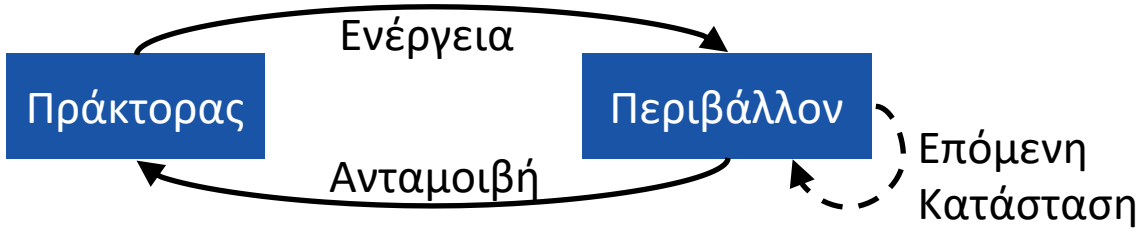
Μηχανική μάθηση

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

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.

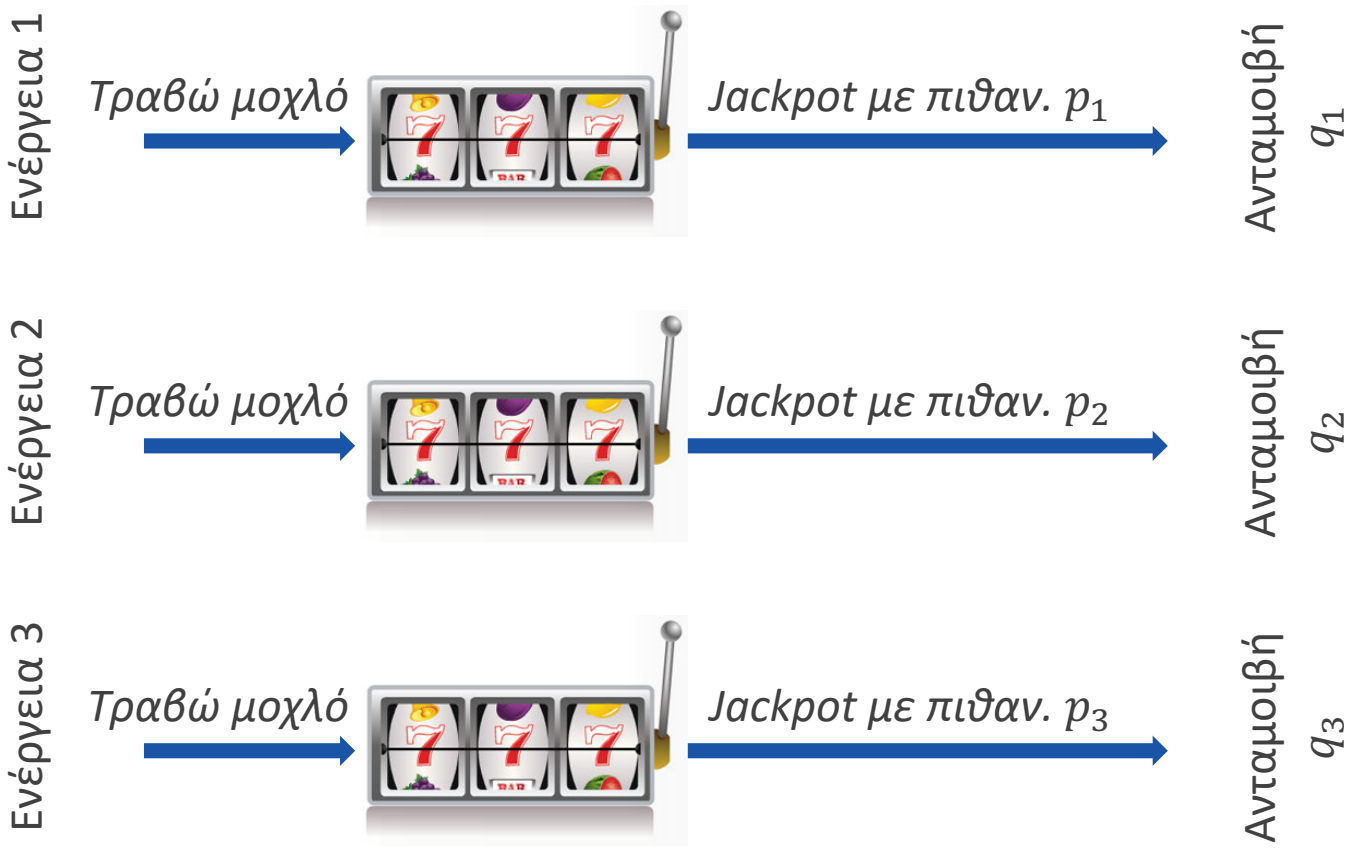
Άλλες εφαρμογές

- 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.

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 $a = 0$:

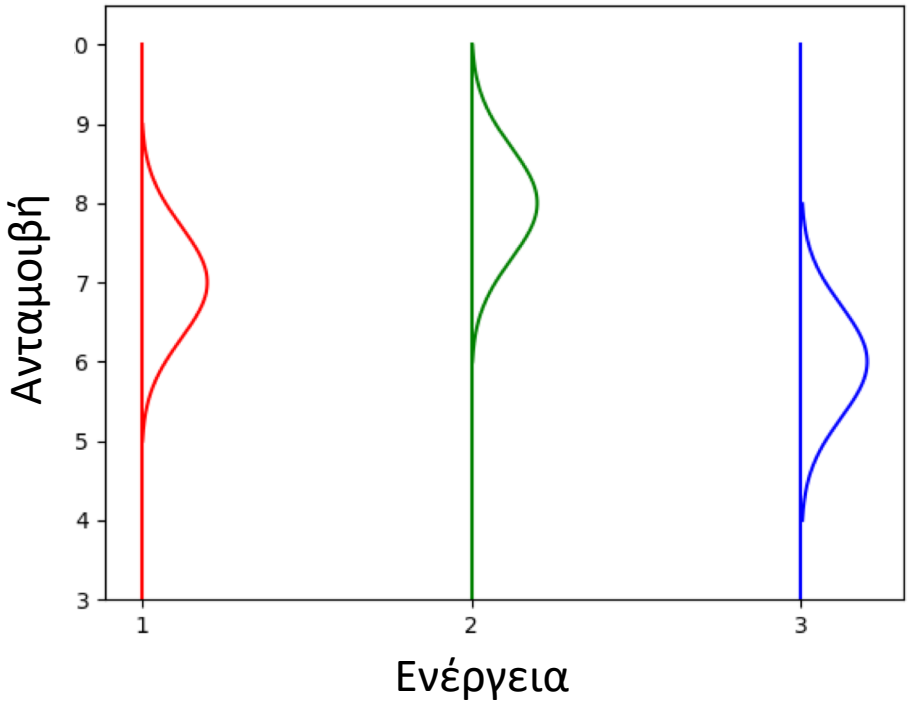
$q(0) \sim N(\mu = 7, \sigma^2 = 1)$

- Ανταμοιβή για $a = 1$:

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

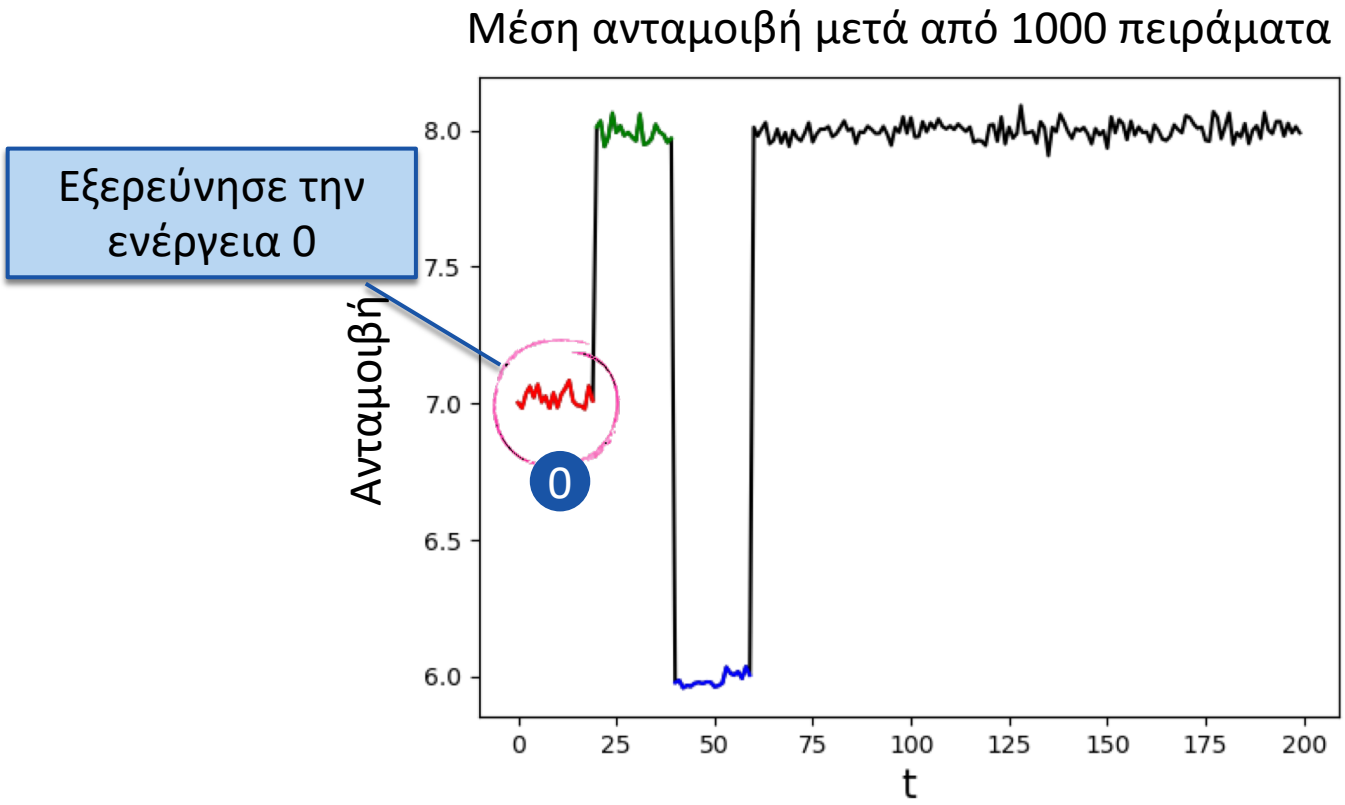
- Ανταμοιβή για $a = 2$:

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



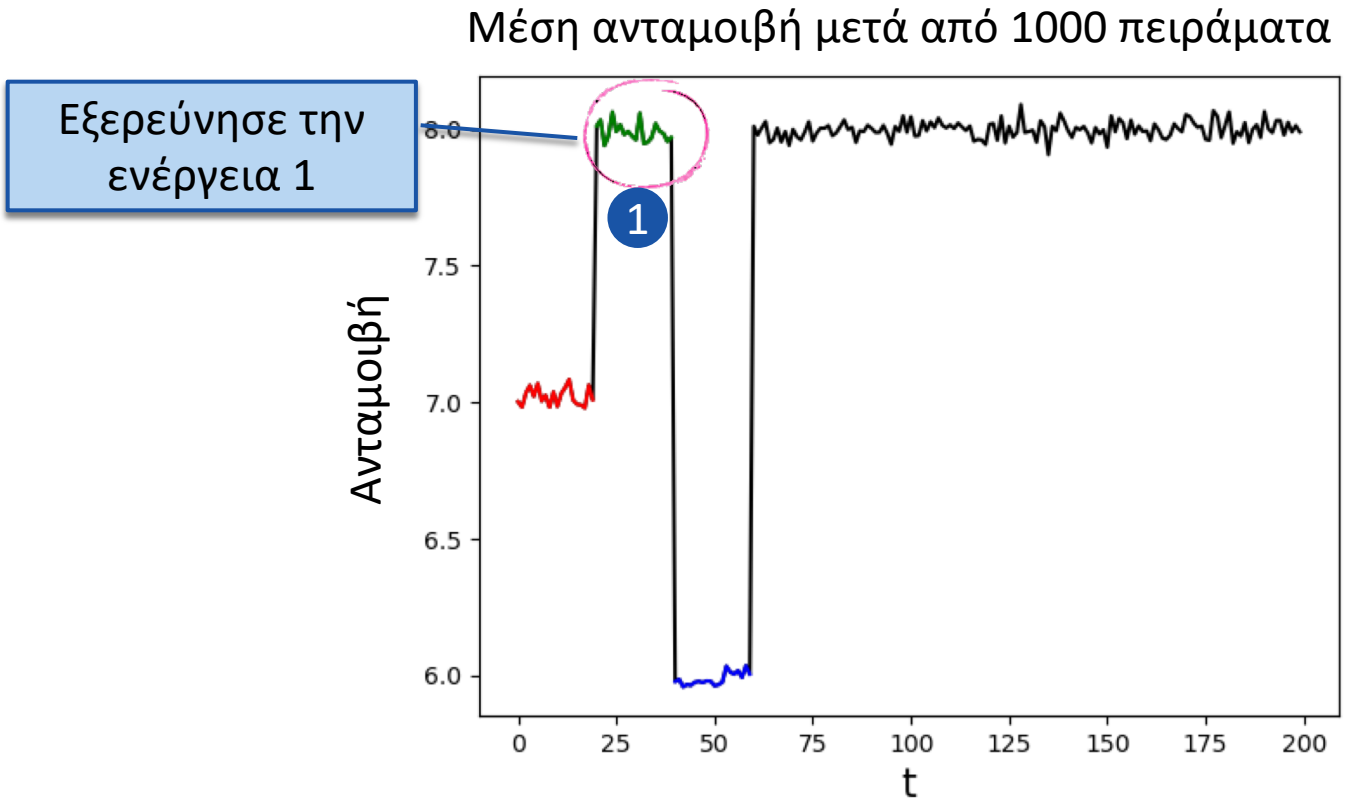
Παράδειγμα: 3 bandits

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



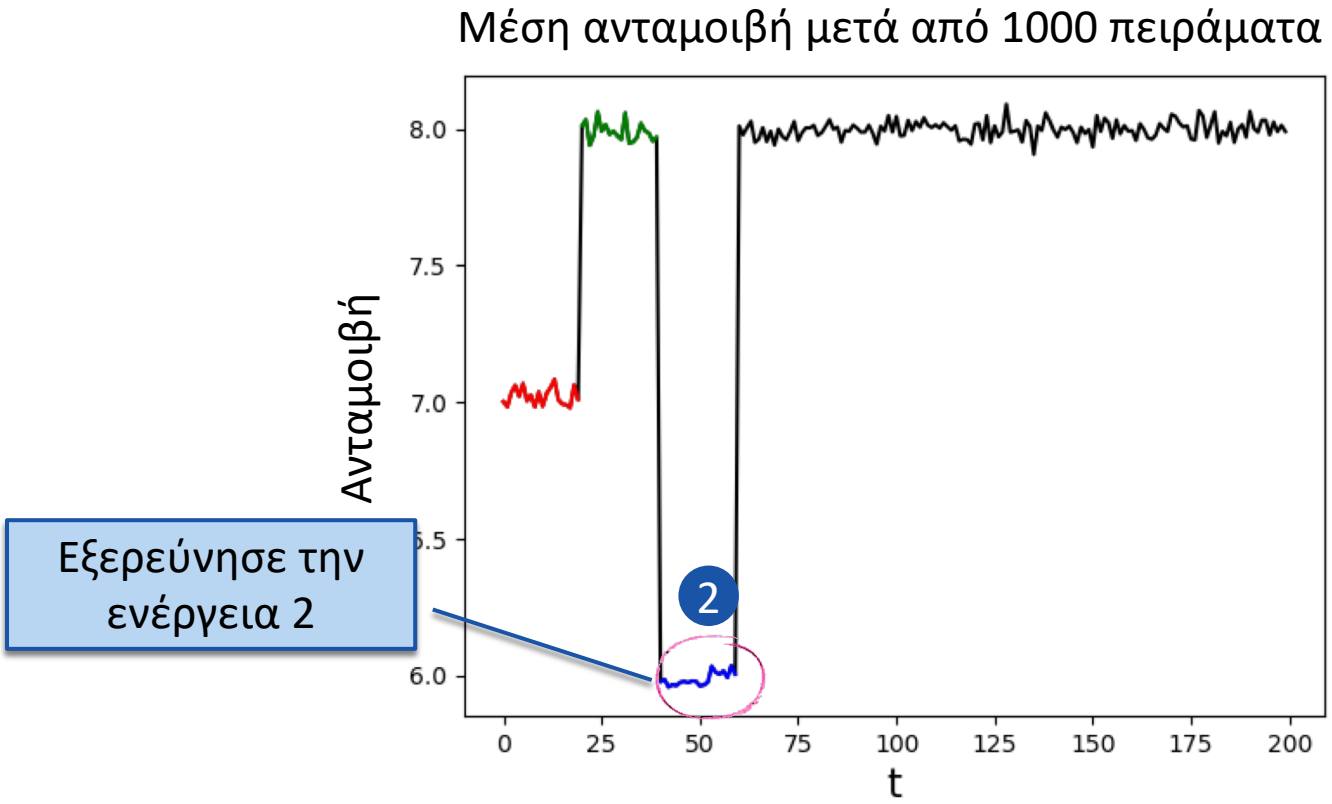
Παράδειγμα: 3 bandits

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



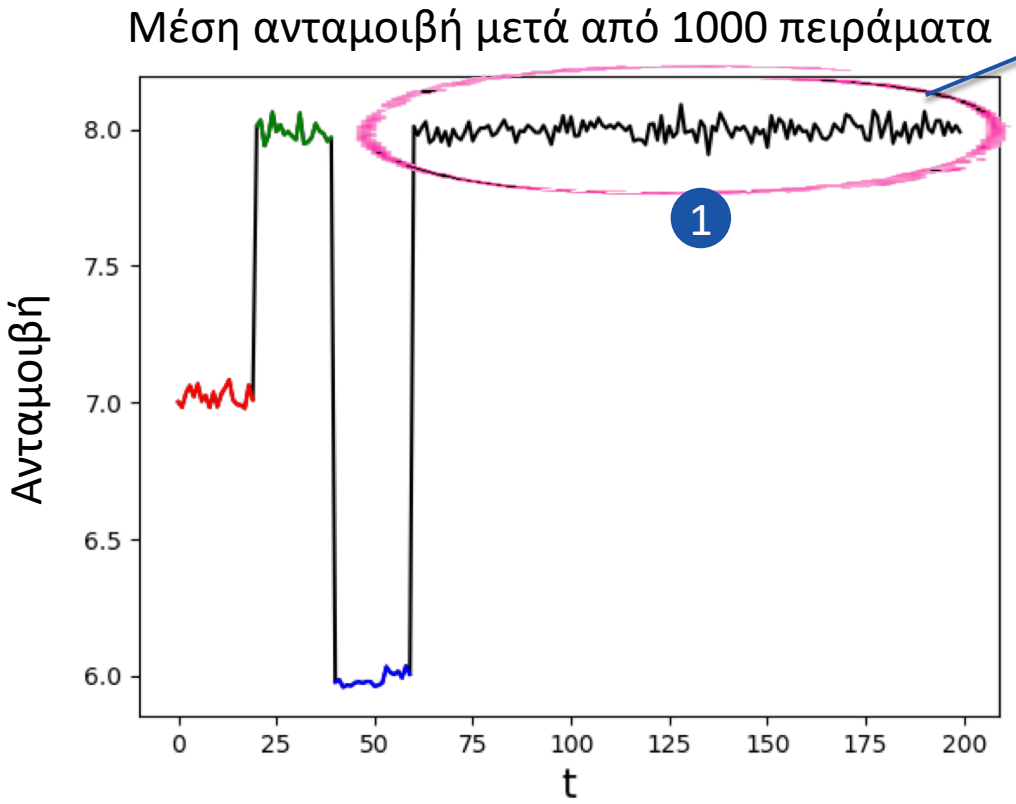
Παράδειγμα: 3 bandits

- 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 $\hat{\mu}(a)$:

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

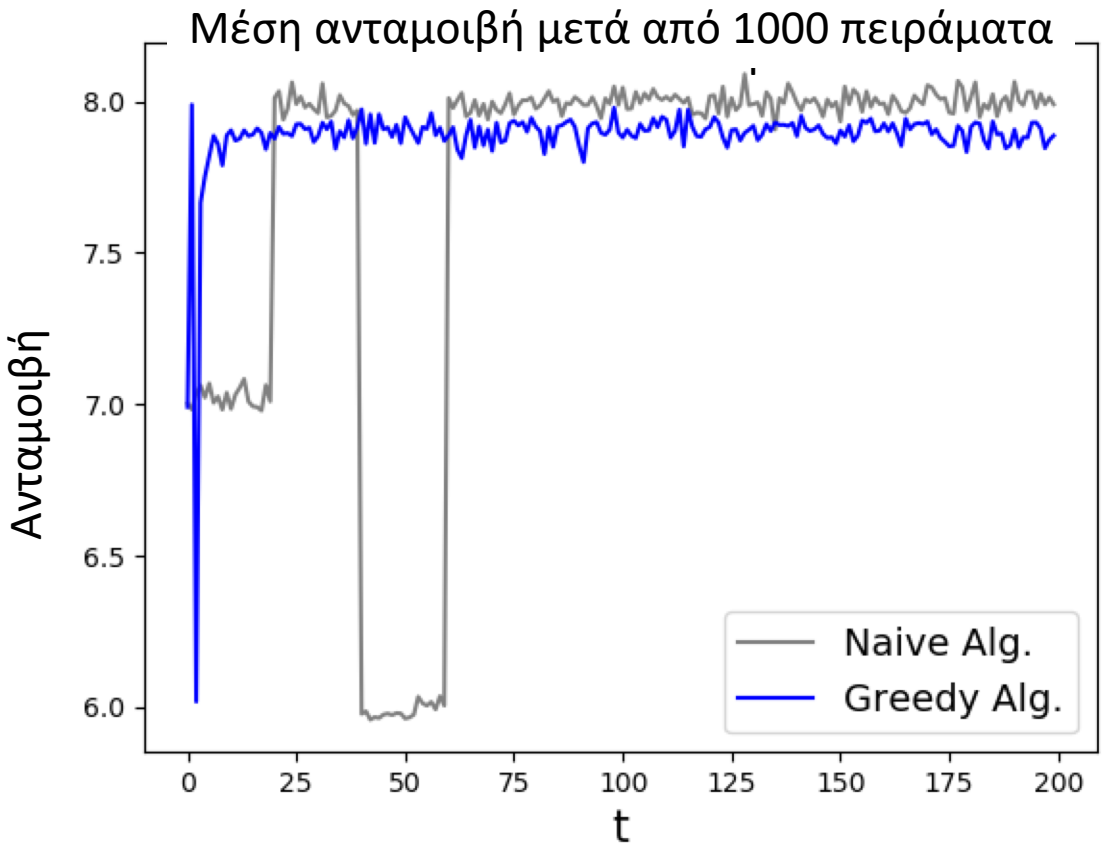
    (σε περίπτωση ισοπαλίας επέλεξε τυχαία)
  - Κατάγραψε την ανταμοιβή $R(t)$
  - Ενημέρωσε τον μετρητή:

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

  - 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:

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

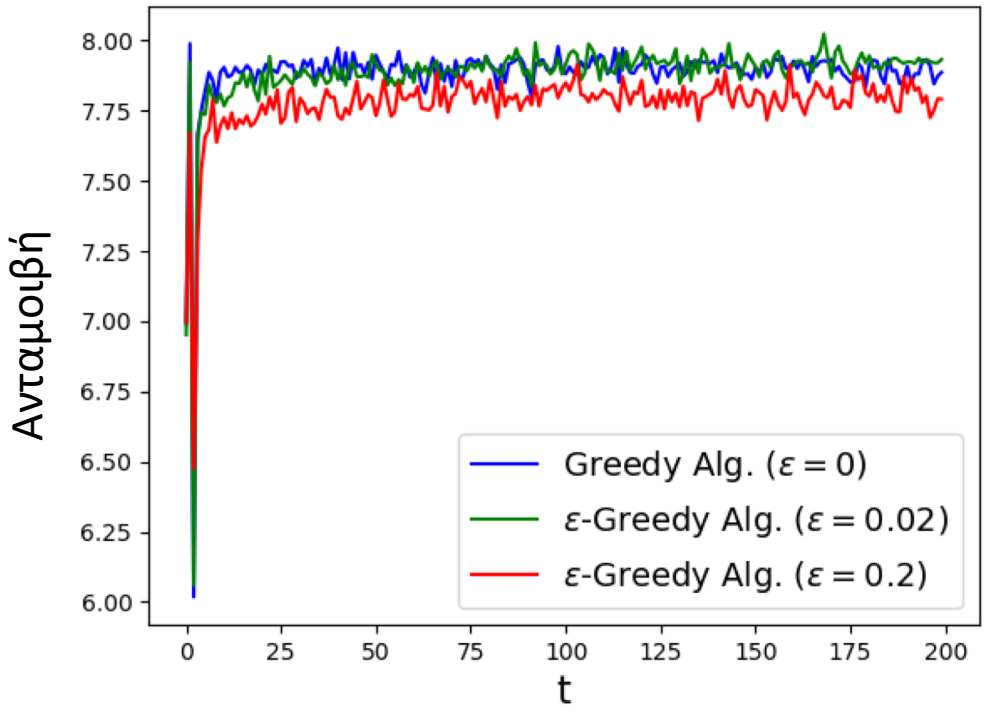
  - Ενημέρωσε την αναμενόμενη ανταμοιβή για την ενέργεια a:

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

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

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

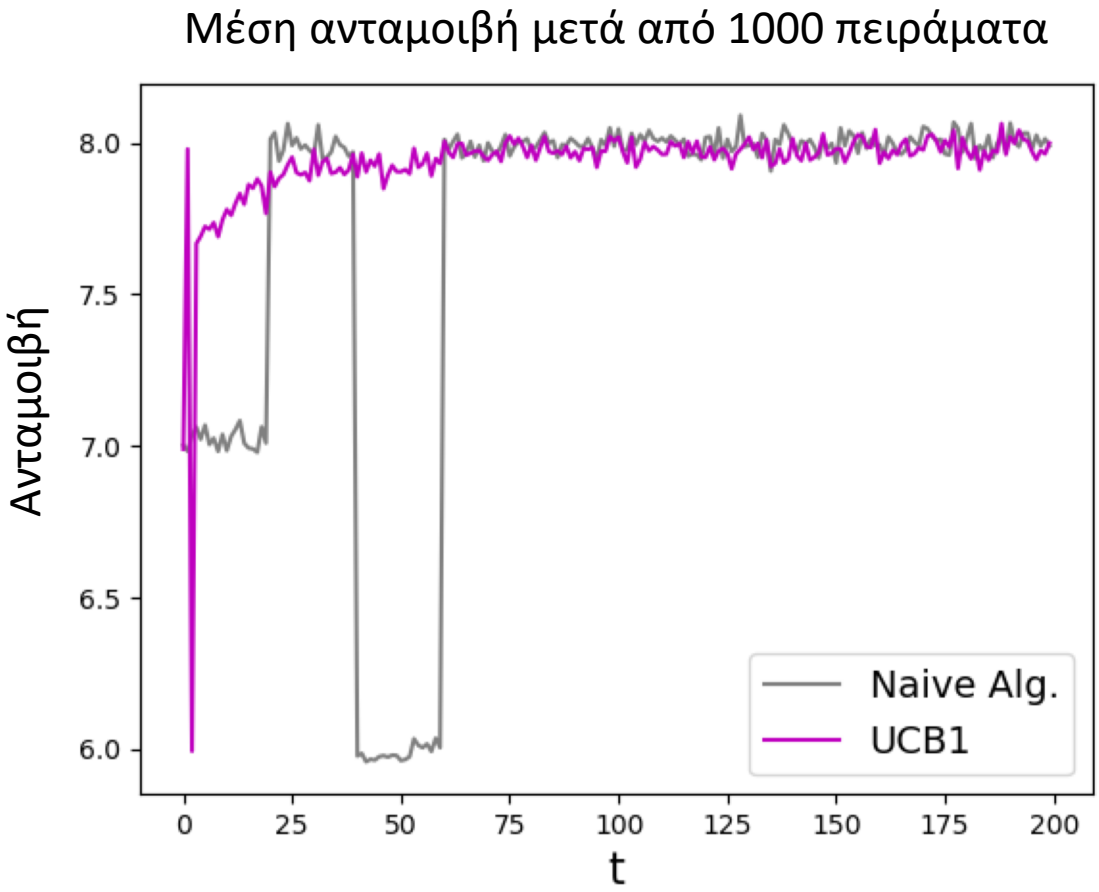
Bonus

  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

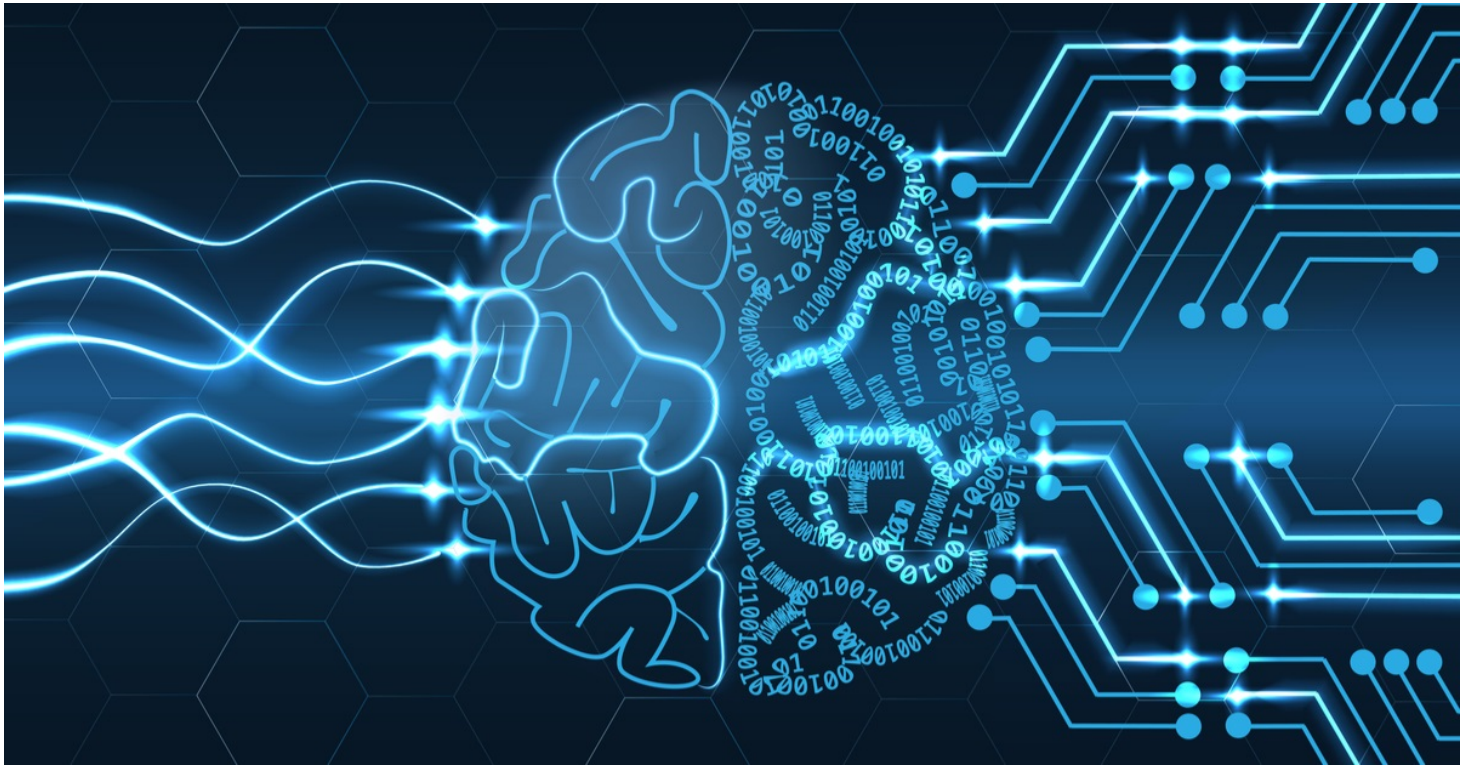
P. Auer, N. Cesa-Bianchi, P. Fischer, “Finite-time Analysis of the Multiarmed Bandit Problem”, Machine Learning, 47, 235–256, 2002

UCB1 εναντίον Απλοϊκής μεθόδου



UCB1 vs. ε-Greedy

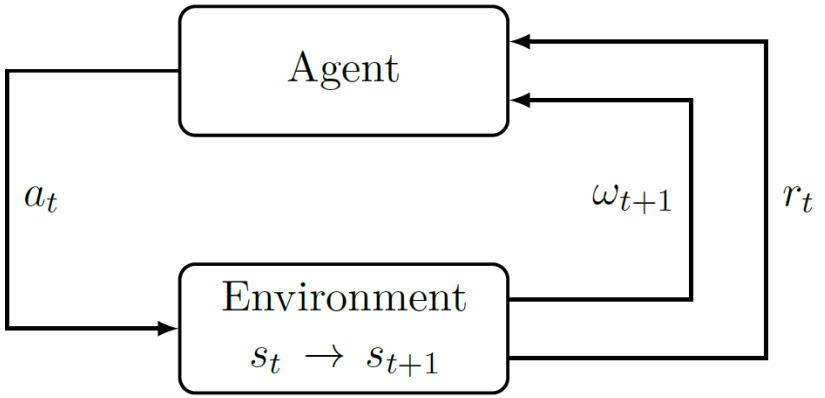
Μαρκοβιανές Διαδικασίες αποφάσεων



Markov Decision Processes (MDP)

Τυπικός ορισμός RL μάθησης

- Ένα πρόβλημα RL ορίζεται ως μια *στοχαστική διαδικασία ελέγχου διακριτού χρόνου*, όπου ο πράκτορας:
  1. Τη *χρονική στιγμή* t επιλέγει τη **δράση** a_t
  2. Το *περιβάλλον* του πράκτορα μεταβαίνει από την **κατάσταση** s_t στην **κατάσταση** s_{t+1}
  3. Λαμβάνει **ανταμοιβή** r_t
  4. Κάνει την **παρατήρηση** ω_{t+1}
- Στην **αρχή** του **χρόνου**, ο πράκτορας βρίσκεται στην *κατάσταση* a_0 και κάνει την *αρχική παρατήρηση* ω_0



Μαρκοβιανή Ιδιότητα

- Μια στοχαστική διαδικασία ελέγχου διακριτού χρόνου έχει τη **μαρκοβιανή ιδιότητα** (*markovian property*) αν ισχύει
  - $\mathbb{P}(\omega_{t+1}|\omega_t, \alpha_t) = \mathbb{P}(\omega_{t+1}|\omega_t, \alpha_t, \dots, \omega_0, \alpha_0)$ και
  - $\mathbb{P}(r_t|\omega_t, \alpha_t) = \mathbb{P}(r_t|\omega_t, \alpha_t, \dots, \omega_0, \alpha_0)$
- Δηλαδή η *επόμενη τιμή* εξαρτάται μόνο από την *τωρινή* και όχι τις παρελθοντικές
  - Ιδιότητα **«αμνησίας»**
- Μια στοχαστική διαδικασία ελέγχου διακριτού χρόνου που έχει τη μαρκοβιανή ιδιότητα ονομάζεται **μαρκοβιανή διαδικασία απόφασης** (*Markov Decision Process – MDP*)

Μάθηση RL ως MDP

- Μια *πλειάδα* $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ 5 στοιχείων, όπου
  1. $\mathcal{S}$ χώρος καταστάσεων
  2. $\mathcal{A}$ χώρος δράσεων
  3. $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow [0,1]$ η **συνάρτηση μετάβασης** (*transition function*)
     - Σύνολο υπό συνθήκη πιθανοτήτων μετάβασης μεταξύ των καταστάσεων
  4. $\mathcal{R}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow \mathcal{R}$ η **συνάρτηση ανταμοιβής** (*reward function*)
     - $\mathcal{R}$ συνεχής σε εύρος $[0, R_{max}]$, όπου $R_{max} \in \mathbb{R}^+$
  5. $\gamma \in [0,1)$
     - Συντελεστής ελάττωσης (discount factor) ανταμοιβής
- Το σύστημα είναι **πλήρως παρατηρήσιμο** $\Longrightarrow \omega_t = s_t$

Μάθηση πράκτορα RL

- Εύρεση **πολιτικής** $\pi \in \Pi$ *μετάβασης* από τη μια κατάσταση στην άλλη
- Μπορεί να πραγματοποιηθεί με *συνδυασμό* κάποιων (ή όλων) από τους παρακάτω τρόπους
  1. Με *απευθείας εκτίμηση* της $\pi(s)$ ή της $\pi(s, a)$
  2. Με τη χρήση **συνάρτησης αποτίμησης κατάστασης** (*state value function*) που εκτιμά πόσο επωφελής είναι μια *κατάσταση* ή ένα *ζεύγος κατάστασης-δράσης* για τον πράκτορα
     - Θα την δούμε σε επόμενη διαφάνεια
  3. Με την κατασκευή **μοντέλου** (*model*) για το περιβάλλον
- Συνδυασμός περιπτώσεων 1 και 2 ⇒ **RL χωρίς μοντέλο** (*model-free RL*)
- Περίπτωση 3 ⇒ **RL βασισμένη σε μοντέλο** (*model-based RL*)

Τεχνικές εκτίμησης πολιτικής

1. **Στάσιμες** (*stationary*) ή **Μη-στάσιμες** (*non-stationary*)
   - Στη δεύτερη περίπτωση, η *πιθανότητα μετάβαση*ς εξαρτάται από τη *χρονική στιγμή* t
2. **Ντετερμινιστικές** ή **Στοχαστικές**
   - Ντετερμινιστικές
     - $\pi(s): \mathcal{S} \to \mathcal{A}$
   - Στοχαστικές
     - $\pi(s, a): \mathcal{S} \times \mathcal{A} \longrightarrow [0,1]$
     - $\pi(s, a)$ η περίπτωση να επιλεγεί η **δράση** a όταν ο πράκτορας βρίσκεται στην **κατάσταση** s

Συνάρτηση αποτίμησης κατάστασης

- Πόσο ωφέλιμη είναι μια **κατάσταση** s για τον πράκτορα;
  - **Επιστρεφόμενη τιμή** (*return value*): $\mathrm{R} = \sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s$
  - *Άθροισμα* γεωμετρικής σειράς
    - Η ανταμοιβή *δεν τείνει* στο άπειρο, όσο περνάει ο χρόνος
    - Ο πράκτορας *προτιμά* τις *πιο «άμεσες»* ανταμοιβές
- **Αναμενόμενη** επιστρεφόμενη τιμή
  - $V_\pi(s) = \mathbb{E}[\mathrm{R}]$

Συνάρτηση Τιμής (Value function)

- Ποια από τις *διαθέσιμες πολιτικές* $\pi(s) \in \Pi$ **μεγιστοποιεί** *την αναμενόμενη τιμή επιστροφής* R σε μια κατάσταση s;
  - $V^{\pi}(s) = \mathbb{E}[\mathrm{R}|\pi] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \pi]$, με $V^{\pi}(s): \mathcal{S} \to \mathbb{R}$
- Η $V^{\pi}(s)$ καλείται **συνάρτηση V-value**
  - Από τον ορισμό της *μεγιστοποιείται* εκεί που $V^*(s) = \max_{\pi(s)\in\Pi} V^{\pi}(s)$
  - *Βέλτιστη πολιτική* π^* που **μεγιστοποιεί** V-value: $\pi^* = arg\max_{\pi} V^{\pi}(s)$

Συνάρτηση Q-value

- Παρότι η αποτίμηση καταστάσεων *αρκεί* για τους πράκτορες RL, σε *πολλές περιπτώσεις* η *αποτίμηση ζευγών κατάστασης-δράσης* μπορεί να είναι ιδιαίτερα βοηθητική
- Ποια από τις *διαθέσιμες πολιτικές* $\pi(s,\alpha) \in \Pi$ **μεγιστοποιεί** την αναμενόμενη τιμή επιστροφής R ζεύγους κατάστασης-δράση (s,α);
  - $Q^{\pi}(s,a) = \mathbb{E}[\mathrm{R}|\pi] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \alpha_0 = \alpha, \pi]$, με $Q^{\pi}(s,a): \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Η $Q^{\pi}(s,a)$ καλείται συνάρτηση Q-value
  - Από τον ορισμό της *μεγιστοποιείται* εκεί που $Q^*(s,\alpha) = \max_{\pi(s,\alpha)\in\Pi} Q^{\pi}(s,\alpha)$
  - *Βέλτιστη πολιτική* π^* που **μεγιστοποιεί** Q-value: $\pi^*(s) = arg\max_{a\in\mathcal{A}} Q^*(s,\alpha)$

Στρατηγικές προσδιορισμού βέλτιστης πολιτικής

1. **Brute force**
   - Δοκίμασε *δειγματοληπτικά* κάθε δυνατή πολιτική
   - *Επέλεξε* εκείνη με τη *μεγαλύτερη αναμενόμενη επιστροφή*
   - Προβληματική προσέγγιση σε μεγάλους χώρους καταστάσεων ή/και ανταμοιβών
2. **Επανάληψη τιμών** (*value iteration*)
3. **Επανάληψη πολιτικών** (*policy iteration*)

- Τις δύο τελευταίες θα τις δούμε πιο αναλυτικά στις επόμενες διαφάνειες

Επανάληψη τιμών

- Προτάθηκε το **1957** από τον **Bellman**
- Καλείται και **προς τα πίσω επαγωγή** (*backward induction*)
- Βέλτιστη τιμή $V^*(s)$ για βέλτιστη στρατηγική π^* αναλύεται σε
  - $V^*(s) = r_0 + \gamma \max_{s_0} r_1 + \gamma^2 \max_{s_1} r_2 + \cdots = \mathbb{E}_{s'}[r + \gamma V(s')]$
    1. *Ξεκινάω* από εκτίμηση V_0
    2. Υ*πολογίζω επαναληπτικά* εκτιμήσεις V_i για όλες τις καταστάσεις μέχρι η σειρά να *συγκλίνει* επαναληπτικά στην αναμενόμενη τιμή του δεξιότερου τμήματος της Σχέσης που καλείται *Εξίσωση Bellman*
  - Η **πολιτική** *ενσωματώνεται* μέσα στη V-value
- Με *αντίστοιχο* τρόπο προσδιορίζεται και η $Q^*(s, \alpha)$
- Η διαδικασία αυτή **εγγυημένα συγκλίνει** σε βέλτιστες τιμές

Επανάληψη πολιτικών

- Προτάθηκε το **1960** από τον **Howard**
- Η πολιτική δεν ενσωματώνεται στις τιμές των $V^{\pi}(s)$, $Q^{\pi}(s,a)$, *όπως προηγουμένως*
- *Μοιάζει*, ως προς την φιλοσοφία, με **αλγόριθμο expectation-maximization**
  1. *Εκτίμηση* π' για βέλτιστη πολιτική
  2. *Προσέγγιση βέλτιστων* τιμών V, Q για τη συγκεκριμένη εκτίμηση
  3. Ενημέρωση εκτίμησης π στη βάση των τιμών που υπολογίστηκαν στο προηγούμενο βήμα
  4. Σύγκλιση όταν π δεν μεταβάλλεται πλέον μεταξύ των επαναλήψεων
- Πιο αργή σύγκληση σε σύγκριση με *επανάληψη τιμών*

Q-learning

- RL χωρίς μοντέλο
- $\mathcal{R}$ συνάρτηση ανταμοιβής $\Longrightarrow$ **Πίνακας ανταμοιβής** (*reward table*) R
  - Γνωστός *εκ των προτέρων* και *σταθερός*
  - Στο διπλανό παράδειγμα με -1 συμβολίζονται τα *μη-δυνατά* ζεύγη κατάστασης-δράσης
- $Q(s,a) \Longrightarrow$ **Πίνακας Q** (Q-table)
  - Αρχικά κενός
  - *Ενσωματώνει «εμπειρία»* πράκτορα καθώς μαθαίνει

$$R = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} \multicolumn{6}{c}{\text{Action}} \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q-learning: Κανόνας μετάβασης

- $Q(s_t, \alpha_t) = R(s,a) + \gamma \max_{\alpha_{t+1}} Q(s_{t+1}, \alpha_{t+1})$
- Έστω ότι *αρχικά* βρίσκομαι στην *κατάσταση* 1
- Μπορώ να επιλέξω <u>είτε</u> τη δράση 3 με ανταμοιβή 0 <u>ή</u> τη δράση 5 με ανταμοιβή 100
  - Προτιμώ **άμεσες ανταμοιβές**.
  - *Επιλέγω* δράση 5 που με *οδηγεί* στην κατάσταση 3
- **Ενημέρωση** πίνακα Q
  - $Q(1,5) = R(1,5) + 0.8 * \max\{Q(3,1), Q(3,2), Q(3,5)\} = 100 + 0.8 * 0 = 100$

$$R = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} \multicolumn{6}{c}{\text{Action}} \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q-learning: Κανόνας μετάβασης (συνέχεια)

- $Q(s_t, \alpha_t) = R(s,a) + \gamma \max_{\alpha_{t+1}} Q(s_{t+1}, \alpha_{t+1})$
- Από το προηγούμενο βήμα βρέθηκα στην κατάσταση 3
  - Όλες οι δράσεις μου δίνουν ίδια ανταμοιβή
  - Μέσω επανάληψης τιμών είτε επανάληψης πολιτικών *επιλέγω* τη δράση 1 που με οδηγεί στην κατάσταση 1
- $Q(3,1) = R(3,1) + 0.8 * \max\{Q(1,3), Q(1,5)\} =$
  $0 + 0.8 * 100 = 80$

$$R = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} & & & \text{Action} & & \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q-learning: Αλγόριθμος μάθησης

- **Αλγόριθμος** μάθησης
  1. *Αρχικοποίηση* πίνακα $Q(s,a)$
  2. Για όλα τα επεισόδια (εποχές) εκπαίδευσης
     i. *Επέλεξε* αρχική κατάσταση s
     ii. *Επανέλαβε*, μέχρις ότου να φτάσεις σε τελική κατάσταση
        a. *Επέλεξε* δράση α (πχ μέσω επανάληψης τιμών ή πολιτικών)
        b. *Λάβε* ανταμοιβή r και εξέτασε τη νέα κατάσταση s'
        c. $\hat{Q}(s_t, \alpha_t) \longleftarrow (1-\eta)Q(s_t, \alpha_t) + \eta\left(R(s,a) + \gamma \max_{\alpha_{t+1}} Q(s_{t+1}, \alpha_{t+1}) - Q(s_t, \alpha_t)\right)$
- η: ρυθμός μάθησης
- *Κανόνας ανταμοιβής* $R(s,a) + \gamma \max_{\alpha_{t+1}} \hat{Q}(s_{t+1}, \alpha_{t+1})$
  - Μπορεί να θεωρηθεί ως *σύνολο* δειγμάτων *ζευγών κατάστασης-δράσης*
  - Μέσω του κανόνα της μάθησης θέλουμε το $\hat{Q}(s_t, \alpha_t)$ να συγκλίνει στη *μέση τιμή* του

Παραπομπές

- [1] Κ. Διαμαντάρας, Δ. Μπότσης, Μηχανική Μάθηση, Εκδόσεις Κλειδάριθμος, 2019.
- [2] Γ. Αλεξανδρίδης, Αυτοενισχυόμενη μάθηση, διαφάνειες.