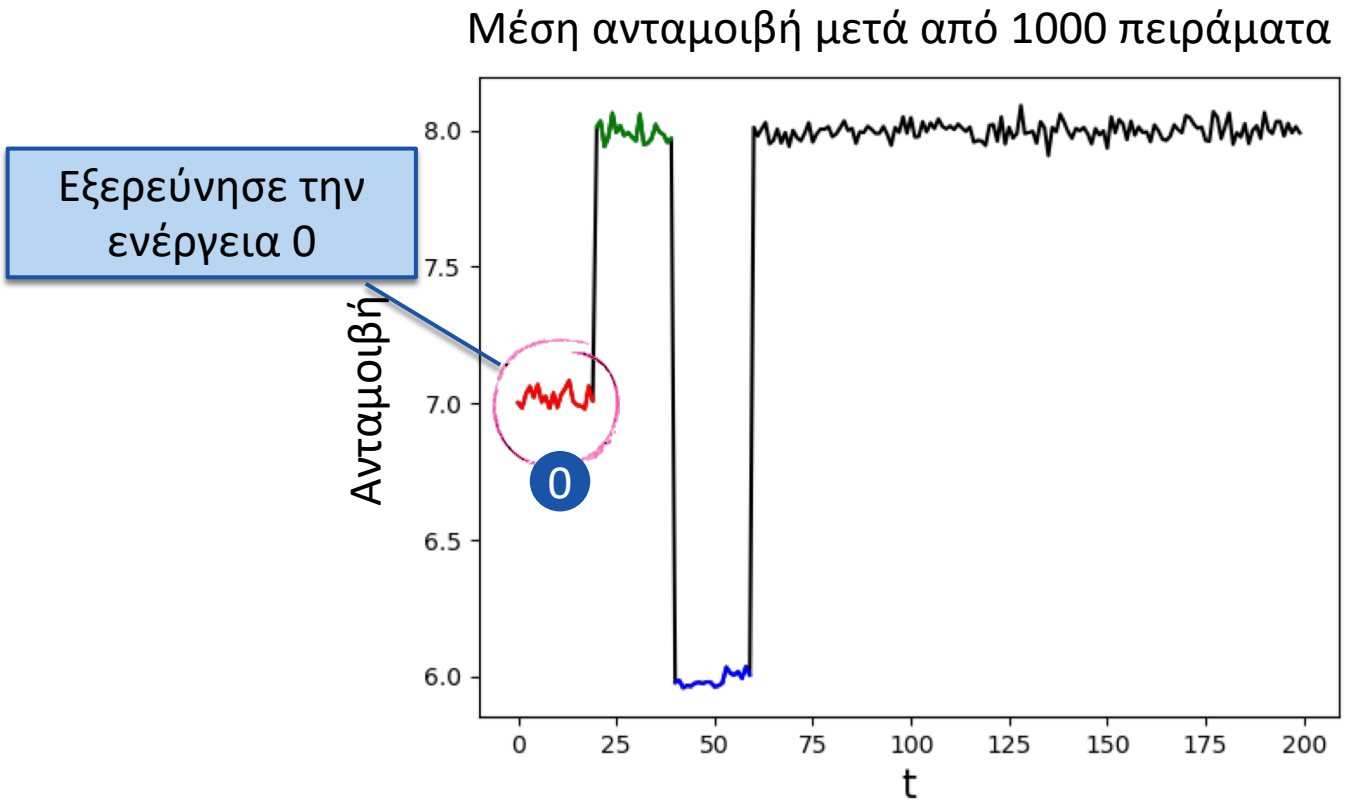
- Ανταμοιβή για $a = 2$:

$q(2) \sim N(\mu = 6, \sigma^2 = 1)$



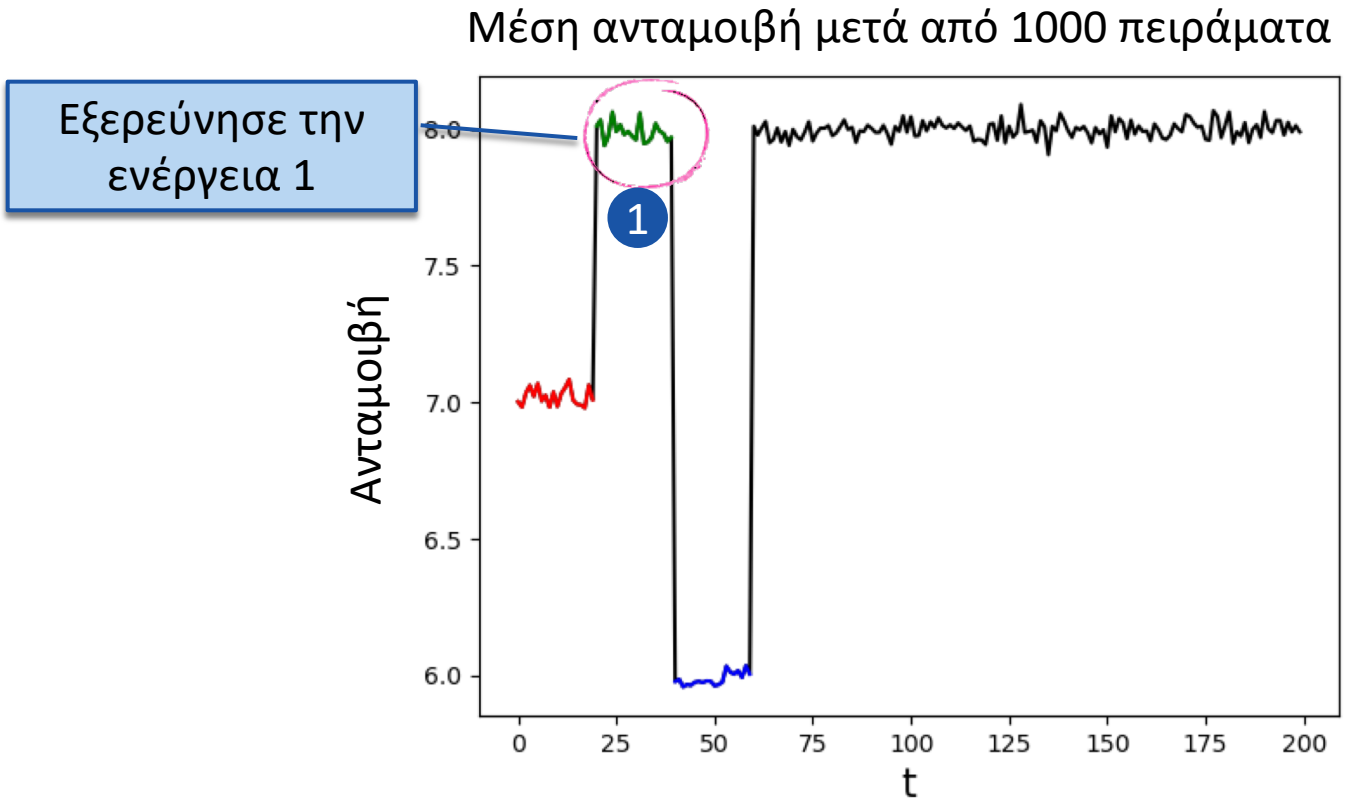
Παράδειγμα: 3 bandits

- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια



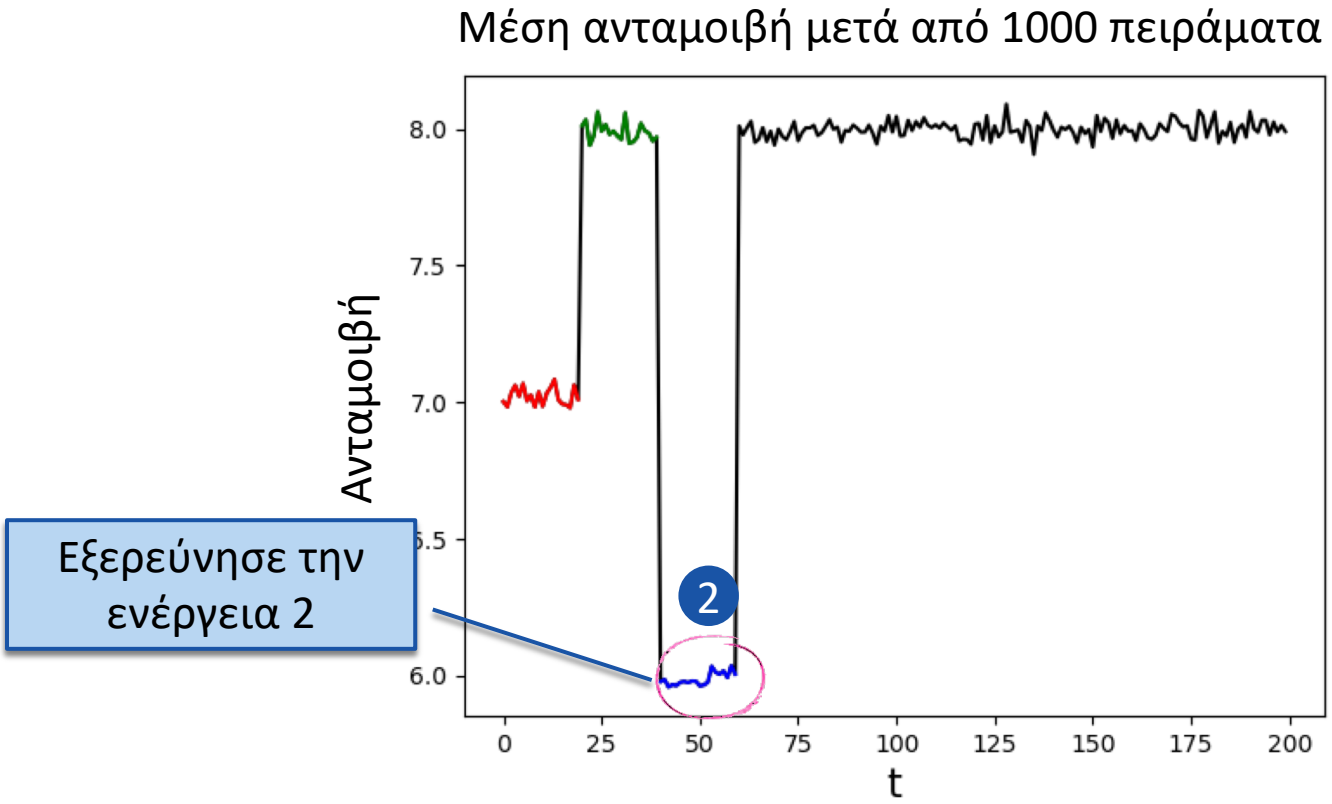
Παράδειγμα: 3 bandits

- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια



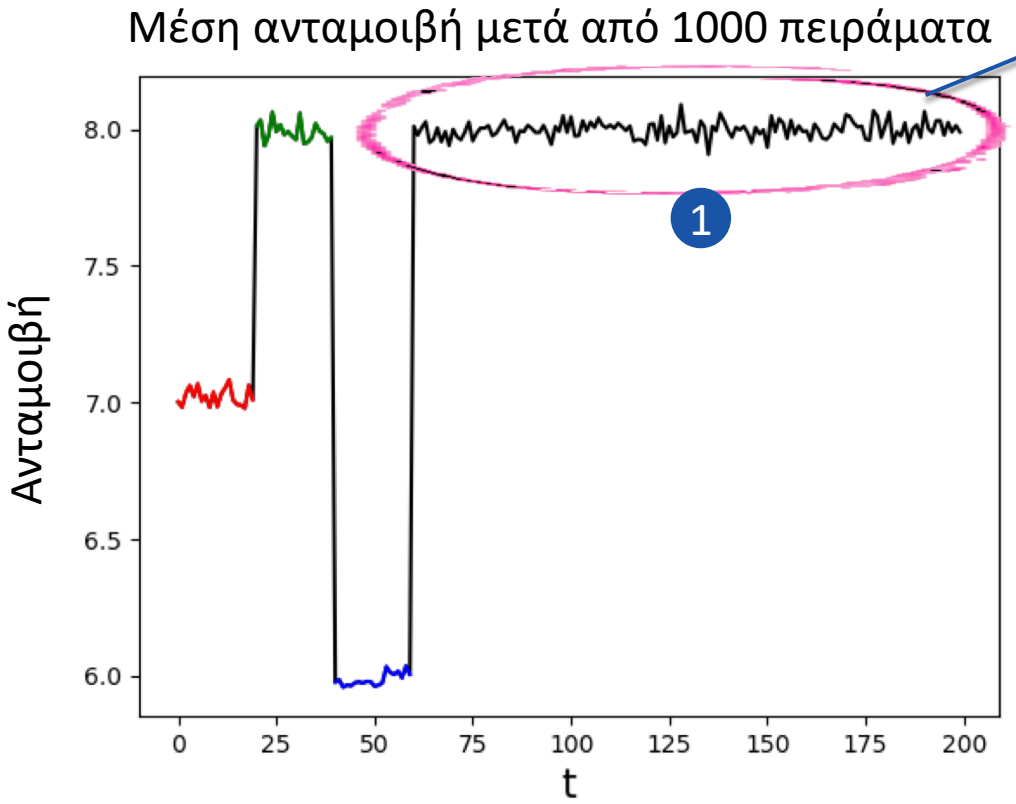
Παράδειγμα: 3 bandits

- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια



Παράδειγμα: 3 bandits

- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια



Επίλεγε πάντα την ενέργεια 1 διότι έχει τη μεγαλύτερη αναμενόμενη ανταμοιβή

Άπληστος αλγόριθμος

- Εναλλακτικά μπορούμε να εκμεταλλευόμαστε αμέσως τη γνώση που έχουμε συλλέξει μέχρι στιγμής.
- Αυτό σημαίνει ότι έχουμε μια εκτίμηση $\hat{\mu}_t(a)$ της αναμενόμενης ανταμοιβής για κάθε ενέργεια a τη στιγμή t με βάση τις μέχρι τώρα παρατηρήσεις μας.

$$\hat{\mu}_t(a) = \frac{\text{Άθροισμα ανταμοιβών } R\ (i) \text{ όταν επιλέγω } a \text{ πριν τη στιγμή } t}{\text{Πλήθος φορών που επέλεξα } a \text{ πριν τη στιγμή } t}$$

$$= \frac{\sum_{i=1, A(i)=a}^{t-1} R(i)}{n_{t-1}(a)}$$

- Σε κάθε χρονική t στιγμή επιλέγω την ενέργεια με τη μέγιστη εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a)$.
- Αφού συλλέξουμε την ανταμοιβή $R(t)$, ενημερώνουμε την εκτίμησή μας $\hat{\mu}_{t+1}(a)$ για την επιλεγμένη ενέργεια $A(t) = a$:

$$\hat{\mu}_{t+1}(a) = \frac{\sum_{i=1}^{t} R(i)}{n_t(a)}$$

- Η μέθοδος καλείται <u>άπληστη</u> διότι κάθε στιγμή επιλέγουμε την δράση με την μέγιστη εκτιμώμενη ανταμοιβή αμέσως μόλις ενημερωθούν οι εκτιμήσεις μας.

Επαναληπτική μέθοδος

- Ξεκίνα με κάποια αρχική εκτίμηση $\hat{\mu}(a)$ και θέσε $n(a) = 0 =$ μετρητής φορών που επιλέγω την ενέργεια a.
- Για κάθε χρονική στιγμή t:
  - Επίλεξε την ενέργεια $A(t) = a^*$ με το μέγιστο $\hat{\mu}(a)$:

$$a^* = \arg\max_a \hat{\mu}(a)$$

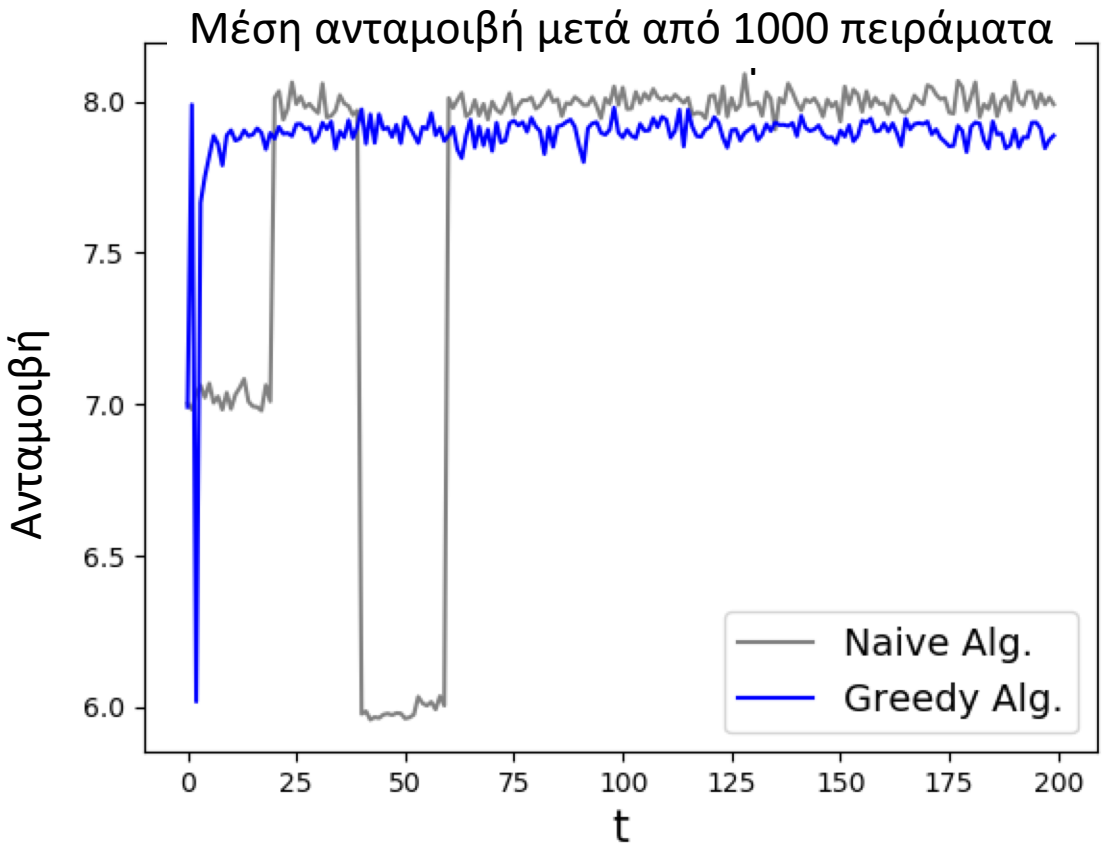
    (σε περίπτωση ισοπαλίας επέλεξε τυχαία)
  - Κατάγραψε την ανταμοιβή $R(t)$
  - Ενημέρωσε τον μετρητή:

$$n(a^*) \leftarrow n(a^*) + 1$$

  - Ενημέρωσε την αναμενόμενη ανταμοιβή για την ενέργεια a^*:

$$\hat{\mu}(a^*) \leftarrow \frac{n(a^*)-1}{n(a^*)}\hat{\mu}(a^*) + \frac{1}{n(a^*)}R(t)$$
$$= \hat{\mu}(a^*) + \frac{1}{n(a^*)}[R(t) - \hat{\mu}(a^*)]$$

Απλοϊκός εναντίον Άπληστου αλγόριθμου



ε-άπληστος αλγόριθμος

- Μειονέκτημα του άπληστου αλγορίθμου: Επιλέγοντας πάντα την ενέργεια με τη μεγαλύτερη εκτιμώμενη ανταμοιβή $\hat{\mu}(a)$ μπορεί να αγνοούμε ενέργειες που δεν έχουμε δει μέχρι στιγμής → **ελλιπής εξερεύνηση!**
- Μια άλλη εναλλακτική μέθοδος είναι με πιθανότητα $(1-\varepsilon)$ να επιλέγουμε ενέργειες με την άπληστη μέθοδο, ενώ με πιθανότητα ε να επιλέγουμε ενέργειες τυχαία ώστε να δίνουμε στο σύστημα την δυνατότητα να εξερευνάει κι άλλες ενέργειες. Αυτή η μέθοδος είναι γνωστή ως ε-άπληστος αλγόριθμος.
- Προφανώς ο άπληστος αλγόριθμος είναι ειδική περίπτωση του ε-άπληστου αλγορίθμου για $\varepsilon = 0$.

ε-άπληστος αλγόριθμος

- Ξεκίνα με $\hat{\mu}(a) = 0$ και $n(a) = 0$
- Για κάθε χρονική στιγμή t:
  - Με πιθανότητα $1 - \varepsilon$ επίλεξε την ενέργεια $A(t) = a = \arg\max_{a'} \hat{\mu}(a')$
  - Με πιθανότητα ε επίλεξε την ενέργεια $A(t) = a =$ τυχαία
  - Σύλλεξε την ανταμοιβή $R(t)$
  - Ενημέρωσε τον μετρητή:

$$n(a) \leftarrow n(a) + 1$$

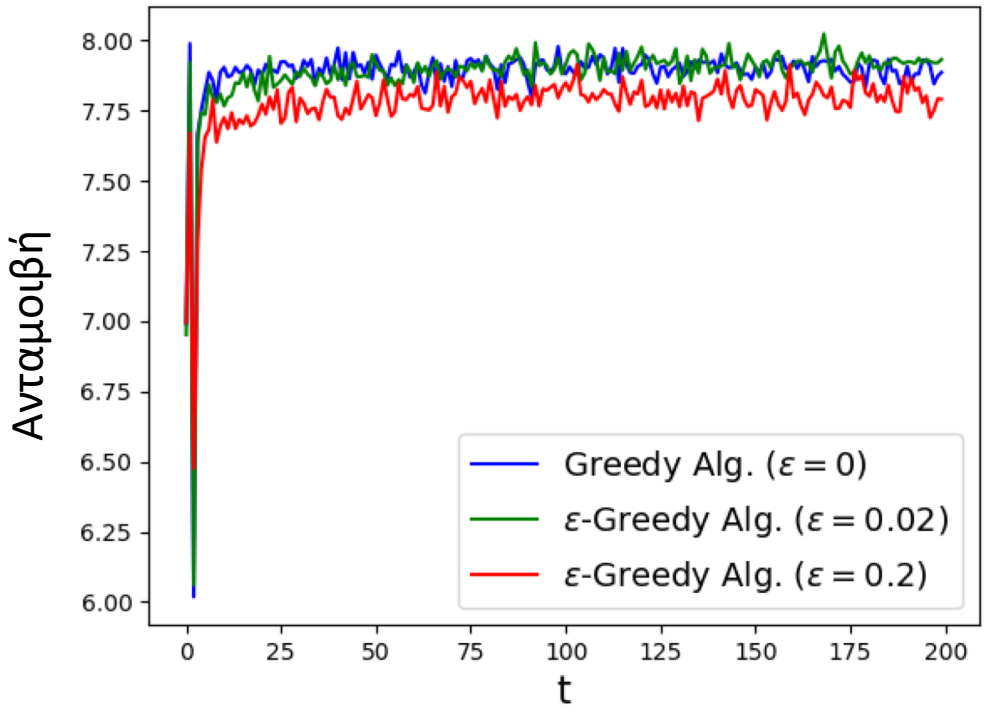
  - Ενημέρωσε την αναμενόμενη ανταμοιβή για την ενέργεια a:

$$\hat{\mu}(a) \leftarrow \frac{n(a^*)-1}{n(a^*)}\hat{\mu}(a) + \frac{1}{n(a)}R(t) = \hat{\mu}(a) + \frac{1}{n(a)}[R(t) - \hat{\mu}(a)]$$

Παράδειγμα: 3 μονόχειρες ληστές

Επίδραση του ε στη μάθηση:

- $\varepsilon = 0$ (άπληστος)
  Όχι καλό αποτέλεσμα
- *Μεγαλύτερο* $\varepsilon \rightarrow$
  Περισσότερη εξερεύνηση
  $\rightarrow$ βελτίωση επίδοσης
  καθώς το t μεγαλώνει.
- *Πολύ μεγάλο* $\varepsilon \rightarrow$
  Πολλή εξερεύνηση
  $\rightarrow$ δεν βελτιώνει
  την επίδοση.



Δυσαρέσκεια (Regret)

- Η **Δυσαρέσκεια (Regret)** είναι ένα κριτήριο βελτιστοποίησης για την επιλογή της καλύτερης ενέργειας.
- Ορίζεται ως το συνολικό άθροισμα των διαφορών μεταξύ της ανταμοιβής $R(a^*)$ από τη βέλτιστη ενέργεια a^* και της ανταμοιβής $R\big(A(t)\big)$ από την ενέργεια $A(t)$ που επιλέχτηκε:

$$r_T = E\left\{\sum_{t=1}^{T} R(a^*) - R\big(A(t)\big)\right\} = \sum_{t=1}^{T} E\left\{R(a^*) - R\big(A(t)\big)\right\}$$
$$= \sum_{t=1}^{T} \mu(a^*) - \mu\big(A(t)\big)$$

- Έχοντας K δυνατές ενέργειες $a \in \{1, \dots, K\}$ η παραπάνω έκφραση γράφεται

$$r_T = \sum_{a=1}^{K} \big(\mu(a^*) - \mu(a)\big) n(a)$$

όπου $n(a)$ είναι ο γνωστός μετρητής φορών που επιλέξαμε την ενέργεια a.

Μέθοδος Upper Confidence Bound

- **Ιδέα**: Τη στιγμή t επίλεξε την ενέργεια a με βάση το συνδυασμό της έως τώρα εκτιμώμενης ανταμοιβής $\hat{\mu}_t(a)$ **και της *αβεβαιότητας* της εκτίμησής μας η οποία προκύπτει από το πόσες φορές έχουμε εκτελέσει αυτή την ενέργεια στο παρελθόν**.
- Αν η ενέργεια a δεν έχει δοκιμαστεί πολλές φορές μέχρι τώρα, τότε **δεν είμαστε πολύ βέβαιοι** για την εκτίμηση $\hat{\mu}_t(a)$. Στην περίπτωση αυτή δίνουμε μεγάλο "*bonus*" στην ενέργεια αυτή αυξάνοντας την πιθανότητα να την επιλέξουμε. Έτσι, διευκολύνουμε την εξερεύνηση.
- Το bonus είναι αντιστρόφως ανάλογο του μετρητή $n_t(a)$: όσες πιο πολλές φορές έχουμε εκτελέσει την ενέργεια a τόσο λιγότερο αβέβαιοι είμαστε για την εκτίμησή μας.
- Αν μια ενέργεια έχει δοκιμαστεί πολλές φορές και έχει μικρή εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a)$ τότε δεν θα επιλεγεί.
- Το bonus ενθαρρύνει την εξερεύνηση αλλά δεν πρέπει να είναι τόσο μεγάλο ώστε να επιλέγουμε ενέργειες που δεν έχουν καμία ελπίδα να είναι βέλτιστες.

Μέθοδος Upper Confidence Bound

- **Παράδειγμα**: Η ενέργεια a_1 δοκιμάστηκε 100 φορές και έχει εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a_1) = 10$. Λόγω μικρής αβεβαιότητας έχει μικρό bonus: $B_1 = 1$
- Η ενέργεια a_2 δοκιμάστηκε μόνο 5 φορές και έχει εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a_2) = 9$. Λόγω μεγάλης αβεβαιότητας έχει μεγάλο bonus: $B_2 = 3$.
- Η πολιτική του αλγορίθμου είναι να επιλέξουμε την ενέργεια με το μεγαλύτερο άθροισμα $\hat{\mu}_t + B$.

$$\hat{\mu}(a_1) + B_1 = 11, \qquad \hat{\mu}(a_2) + B_2 = 12$$

- Έτσι, αν και η ενέργεια a_2 έχει μικρότερη αναμενόμενη ανταμοιβή την επιλέγουμε λόγω μεγαλύτερης αβεβαιότητας.
- Αν ωστόσο, η αναμενόμενη ανταμοιβή της a_2 ήταν πολύ μικρή, πχ $\hat{\mu}_t(a_2) = 1$, τότε ακόμη και με το bonus δεν θα την επιλέγαμε διότι είναι πολύ χειρότερη από την a_1. Έτσι δεν σπαταλάμε εξερεύνηση σε ενέργειες που δεν έχουν καμία ελπίδα.

Μέθοδος Upper Confidence Bound

Αλγόριθμος UCB1

- Αρχικοποίηση: Δοκίμασε κάθε ενέργεια από μία φορά. Θέσε αρχικό $\hat{\mu}_1(a)$=ανταμοιβή για την ενέργεια a.
- Για $t = 1 \dots T$

  Επίλεξε την ενέργεια a με το μέγιστο άθροισμα

$$\hat{\mu}_t(a) + \sqrt{\frac{2 \ln t}{n_t(a)}}$$

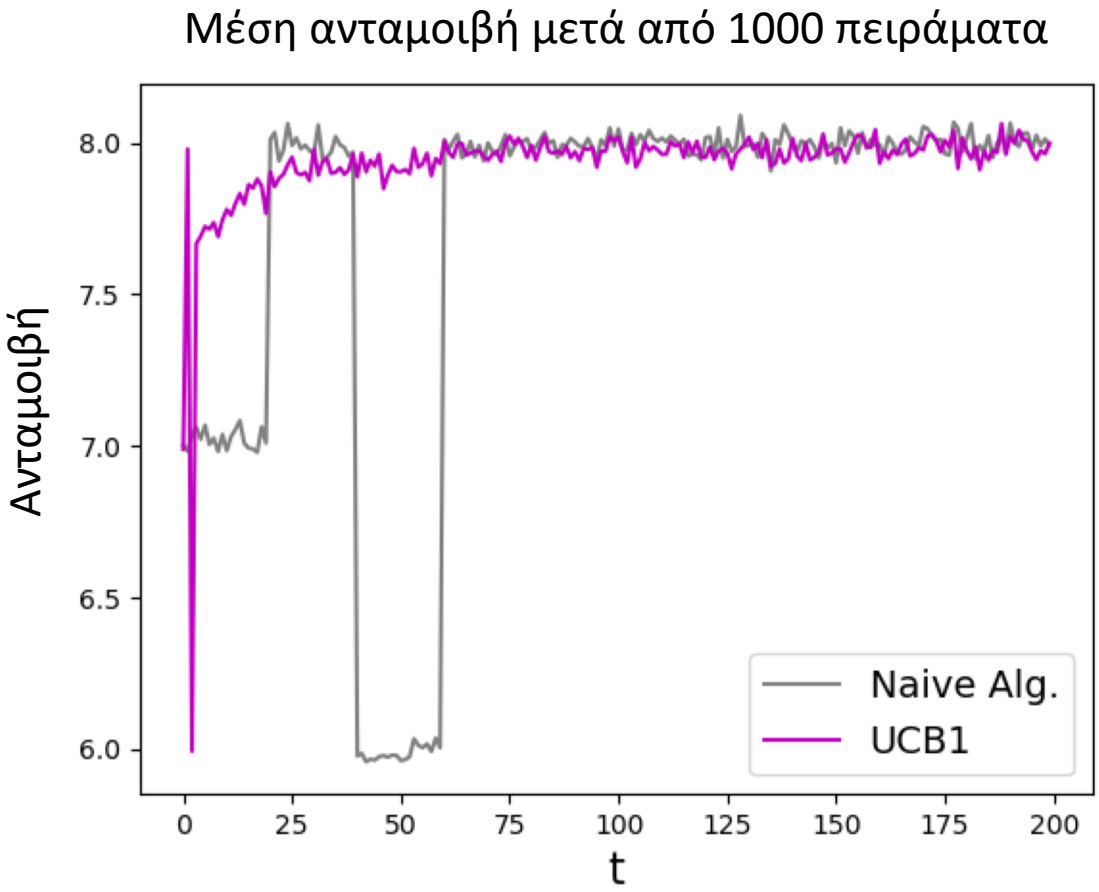
Bonus

  όπου $\hat{\mu}_t(a)$ είναι η εκτιμώμενη ανταμοιβή μέχρι τη στιγμή t

  $n_t(a)$ είναι ο μετρητής των φορών που επιλέξαμε την ενέργεια a μέχρι τη στιγμή t

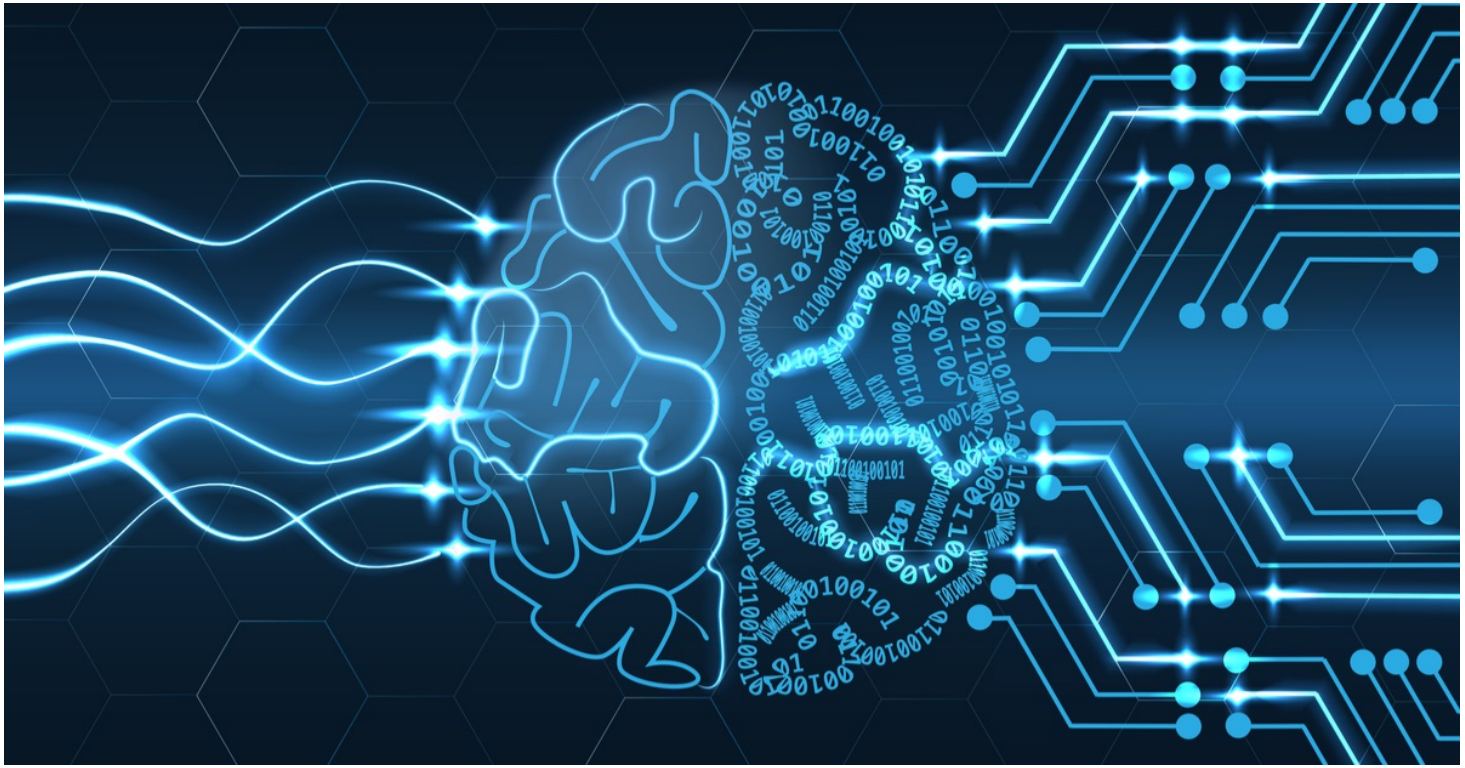
P. Auer, N. Cesa-Bianchi, P. Fischer, “Finite-time Analysis of the Multiarmed Bandit Problem”, Machine Learning, 47, 235–256, 2002

UCB1 εναντίον Απλοϊκής μεθόδου



UCB1 vs. ε-Greedy

Μαρκοβιανές Διαδικασίες αποφάσεων



Markov Decision Processes (MDP)

Αλληλεπίδραση του πράκτορα με το περιβάλλον

Βασικές Υποθέσεις:

- Ένας *πράκτορας* παίρνει αποφάσεις επιλέγοντας ανάμεσα σε K δυνατές *ενέργειες*
- Ο πράκτορας αλληλεπιδρά με το *περιβάλλον* σε διακριτές χρονικές στιγμές t.
- Το περιβάλλον βρίσκεται πάντα σε μια από N δυνατές καταστάσεις.
- Ανάλογα με την ενέργεια $A(t)$ και την κατάσταση $S(t)$ τη στιγμή t ο πράκτορας λαμβάνει μια ανταμοιβή $R(t)$ και το περιβάλλον μεταβαίνει στην κατάσταση $S(t+1)$.
- Ορίζουμε την πιθανότητα το σύστημα να βρίσκεται στην κατάσταση s' δίνοντας ανταμοιβή r τη στιγμή $t+1$ δεδομένου ότι τη στιγμή t ήταν στην κατάσταση s και η επιλεγμένη ενέργεια ήταν η a:
- $p(s', r|s, a) = \Pr(S(t+1) = s', R(t+1) = r \mid S(t) = s, A(t) = a)$

Παράδειγμα: Ρομπότ ανακύκλωσης

- Ένα ρομπότ συλλέγει άδεια κουτάκια αναψυκτικών. Οι δυνατές ενέργειες του ρομπότ είναι:

$$\mathcal{A} = \{\text{"search for cans"}, \text{"wait"}, \text{"recharge"}\}$$

- Η μπαταρία του ρομπότ είναι στην κατάσταση

"low" ή "high".

- Η ανταμοιβή του ρομπότ για την ενέργεια "search" είναι υψηλή, πχ.

$$r_{search} = 10$$

αφού ο σκοπός του είναι η αναζήτηση άδειων κουτιών.

- Η ανταμοιβή του ρομπότ για την ενέργεια "wait", είναι μικρότερη αλλά όχι μηδέν, πχ.

$$r_{wait} = 2$$

Ο λόγος είναι ότι μετά από κάποιο χρόνο αναζήτησης το ρομπότ μπορεί να έχει μαζέψει τα περισσότερα κουτιά και είναι καλύτερο να περιμένει για να μαζευτούν και άλλα αντί να συνεχίζει την αναζήτηση.

Παράδειγμα: Ρομπότ ανακύκλωσης

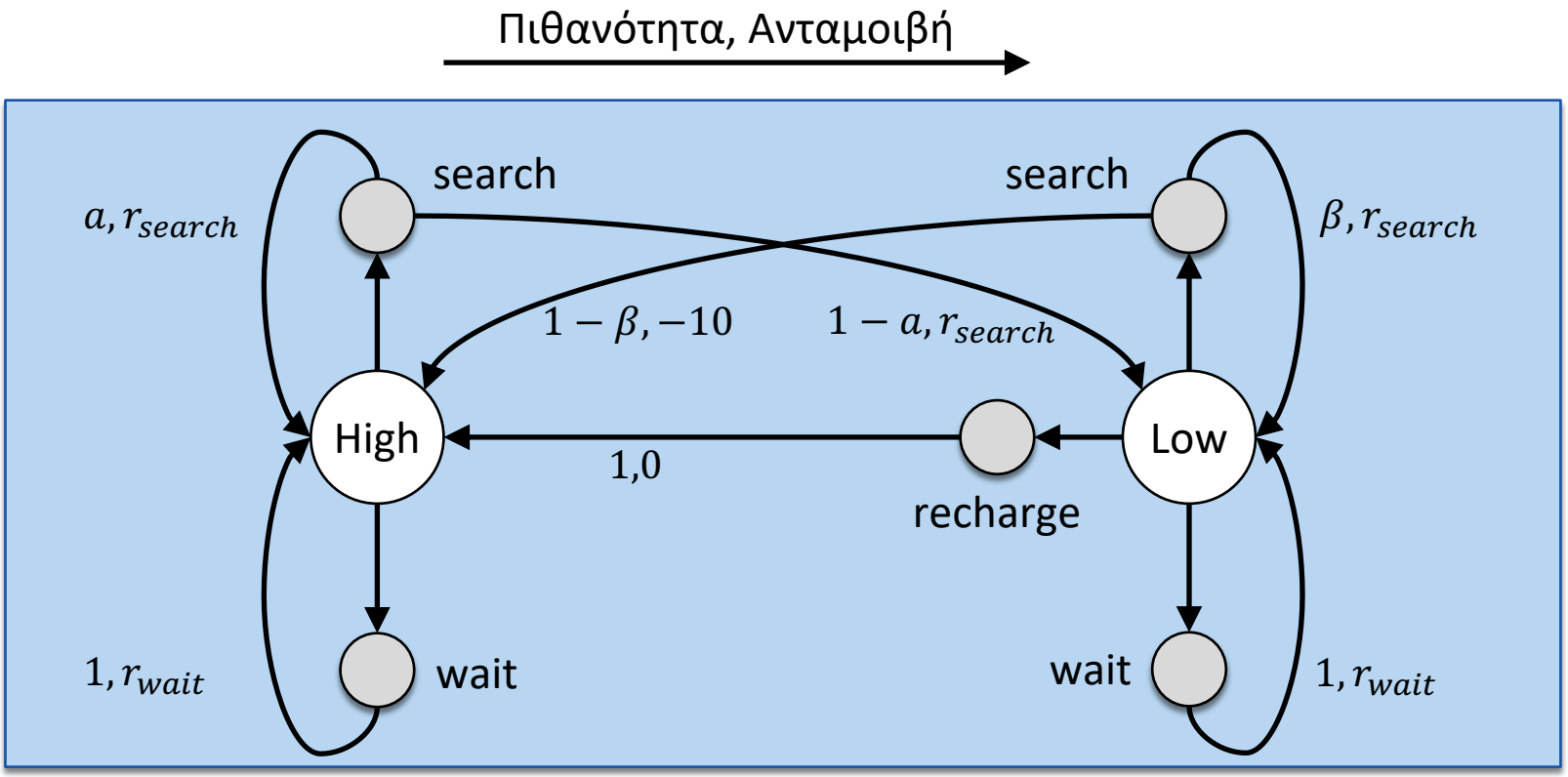
- Πίνακας πιθανοτήτων μετάβασης (Όχι “recharge” όταν κατάσταση=“High”)

$S(t)$	$A(t)$	$S(t+1)$	$p(s'\|s,a)$	$R(t)$
High	Search	High	α	r_{search}
High	Search	Low	$1-\alpha$	r_{search}
Low	Search	High	$1-\beta$	-10
Low	Search	Low	β	r_{search}
High	Wait	High	1	r_{wait}
High	Wait	Low	0	r_{wait}
Low	Wait	High	0	r_{wait}
Low	Wait	Low	1	r_{wait}
Low	Recharge	High	1	0
Low	Recharge	Low	0	0

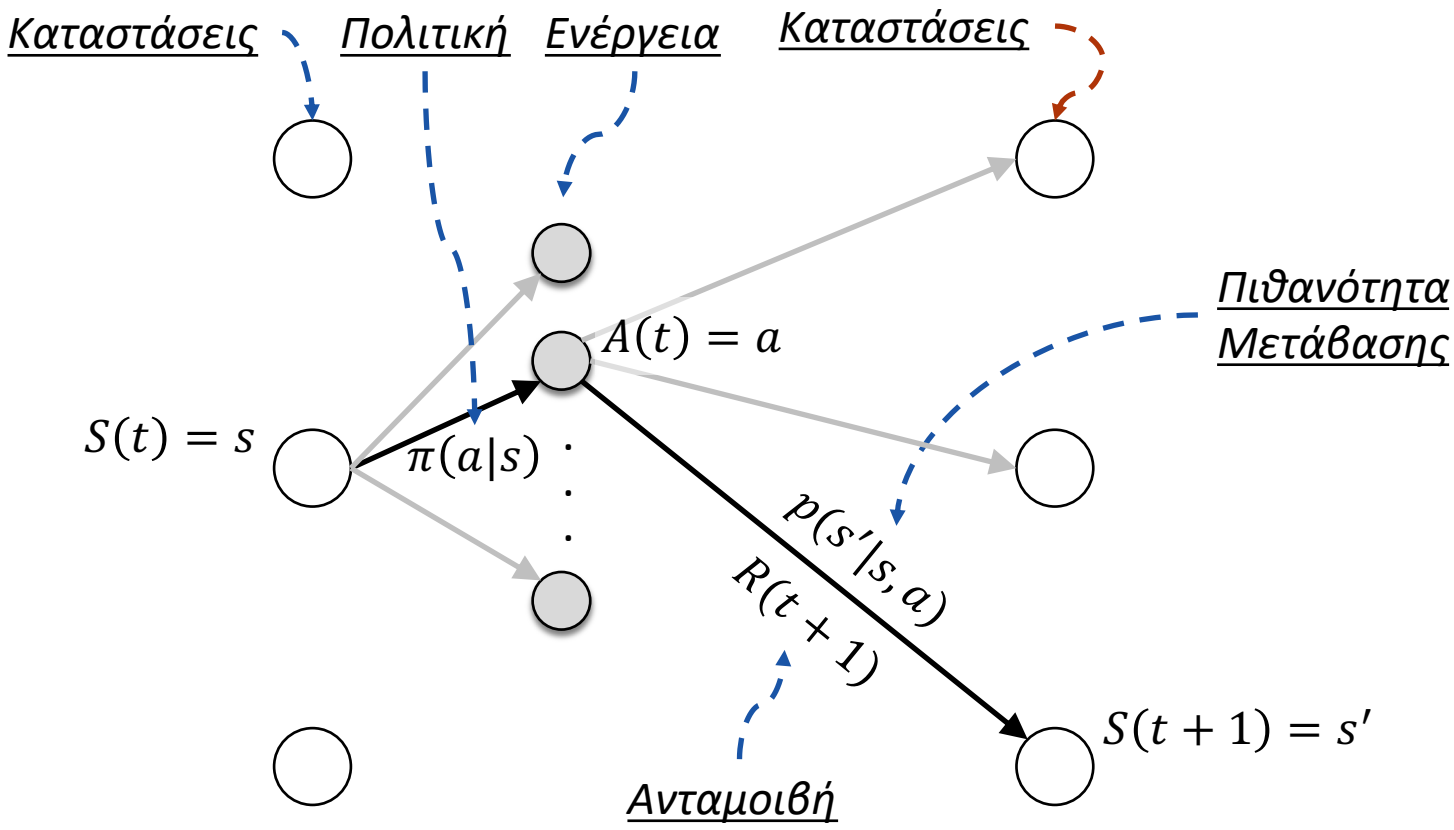
Πρέπει να σώσουμε το ρομπότ και να το φορτίσουμε

Παράδειγμα: Ρομπότ ανακύκλωσης

- Γράφος μετάβασης. = Κόμβος κατάστασης = Κόμβος ενέργειας



Γράφος μετάβασης



Πολιτική, επεισοδιακές και συνεχείς εργασίες

- Ο αλγόριθμος επιλογής ενεργειών με βάση την τρέχουσα κατάσταση του συστήματος και την προηγούμενη εμπειρία καλείται **πολιτική** (**policy**)
- Με δεδομένη την πολιτική επιλέγουμε μια σειρά ενεργειών $A(t)$ και συλλέγουμε μια σειρά ανταμοιβών $R(t)$, για $t = 1, \dots, T$.
- Σε μερικές περιπτώσεις το T είναι πεπερασμένος αριθμός. Για παράδειγμα, όταν ένα ρομπότ βγαίνει από ένα λαβύρινθο το κάνει σε πεπερασμένο πλήθος βημάτων. Παρομοίως όταν παίζουμε ένα παιχνίδι αυτό τελειώνει σε πεπερασμένο πλήθος κινήσεων. Τέτοιες περιπτώσεις καλούνται επεισοδιακές εργασίες ή εργασίες πεπερασμένου ορίζοντα.
- Σε άλλες περιπτώσεις, ωστόσο, είναι ορθό να υποθέσουμε ότι $T = \infty$. Για παράδειγμα ένα βιομηχανικό ρομπότ μπορεί να εκτελεί την ίδια εργασία αδιάλειπτα, ή ένας ελεγκτής πρέπει να εφαρμόζει συνεχώς έλεγχο σε μια συσκευή ή ένα σύστημα. Τέτοιες περιπτώσεις καλούνται συνεχείς εργασίες ή εργασίες άπειρου ορίζοντα.

Κέρδος (Gain)

- Ένα μέτρο της επίδοσης μιας πολιτικής κατά το χρόνο t είναι το άθροισμα των ανταμοιβών που ελήφθησαν μετά το t χρησιμοποιώντας αυτή την πολιτική.
- Ωστόσο, δεδομένου ότι το T μπορεί να είναι άπειρο, το άθροισμα μπορεί να αποκλίνει. Μια λύση στο πρόβλημα είναι να εισαχθεί ένας θετικός *συντελεστής λήθης* (ή *έκπτωσης*) $\gamma < 1$ που μειώνει εκθετικά τη σημασία των ανταμοιβών $R(t+k)$ καθώς $k \to \infty$

$$G(t) = R(t+1) + \gamma R(t+2) + \gamma^2 R(t+3) + \gamma^3 R(t+4) + \cdots$$
$$= \sum_{k=0}^{\infty} \gamma^k R(t+k+1)$$

- Αυτό το κριτήριο ονομάζεται κέρδος κατά το χρόνο t. Εάν οι ανταμοιβές οριοθετούνται μεταξύ πεπερασμένων ορίων, το κέρδος θα συγκλίνει πάντα σε μια πεπερασμένη τιμή.
- Προφανώς το κέρδος υπακούει στον αναδρομικό τύπο

$$G(t) = R(t+1) + \gamma G(t+1).$$

Πολιτική και Αξίες καταστάσεων

- Η πολιτική (policy) π καθορίζεται πλήρως από την πιθανότητα ανάληψης της ενέργειας a με δεδομένη μια κατάσταση s:

$$\pi(a, s) = p(A(t) = a \mid S(t) = s)$$

- Ορίζουμε ως αξία (value) μιας κατάστασης s (στο πλαίσιο μιας πολιτικής π) το αναμενόμενο κέρδος αν ξεκινήσουμε από την κατάσταση s και ακολουθήσουμε την π:

$$v_\pi(s) = E_\pi\{G(t) \mid S(t) = s\}$$

- Ο συμβολισμός $E_\pi\{\cdot\}$ δηλώνει την αναμενόμενη τιμή μιας τυχαίας μεταβλητής, δεδομένου ότι ο πράκτορας ακολουθεί την πολιτική π σε κάθε βήμα στιγμή t.
- Σημειώστε ότι η συνάρτηση αξίας $v_\pi(s)$ δεν εξαρτάται από το t. Υποθέτουμε ότι το μοντέλο είναι στατικό, δηλαδή τα στατιστικά στοιχεία των ανταμοιβών καθώς και οι πιθανότητες μετάβασης από κατάσταση σε κατάσταση δεν αλλάζουν με το χρόνο.

Εξίσωση Bellman για την συνάρτηση v

- Γενική αναδρομική σχέση:

$$v_\pi(s) = \sum_a \pi(a,s) \sum_{r,s'} p(r,s'|s,a)\,[r + \gamma v_\pi(s')]$$

- Στην ειδική περίπτωση που η ανταμοιβή r είναι ντετερμινιστική συνάρτηση του ζεύγους s, a, τότε η σχέση απλοποιείται:

$$v_\pi(s) = \sum_a \pi(a,s) \left[r + \gamma \sum_{s'} p(s'|s,a)\, v_\pi(s') \right]$$

- Αν επί πλέον η επόμενη κατάσταση s' είναι επίσης ντετερμινιστική συνάρτηση του ζεύγους s, a, τότε :

$$v_\pi(s) = \sum_a \pi(a,s)[r + \gamma v_\pi(s')]$$

- Τέλος, αν η πολιτική μας είναι ντετερμινιστική επιλογή μιας ενέργειας a ως συνάρτηση της κατάστασης s τότε:

$$v_\pi(s) = r + \gamma v_\pi(s')$$

Βασικό πρόβλημα

- **Θέλουμε να βρούμε τη βέλτιστη πολιτική π που μεγιστοποιεί την αναμενόμενο κέρδος ξεκινώντας από οποιαδήποτε αρχική κατάσταση s.**
- Με άλλα λόγια, η συνάρτηση ποιότητας που πρέπει να μεγιστοποιηθεί είναι:

$$J(s) = E_\pi\{G(0)|S(0) = s\} = v_\pi(s)$$

- Η βέλτιστη πολιτική για την κατάσταση s είναι:

$$\pi^*(s) = \arg\max_\pi v_\pi(s)$$

- Και η βέλτιστη αξία της κατάστασης s είναι:

$$v^*(s) = \max_\pi v_\pi(s) = v_{\pi^*}(s)$$

Συνάρτηση Q

- Για την επίλυση του προβλήματος της βέλτιστης πολιτικής, είναι χρήσιμο να ορίσουμε τη συνάρτηση

$$q_\pi(s,a) = E_\pi\{G(t) \mid S(t) = s, A(t) = a\}$$

δηλαδή, το αναμενόμενο κέρδος αν ξεκινήσουμε από την κατάσταση s και ακολουθήσουμε τις ενέργειες a που υποδεικνύει η πολιτική π.

- Η $q_\pi(\cdot)$ είναι γνωστή ως **συνάρτηση-Q** (**Q-function**)
- Αποδεικνύεται ότι υπάρχουν οι εξής σχέσεις μεταξύ q_π και v_π

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')]$$

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$

Βέλτιστη πολιτική

- Δεδομένης της σχέσης μεταξύ $v_\pi(s)$ και $q_\pi(s,a)$: $v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$ μπορούμε να καταλήξουμε σε ένα σημαντικό συμπέρασμα όσον αφορά τη βέλτιστη πολιτική:
  - Αν υποθέσουμε ότι $q_{\pi^*}(s,a)$ είναι η συνάρτηση Q υπό τη βέλτιστη πολιτική π^*, τότε η συνάρτηση $v_\pi(s)$ μεγιστοποιείται αν

$$\pi^*(a|s) = \begin{cases} 1 & \text{αν } a = \arg\max_{a'}\left(q_{\pi^*}(s,a')\right) \\ 0 & \text{Διαφορετικά} \end{cases}$$

- Με άλλα λόγια, **η βέλτιστη πολιτική για μια δεδομένη κατάσταση s είναι να επιλέγουμε πάντα την ενέργεια a που δίνει το μέγιστο $q_{\pi^*}(s,a)$.** Επομένως:

$$v_{\pi^*}(s) = \max_{a'} q_{\pi^*}(s,a')$$

Αναδρομή υπό βέλτιστη πολιτική

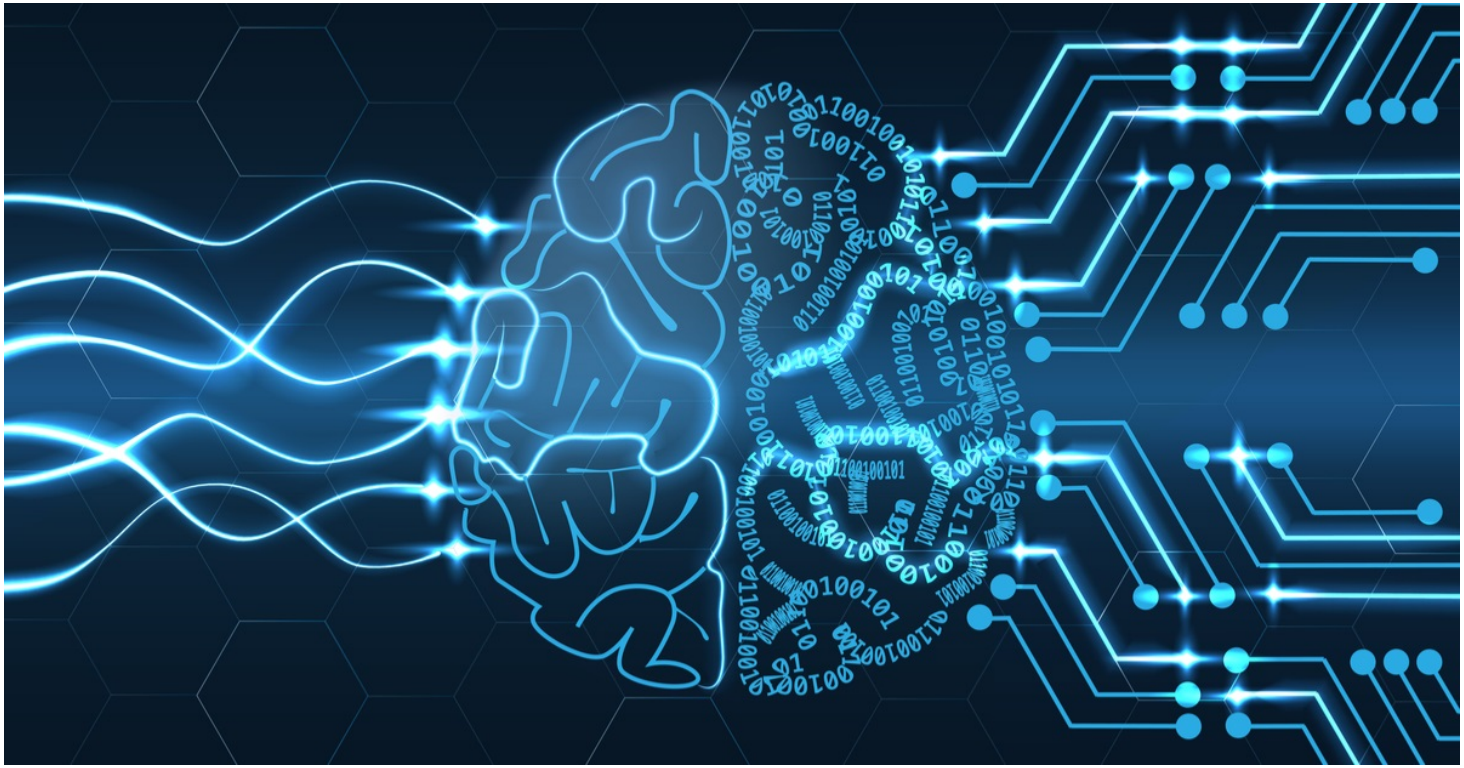
- Εάν η πολιτική είναι βέλτιστη, τότε

$$q_{\pi^*}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) v_{\pi^*}(s')$$

$$q_{\pi^*}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} q_{\pi^*}(s',a')$$

- Ο τύπος ονομάζεται **εξίσωση βελτιστοποίησης Bellman**.
- Είναι ένας αναδρομικός τύπος για το q.
- Σημειώστε ότι ο όρος $r(s,a)$ δεν εξαρτάται από την πολιτική π και μπορεί να υπολογιστεί εκ των προτέρων για κάθε ζεύγος κατάστασης s και ενέργειας a.

Δυναμικός προγραμματισμός



Dynamic Programming

Αξιολόγηση Πολιτικής με δυναμικό προγραμματισμό

- Πρόβλημα: Δεδομένης της πολιτικής $\pi(a|s)$ να υπολογιστούν οι αξίες των καταστάσεων $v_\pi(s)$.
- Θυμόμαστε τη σχέση: $v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$
- Και επίσης

$$\begin{aligned} q_\pi(s,a) &= E\{R(t+1)|\, S(t)=s, A(t)=a\} + \gamma \sum_{s'} p(s'|s,a) v_\pi(s') \\ &= \sum_{s'} [\sum_r p(s',r|s,a)\, r + \gamma \sum_r p(s',r|s,a)\, v_\pi(s')] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \end{aligned}$$

- Οπότε

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\, [r + \gamma v_\pi(s')]$$

Αξιολόγηση πολιτικής

- Ας υποθέσουμε ότι γνωρίζουμε την από κοινού πιθανότητα ανταμοιβής $R(t) = r$ και επόμενης κατάστασης $S(t+1) = s'$ με δεδομένη την προηγούμενη κατάσταση $S(t) = s$ και ενέργεια $A(t) = a$:

$$p(s', r|s, a)$$

- Από αυτό μπορούμε να βρούμε:
  - Την πιθανότητα της επόμενης κατάστασης $S(t+1) = s'$, με δεδομένη προηγούμενη κατάσταση $S(t) = s$ και την ενέργεια $A(t) = a$:

$$p(s'|s, a) = \int_r p(s', r|s, a)dr$$

  - Την πιθανότητα ανταμοιβής $R(t) = r$ με δεδομένη την προηγούμενη κατάσταση $S(t) = s,$ ενέργεια $A(t) = a$ και επόμενη κατάσταση $S(t+1) = s'$:

$$p(r|s', s, a) = \frac{p(s', r|s, a)}{\mathrm{p}(s'|s, a)}$$

Επαναληπτική αξιολόγηση πολιτικής

- Επομένως πρέπει να λύσουμε το παρακάτω σύστημα γραμμικών εξισώσεων με $K = |\mathcal{S}|$ αγνώστους $v_\pi(1), \dots, v_\pi(K)$:

$$v_\pi(s) = \sum_a \pi(a|s)' \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right]$$

- Μια προφανής προσέγγιση είναι η επίλυση του παραπάνω γραμμικού συστήματος.
- Μια εναλλακτική προσέγγιση είναι η **επαναληπτική μέθοδος**:
  - Ξεκινάμε με μια αρχική εκτίμηση των τιμών κατάστασης $v_0(1), \dots, v_0(K)$
  - Ενημερώνουμε με χρήση του ακόλουθου κανόνα:

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right]$$

Παράδειγμα: Μετακίνηση σε πλέγμα

Θεωρήστε το πρόβλημα μετακίνησης σε έναν 2-Διάστατο κόσμο που αποτελείται από 16 τετράγωνα

όπως φαίνεται στο διπλανό σχήμα.

Το πράσινο τετράγωνο είναι τερματικό.

Σε κάθε θέση υπάρχουν 4 δυνατές ενέργειες:

- Κίνηση προς τα πάνω
- Κίνηση προς τα κάτω
- Κίνηση προς τα δεξιά
- Κίνηση προς τα αριστερά

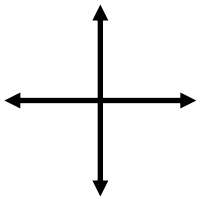
Ο πράκτορας πρέπει:

- Να αποφύγει το κόκκινο τετράγωνο
- Να καταλήξει στο πράσινο τετράγωνο

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Παράδειγμα: Μετακίνηση σε πλέγμα

Κάθε μετακίνηση έχει ανταμοιβή $r = -1$ (διότι σπαταλήθηκε ενέργεια)
Μετακίνηση εκτός ταμπλό επαναφέρει τον πράκτορα στην αρχική του θέση. Πχ, η ανταμοιβή r και η επόμενη κατάσταση s' αν βρισκόμαστε στη θέση 8 και κινηθούμε δεξιά είναι:

$$r(s = 8, a = \text{right}) = -1$$
$$s'(s = 8, a = \text{right}) = 9$$

Επίσης

$$r(s = 1, a = \text{up}) = -1$$
$$s'(s = 1, a = \text{up}) = 1$$

Το κόκκινο τετράγωνο είναι παγίδα.
Μετακίνηση από αυτό έχει ανταμοιβή $r = -10$

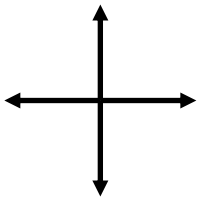
$$r(s = 6, a = \text{left}) = -10$$

Μετακίνηση στο πράσινο τετράγωνο δίνει $r = 0$.
Καθώς το τετράγωνο αυτό είναι τερματικό κάθε μετακίνηση από αυτό επαναφέρει τον πράκτορα σε αυτό.

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Παράδειγμα: Μετακίνηση σε πλέγμα

Αποτιμούμε την πολιτική όπου όλες οι ενέργειες είναι ισοπίθανες για κάθε s :

$\pi(\text{up}|s) = \pi(\text{down}|s) = \pi(\text{left}|s) = \pi(\text{right}|s) = \frac{1}{4}$.

Στην περίπτωση αυτή μια συγκεκριμένη ενέργεια a
Δίνει μια συγκεκριμένη ανταμοιβή r
Και φέρνει τον πράκτορα σε μια νέα κατάσταση s'.
Αυτό απλοποιεί την φόρμουλα:

$$v_{k+1}(s) = \sum_a \pi(a|s)[r + \gamma v_k(s')]$$

Αποτελέσματα επαναληπτικής αποτίμησης: $(\gamma = 0.9)$

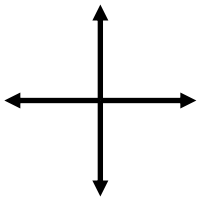
- Επανάληψη 1: $V =$

-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-10.0	-1.0
-1.0	-1.0	-1.0	-0.8
-1.0	-1.0	-0.8	0.0

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Παράδειγμα: Μετακίνηση σε πλέγμα

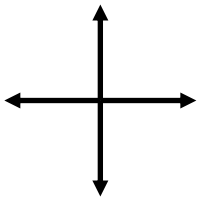
- Επανάληψη 2: $V =$

-1.9	-1.9	-3.9	-1.9
-1.9	-3.9	-10.9	-3.9
-1.9	-1.9	-3.8	-1.4
-1.9	-1.8	-1.4	0.0

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Παράδειγμα: Μετακίνηση σε πλέγμα

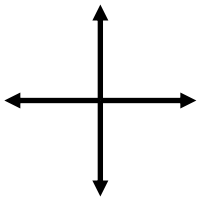
- Επανάληψη 3: $V =$

-2.7	-3.6	-5.2	-3.6
-3.2	-4.7	-13.5	-5.1
-2.7	-3.6	-4.5	-2.8
-2.7	-2.6	-2.3	0.0

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Παράδειγμα: Μετακίνηση σε πλέγμα

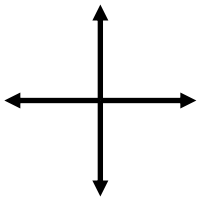
- Επανάληψη 10: $V =$

-8.3	-9.7	-12.1	-10.4
-8.4	-10.8	-19.4	-11.3
-7.6	-8.4	-9.6	-6.7
-7.0	-6.9	-5.4	0.0

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Παράδειγμα: Μετακίνηση σε πλέγμα

- Σταθερή μετά από την επανάληψη 50.

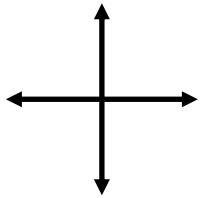
- Επανάληψη ∞: $V =$

-12.8	-14.0	-16.3	-14.5
-12.7	-15.0	-23.2	-14.9
-11.6	-12.1	-12.7	-9.0
-10.7	-10.1	-7.6	0.0

Καταστάσεις:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ενέργειες:



Επιλογή πολιτικής

- Η αξιολόγηση των καταστάσεων μας επιτρέπει να εκτιμήσουμε τις βέλτιστες δράσεις σε μια δεδομένη κατάσταση.
- Ας επικεντρωθούμε τώρα σε ντετερμινιστικές πολιτικές όπου το π δεν είναι πλέον μια πιθανότητα, αλλά μια συνάρτηση που αντιστοιχεί καταστάσεις σε ενέργειες: $a = \pi(s)$
- **Θεώρημα βελτίωσης πολιτικής:**

Έστω π και π' είναι δύο ντετερμινιστικές πολιτικές, έτσι ώστε για όλες τις καταστάσεις $s \in \mathcal{S}$ έχουμε:

$$q_\pi\big(s, \pi'(s)\big) \geq v_\pi(s)$$

Τότε οι τιμές των καταστάσεων υπό την πολιτική π' είναι μεγαλύτερες ή ίσες από τις αξίες των καταστάσεων υπό την π:

$$v_{\pi'}(s) \geq v_\pi(s), \text{ για κάθε } s \in \mathcal{S}$$

Άπληστη επιλογή πολιτικής

- Σε κάθε κατάσταση επιλέγουμε την ενέργεια που οδηγεί στην κατάσταση με την μέγιστη τιμή:

$$\pi(s) = \arg\max_{a} v(s'(a, s))$$

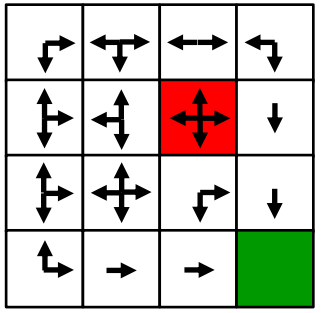
- Αν δύο ή περισσότερες ενέργειες οδηγούν σε καταστάσεις με ίσες τιμές επιλέγουμε τυχαία.

- **Παράδειγμα: Μετακίνηση σε πλέγμα**

- Επανάληψη 1: $V =$

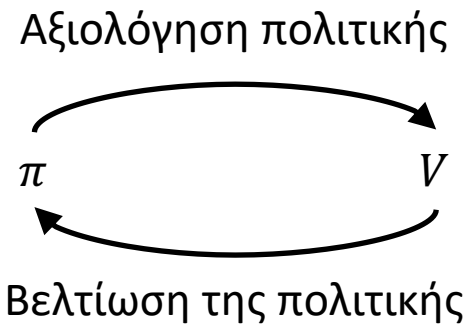
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-10.0	-1.0
-1.0	-1.0	-1.0	-0.8
-1.0	-1.0	-0.8	0.0

Άπληστη Πολιτική



Επαναλήψεις πολιτικής

- Εκτέλεση βρόχου που εναλλάσσεται ανάμεσα σε:
  - Αξιολόγηση πολιτικής
  - Βελτίωση πολιτικής



- Επαναλαμβάνουμε το βρόχο αξιολόγησης για L βήματα πριν εφαρμόσουμε τη βελτίωση της πολιτικής.
- Στην ειδική περίπτωση όπου $L = 1$, η διαδικασία ονομάζεται "Επανάληψη τιμής".

Παράδειγμα: Επαναλήψεις πολιτικής στο πλέγμα

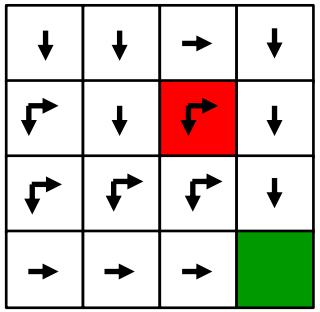
- Μετά από δύο επαναλήψεις:

$$\pi^{(0)} \rightarrow V^{(0)} \rightarrow \pi^{(1)} \rightarrow V^{(1)} \rightarrow \pi^{(2)}$$

- $V^{(2)} =$

-4.1	-4.7	-5.2	-1.9
-3.4	-2.7	-10.9	-1.0
-2.7	-1.9	-1.0	0.0
-1.9	-1.0	0.0	0.0

Πολιτική $\pi^{(2)}$:



- Η πολιτική $\pi^{(2)}$ είναι η βέλτιστη: Ξεκινώντας από κάθε κατάσταση, οι ενέργειες παίρνουν το συντομότερο μονοπάτι προς το πράσινο τετράγωνο, αποφεύγοντας το κόκκινο τετράγωνο.

Δυναμικός Προγραμματισμός:

πλεονεκτήματα και μειονεκτήματα

- Ισχυρή προσέγγιση.
- Πάντα οδηγεί σε βελτίωση της πολιτικής.
- Ωστόσο, για να εφαρμόσουμε τη μέθοδο δυναμικού προγραμματισμού χρειαζόμαστε εκτιμήσεις όλων των πιθανοτήτων

$$p(s', r|s, a)$$

- Αυτό δεν είναι πάντα εύκολο ή απλό να γίνει.
- Σε πολλά προβλήματα ο αριθμός των καταστάσεων σε συνδυασμό με τις πιθανές ενέργειες είναι πολύ μεγάλος.

Η προσέγγιση Monte Carlo

- Γενική έννοια του πειράματος Monte Carlo:
- Εκτέλεση πολλών προσομοιώσεων ενός πειράματος. Προφανώς πρέπει να υπάρχει κάποια τυχαιότητα στα πειράματα.
- Συλλογή στατιστικών στοιχείων μετά την εκτέλεση N πειράματα και λήψη απόφασης ή εκτίμησης με βάση τα στατιστικά. Όσο μεγαλύτερη είναι η τιμή του N τόσο πιο αξιόπιστες είναι οι εκτιμήσεις.
- **Απλό παράδειγμα**: Θέλουμε να ελέγξουμε αν ένα ζάρι είναι «πειραγμένο». Δηλαδή θέλουμε να ελέγξουμε την υπόθεση $P(Dice = 1) = \cdots = P(Dice = 6)$.
- Ρίχνουμε το ζάρι N φορές και μετράμε πόσες φορές το αποτέλεσμα ήταν $1, 2, \ldots, 6$. Οι εκτιμήσεις των πιθανοτήτων είναι $\hat{P}(Dice = i) = \frac{N_{Dice=i}}{N}$. Ελέγχουμε αν είναι ίσες (ή περίπου ίσες).

Αξιολόγηση πολιτικής Monte Carlo

- Η αξιολόγηση της πολιτικής μπορεί να πραγματοποιηθεί με τη μέθοδο Monte Carlo, εάν το σύστημα έχει πεπερασμένο ορίζοντα (δηλαδή το πείραμα δεν τρέχει για πάντα).
- Το βασικό πείραμα Μόντε Κάρλο (MC) καλείται **Επεισόδιο**:

1. Έναρξη από τυχαία αρχική κατάσταση
2. Επιλογή ενεργειών σύμφωνα με την πολιτική
3. Μετάβαση στην επόμενη κατάσταση σύμφωνα με την πιθανότητα μετάβασης
4. Επαναλάβετε τα βήματα 2-3 μέχρι να φτάσετε σε κατάσταση τερματισμού

Τυπική εφαρμογή είναι τα παιχνίδια (πχ. black-jack, Τάβλι, Κλπ).

Αξιολόγηση πολιτικής Monte Carlo

Είσοδος: π: πολιτική που πρέπει να αξιολογηθεί

Αρχικοποίηση:

- $\hat{v}_\pi$: τυχαίες αρχικές αξίες καταστάσεων
- $Return(s)$: κενή λίστα κερδών για κάθε κατάσταση $s \in S$

- Βρόχος:

  Δημιουργία επεισοδίου με χρήση πολιτικής π

  Για κάθε κατάσταση s που εμφανίζεται στο επεισόδιο:

  $G \leftarrow$ κέρδος που συμβαίνει μετά την πρώτη εμφάνιση του s

  Προσθήκη του G στη λίστα $Return(s)$

  $\hat{v}_\pi(s) \leftarrow Average\big(Return(s)\big)$

Βελτίωση πολιτικής Monte Carlo

- Στη βελτίωση της πολιτικής, αντί να υπολογιστεί η αξία των κατάστασεων, είναι πιο βολικό να εκτιμηθεί η συνάρτηση Q, δεδομένου ότι η πολιτική θα βελτιωθεί με την επιλογή του μέγιστου $q_\pi(s,a)$.

- Αρχικοποίηση: $\pi(s)$: Αυθαίρετη; $\hat{q}(s,a)$: αυθαίρετη για κάθε ζεύγος $s \in \mathcal{S}$ and $a \in \mathcal{A}$; $Return(s,a)$: κενή λίστα κερδών για κάθε ζεύγος $s \in \mathcal{S}$ and $a \in \mathcal{A}$
- Βρόχος:

  Δημιουργία επεισοδίου με τυχαία αρχική κατάσταση s_0 και ενέργεια a_0 με βάση την πολιτική π

  Για κάθε ζεύγος (s,a) που εμφανίζεται στο επεισόδιο:

  $G \leftarrow$ κέρδος που συμβαίνει μετά την πρώτη εμφάνιση του ζεύγους s,a

  Προσθήκη G στη λίστα $Return(s,a)$

  $\hat{q}_\pi(s,a) \leftarrow Average\big(Return(s,a)\big)$

  Για κάθε s που εμφανίζεται στο επεισόδιο:

  $$\pi^{new}(s) = \arg\max_a \hat{q}_\pi(s,a)$$

Προσέγγιση χρονικής διαφοράς

- Ένα πρόβλημα με τις απλές μεθόδους Monte Carlo είναι ότι πρέπει να φτάσουμε στο τέλος του επεισοδίου πριν ενημερώσουμε την αξία των καταστάσεων.
- Θυμόμαστε ότι

$$v(s) = E_{\pi}\{R(t+1) + \gamma v(S(t+1)) | \, S(t) = s\}$$

- Μπορούμε να επιταχύνουμε τη διαδικασία αξιολόγησης κάθε φορά που είμαστε σε κατάσταση s, εκτελούμε ενέργεια a, παρατηρήστε μια ανταμοιβή r και προχωράμε σε μια μεταγενέστερη κατάσταση s', με την άμεση ενημέρωση του $v(s)$ μέσω της ακόλουθης προσέγγισης:

$$\hat{v}(s) \leftarrow Average\big(R(t+1) + \gamma v(S(t+1)) | \, S(t) = s\big)$$

- Αυτή η μέθοδος ονομάζεται μέθοδος TD(0), δεδομένου ότι κοιτάμε μόνο ένα βήμα μπροστά στην κατάσταση $S(t+1)$.

Αλγόριθμος TD(0)

- Η μέθοδος TD(0) για την αξιολόγηση της πολιτικής θα είναι η ακόλουθη:
- Για κάθε επεισόδιο:
  1. Έναρξη σε τυχαία κατάσταση s_0
  2. Εκτελούμε ενέργεια a καταγράφουμε την ανταμοιβή r και προχωράμε σε κατάσταση s ′
  3. Υπολογισμός μέσου όρου του όρου $r + \gamma v(s')$:
     - Αύξηση μετρητή $n(s) \leftarrow n(s) + 1$
     - Ενημέρωση: $v(s) \leftarrow \frac{n(s)-1}{n(s)} v(s) + \frac{1}{n(s)} \left(r + \gamma v(s')\right)$

     ή ισοδύναμα, $v(s) \leftarrow v(s) + \beta\left(r + \gamma v(s') - v(s)\right)$

     με $\beta = 1/n(s)$.
  4. Θέτουμε $s \leftarrow s'$ και επαναλαμβάνουμε τα βήματα 2-3 μέχρι να φτάσουμε σε τερματική κατάσταση.
- Επαναλάβετε για όσο το δυνατόν περισσότερα επεισόδια.

Αλγόριθμος TD(0)

- Είσοδος: π την πολιτική που πρέπει να αξιολογηθεί
- Αρχικοποίηση: $\hat{v}_\pi(s)$=Αυθαίρετο (πχ $\hat{v}_\pi(s) = 0$, για κάθε $s \in \mathcal{S}$)
- Για κάθε επεισόδιο:

  Θέσε $s = s_0$

  Για κάθε βήμα του επεισοδίου:

  Εκτέλεση ενέργειας $a = \pi(s)$

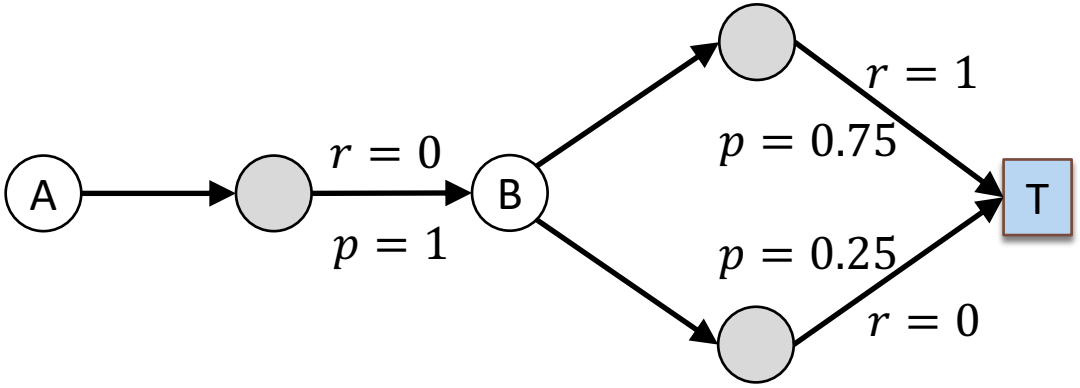
  Παρατηρούμε την ανταμοιβή r και την επόμενη κατάσταση s'

  $$\hat{v}_\pi(s) \leftarrow \hat{v}_\pi(s) + \beta(r + \gamma\hat{v}_\pi(s') - \hat{v}_\pi(s))$$

Temporal Difference

Monte Carlo vs. TD(0)

- Εξετάστε το ακόλουθο απλό διάγραμμα ενεργειών καταστάσεων



- Έστω ότι παρατηρούμε τα ακόλουθα 8 επεισόδια:

1) $A, r = 0, B, r = 0$
2) $B, r = 1$
3) $B, r = 1$
4) $B, r = 1$
5) $B, r = 1$
6) $B, r = 1$
7) $B, r = 0$
8) $B, r = 1$

Monte Carlo vs. TD(0)

- Εάν χρησιμοποιήσουμε TD(0) για να εκτιμήσουμε τις αξίες των καταστάσεων θα βρούμε

$$\hat{v}(B) = 0.75$$

διότι 75% του χρόνου παίρνουμε ανταμοιβή $r = 1$.

$$\hat{v}_{TD}(B) = Average\big(r + \hat{v}_{TD}(T)\big) = Average(r + 0) = 0.75$$

- Στη συνέχεια, αυτή η τιμή θα μεταδοθεί στην κατάσταση Α, επειδή

$$\hat{v}_{TD}(A) = Average\big(r + \hat{v}_{TD}(B)\big) = Average\big(0 + \hat{v}_{TD}(B)\big) = 0.75$$

(υποθέσαμε $\gamma = 1$).

- Αυτό είναι λογικό δεδομένου ότι η κατάσταση Α οδηγεί πάντα στην κατάσταση Β χωρίς πρόσθετη ανταμοιβή. Ως εκ τούτου, οι δύο τιμές θα πρέπει να είναι ίσες: $v(A) = v(B)$.

Monte Carlo vs. TD(0)

- Αν χρησιμοποιήσουμε την προσέγγιση Monte Carlo, τότε

$$\hat{v}_{MC}(A) = 0$$

επειδή έχουμε μόνο ένα επεισόδιο που εμπλέκει την κατάσταση A και το οποίο δίνει κέρδος $G = 0 + 0 = 0$.

- Η εκτίμηση $v(B)$ θα είναι ακριβής

$$\hat{v}_{MC}(B) = 0.75$$

διότι 6 από τα 8 επεισόδια που εμπλέκουν το B δίνουν κέρδος $G = 1$ ενω 2 από τα 8 δίνουν κέρδος $G = 0$.

- Η μέθοδος MC είναι πιο πιστή στα δεδομένα εκπαίδευσης και έχει φτωχότερη επίδοση στα δεδομένα ελέγχου.

Άλλες εφαρμογές

- Παιχνίδι “Go”. Πρόγραμμα “*AlphaGo*” της Deep Mind: μέθοδος Monte Carlo με βαθύ συνελικτικό δίκτυο για τη μοντελοποίηση της αξίας καταστάσεων. Νίκησε τον Fan Hui 5/0 και τον Lee Sedol 4/1. Η νεότερη έκδοση “*AlphaGo Zero*” νίκησε το “*AlphaGo*” 100/0 και το “*AlphaGo Master*” 89/11.
- Σύσταση περιεχομένου Web: Ποια σελίδα να συστήσουμε μεταξύ n διαφορετικών σελίδων? → Πρόβλημα πολλαπλών μονόχειρων ληστών. Ανταμοιβή = Click-through rate = [αριθμός κλικ στη σελίδα]/[αριθμός επισκέψεων]
- Βελτιστοποίηση ελεγκτών μνήμης (Βελτιστοποίηση DRAM)
- Παίξιμο βίντεο-παιχνιδιών σε επίπεδο αντίστοιχο ή καλύτερο του ανθρώπου.

Άλλες εφαρμογές

- Ρομποτική
- (*Έλεγχος βάδισης τετράποδου*) Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion by Nate Kohl and Peter Stone
- (*Πιάσιμο μπάλας από τετράποδο)* Learning Ball Acquisition on a Physical Robot by Peggy Fidelman and Peter Stone
- (*Air Hockey*) Learning from Observation Using Primitives, and particularly the movie of a humanoid robot playing air hockey. An example paper.
- (*Active Sensing*) Active Sensing Using Reinforcement Learning by Cody Kwok and Dieter Fox.

Άλλες εφαρμογές

- Έλεγχος
- (Έλεγχος ελικοπτέρων) Inverted autonomous helicopter flight via reinforcement learning, by Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger and Eric Liang. In International Symposium on Experimental Robotics, 2004.
- Autonomous helicopter control using Reinforcement Learning Policy Search Methods, by J.A. Bagnell and J. Schneider. In Proceedings of the International Conference on Robotics and Automation, 2001.

Other applications

- Επιχειρησιακή Έρευνα
- (*Τιμολόγηση*) Opportunities and Challenges in Using Online Preference Data for Vehicle Pricing: A Case Study at General Motors by P. Rusmevichientong, J. A. Salisbury, L. T. Truss, B. Van Roy, and P. W. Glynn.
- (*Δρομολόγηση οχημάτων*) Scaling Average-reward Reinforcement Learning for Product Delivery by S. Proper and P. Tadepalli.
- (*Στοχευμένο μάρκετινγκ*) Cross Channel Optimized Marketing by Reinforcement Learning, by Naoki Abe, Naval Verma, Chid Apte and Robert Schroko, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2004.

Άλλες εφαρμογές

- Παιχνίδια
- (*Τάβλι*) Temporal difference learning and TD-Gammon by Gerald Tesauro, Communications of the ACM, 38(3), March 1995.
- (*Πασιέντζα*) Solitaire: Man Versus Machine, by X. Yan, P. Diaconis, P. Rusmevichientong, and B. Van Roy, to appear in Advances in Neural Information Processing Systems 17, MIT Press, 2005.
- (*Σκάκι*) The KnightCap program, which went from a rating of 1600 to a rating of 2100 by altering its heuristic evaluation function using TD-lambda. pdf
- (*Ντάμα*) Temporal Difference Learning Applied to a High-Performance Game-Playing Program by Jonathan Schaeffer, Markian Hlynka, and Vili Jussila, International Joint Conference on Artificial Intelligence (IJCAI), pp. 529-534, 2001.

Άλλες εφαρμογές

- Human Computer Interaction
- (*Συστήματα Προφορικού Διαλόγου*) Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. S. Singh, D. Litman, M. Kearns and M. Walker. In Journal of Artificial Intelligence Research (JAIR), Volume 16, pages 105-133, 2002
- (*Software Agent in MOOs*) Cobot in LambdaMOO: An Adaptive Social Statistics Agent. C. Isbell, M. Kearns, S. Singh, C. Shelton, P. Stone and D. Korman.

Άλλες εφαρμογές

- Οικονομικά
- (*Trading*) Learning to Trade via Direct Reinforcement. John Moody and Matthew Saffell, IEEE Transactions on Neural Networks, Vol 12, No 4, July 2001.
- Σύνθετες προσομοιώσεις
- (*Robot Soccer*) Scaling Reinforcement Learning toward RoboCup Soccer, by Peter Stone and Richard S. Sutton, Proceedings of the Eighteenth International Conference on Machine Learning, pp. 537–544, Morgan Kaufmann, San Francisco, CA, 2001.

Βιβλιογραφία

- [1] Κ. Διαμαντάρας, Δ. Μπότσης, Μηχανική Μάθηση, Εκδόσεις Κλειδάριθμος, 2019.
- [2] Διαφάνειες των συγγραφέων για το σύγγραμμα [1].