ε.δε.μ²
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ & ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΕΡΓΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Δυναμικός Προγραμματισμός με Μεθόδους Monte Carlo:

1. Μάθηση Χρονικών Διαφορών (Temporal-Difference Learning)

2. Στοχαστικός Αλγόριθμος Q-Learning

καθ. Βασίλης Μάγκλαρης

maglaris@netmode.ntua.gr

www.netmode.ntua.gr

Video Conference μέσω Cisco Webex

Πέμπτη 7/5/2020

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Αλγόριθμος Policy Iteration (1/2) (*Επανάληψη*)

Ορισμός Q-factor

Έστω χρονοσταθερή πολιτική $\pi = \{\mu, \mu, \dots\}$ που οδηγεί σε αναμενόμενα ***costs-to-go*** $J^{\mu}(i), \forall i \in \mathcal{X}$ (καταστάσεις του ***περιβάλλοντος***) με αποφάσεις του ***agent*** $a = \mu(i) \in \mathcal{A}_i$

Για κάθε ζεύγος (i, a) στο υπό εξέταση βήμα και πολιτική για τα υπολειπόμενα βήματα $\pi = \{\mu, \mu, \dots\}$ ορίζω τους ***Q-factors*** $Q^{\mu}(i, a)$ σαν μέτρο κατάταξης εναλλακτικών άμεσων αποφάσεων $a \in \mathcal{A}_i$ του ***agent*** που θα οδηγούσαν το ***περιβάλλον*** σε καταστάσεις j με αναμενόμενα υπολειπόμενα ***costs-to-go*** $J^{\mu}(j), \forall j \in \mathcal{X}$

$$Q^{\mu}(i, a) \triangleq c(i, a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, J^{\mu}(j)$$

Μια πολιτική $\pi = \{\mu, \mu, \dots\}$ ικανοποιεί τις συνθήκες απληστίας (***greedy conditions***) σε σχέση με τα αναμενόμενα ***costs-to-go*** $J^{\mu}(j)$ στα υπολειπόμενα βήματα όταν σε κάθε βήμα και $\forall i \in \mathcal{X}$ ο ***agent*** επιλέγει $a = \mu(i)$ ώστε

$$Q^{\mu}(i, \mu(i)) = \min_{a \in \mathcal{A}_i} Q^{\mu}(i, a), \forall i \in \mathcal{X}$$

Μια πολιτική $\pi^* = \{\mu^*, \mu^*, \dots\}$ είναι βέλτιστη για όλα τα βήματα αν ικανοποιεί τις συνθήκες απληστίας (***greedy conditions***) του δυναμικού προγραμματισμού:

$$Q^{\mu^*}(i, \mu^*(i)) = \min_{a \in \mathcal{A}_i} Q^{\mu^*}(i, a)$$

Σημείωση: Όταν τα άμεσα αναμενόμενα κόστη $c(i, a)$ αντικαθίστανται από ***rewards*** $r(i, a)$, τα ***costs-to-go*** $J^{\mu}(i)$ αποκαλούνται ***Value Functions*** $V^{\mu}(i)$ και έχουμε κατ' αντιστοιχία:

$$Q^{\mu}(i, a) \triangleq r(i, a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, V^{\mu}(j) \text{ και } Q^{\mu^*}(i, \mu^*(i)) = \max_{a \in \mathcal{A}_i} Q^{\mu^*}(i, a)$$

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Αλγόριθμος Policy Iteration (2/2) (*Επανάληψη*)

Αρχιτεκτονική *Actor – Critic* (*Barto 1983*)

Επαναλήψεις $n = 0,1,2, \ldots$ από δύο βήματα μέχρι $\mu_{n+1}(i) = \mu_n(i),\ J^{\mu_{n+1}}(i) = J^{\mu_n}(i)\ , \forall i$

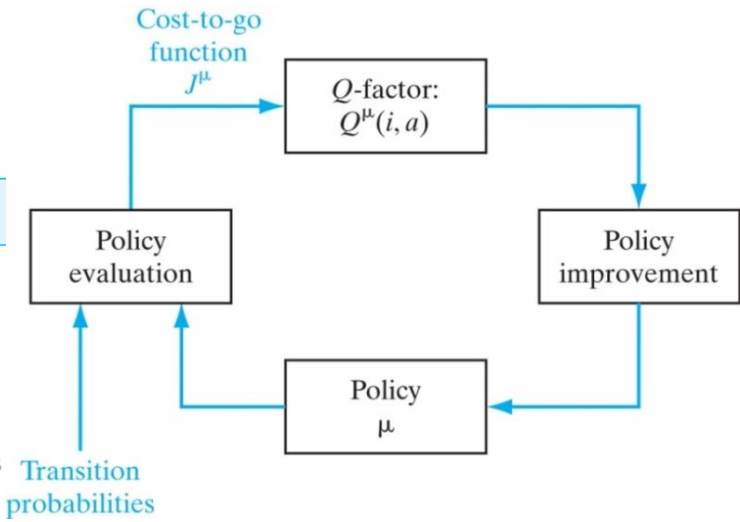
Βήμα 1. **Policy Evaluation** (ο ***critic*** αναλύει τις αποφάσεις του ***agent***):
Με βάση την παρούσα πολιτική $\pi_n = \{\mu_n, \mu_n, \ldots\}$ υπολογίζονται τα ***costs-to-go***
$J^{\mu_n}(i) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, J^{\mu_n}(j)$ για $i = 1,2, \ldots, N$
και οι ***Q-factors*** $Q^{\mu_n}(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, J^{\mu_n}(j)$ για $i = 1,2, \ldots, N$ και $a \in \mathcal{A}_i$

Βήμα 2. **Policy Improvement** (ο ***actor*** καθοδηγεί τις αποφάσεις του ***agent***):
Η πολιτική π_n βελτιώνεται σε π_{n+1} μέσω της $\mu_{n+1}(i) = \arg\min_{a \in \mathcal{A}_i} Q^{\mu_n}(i,a)$ για $i = 1,2, \ldots, N$

$\arg\min_x f(x)$: Η τιμή της x που οδηγεί την $f(x)$ σε ελάχιστο

TABLE 12.1 Summary of the Policy Iteration Algorithm

1. Start with an arbitrary initial policy μ_0.
2. For $n = 0, 1, 2, \ldots$, compute $J^{\mu_n}(i)$ and $Q^{\mu_n}(i, a)$ for all states $i \in \mathcal{X}$ and actions $a \in \mathcal{A}_i$.
3. For each state i, compute
$$\mu_{n+1}(i) = \arg\min_{a \in \mathcal{A}_i} Q^{\mu_n}(i, a)$$
4. Repeat steps 2 and 3 until μ_{n+1} is not an improvement on μ_n, at which point the algorithm terminates with μ_n as the desired policy.



Ο αλγόριθμος συγκλίνει σε βέλτιστη πολιτική σε πεπερασμένα βήματα n λόγω πεπερασμένου πλήθους καταστάσεων N και πεπερασμένων επιλογών αποφάσεων

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Value Iteration Algorithm (*Επανάληψη*)

Εκτίμηση των Συναρτήσεων Cost-to-Go μέσω Διαδοχικών Προσεγγίσεων $J_n(i) \to J_{n+1}(i)$

- Εκκίνηση με αυθαίρετες τιμές $J_0(i) \ \ \forall i$
- Επαναλήψεις $n \to n+1$ μέχρι ***ανεκτή σύγκλιση*** (θεωρητικά $n \to \infty$) μέσω σχέσεων ***backup***:

  $J_{n+1}(i) = \min_{a \in \mathcal{A}_i} \{ c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j) \}$ για $i = 1,2,\dots,N$ (από εξισώσεις ***Bellman***)
- Τελικός υπολογισμός των βέλτιστων ***Costs-to-Go***

  $$J^*(i) = \lim_{n \to \infty} J_n(i), \ Q^*(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^*(j)$$

  και προσδιορισμός της ***βέλτιστης πολιτικής*** $\mu^*(i) = \arg\min_{a \in \mathcal{A}_i} Q^*(i,a)$ για $i = 1,2,\dots,N$

TABLE 12.2 Summary of the Value Iteration Algorithm

1. Start with arbitrary initial value $J_0(i)$ for state $i = 1, 2, \dots, N$.
2. For $n = 0, 1, 2, \dots$, compute

$$J_{n+1}(i) = \min_{a \in \mathcal{A}_i} \left\{ c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j), \right\}, \qquad \begin{matrix} a \in \mathcal{A}_i \\ i = 1, 2, \dots, N \end{matrix}$$

Continue this computation until

$$|J_{n+1}(i) - J_n(i)| < \epsilon \qquad \text{for each state } i$$

where ϵ is a prescribed tolerance parameter. It is presumed that ϵ is sufficiently small for $J_n(i)$ to be close enough to the optimal cost-to-go function $J^*(i)$. We may then set

$$J_n(i) = J^*(i) \qquad \text{for all states } i$$

3. Compute the Q-factor

$$Q^*(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^*(j) \qquad \begin{matrix} \text{for } a \in \mathcal{A}_i \text{ and} \\ i = 1, 2, \dots, N \end{matrix}$$

Hence, determine the optimal policy as a greedy policy for $J^*(i)$:

$$\mu^*(i) = \arg\min_{a \in \mathcal{A}_i} Q^*(i,a)$$

Ο αλγόριθμος ***Value Iteration*** αν συγκλίνει σε ικανοποιητικό χρόνο, αποφεύγει υπολογισμούς ***Q-factors*** και ενδιάμεση ανανέωση πολιτικής σε κάθε βήμα όπως ο ***Policy Iteration***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Βελτιστοποίηση Δρομολόγησης (*Επανάληψη*)

Εύρεση Δρόμων Ελάχιστου Κόστους από Κόμβο A *σε* Κόμβο J μέσω του μονο-κατευθυντικού γράφου όπως στο σχήμα με κατεύθυνση γραμμών ***Α → Δ***

Ενδεικτικό κόστος γραμμών: $A \to B: 2,\ B \to A: \infty$

$B \to F: 4,\ F \to B: \infty$

Ενδεικτικό κόστος δρόμου: Δρόμος $\{A, B, F, I, J, Q\}$: $2 + 4 + 3 + 4 = 13$

Κατάσταση Περιβάλλοντος: Κόμβος σε παρούσα διερεύνηση $\{A, B, \dots, J\}$

Αποφάσεις Agent: Επόμενος κόμβος για διερεύνηση $\{up, down, staight\}$

Αναδρομικός Υπολογισμός *Q-Factors*:

$Q(H, down) = 3, \qquad Q(I, up) = 4$

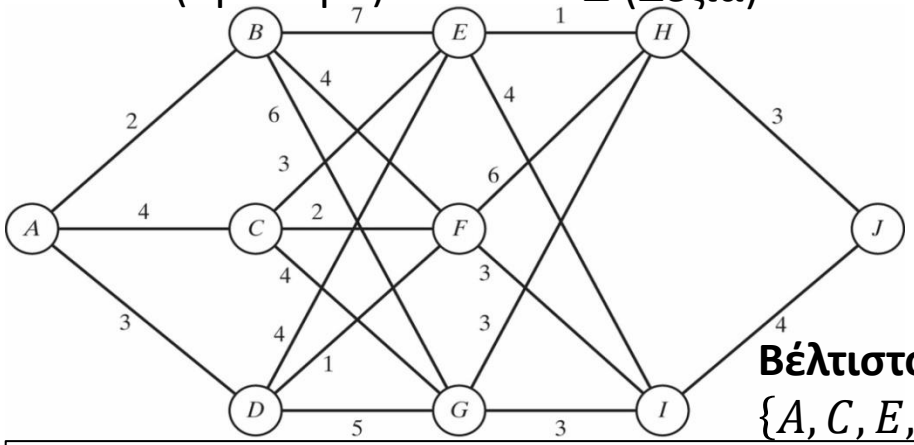
$Q(E, staight) = 1 + 3 = 4, \qquad Q(E, down) = 4 + 4 = 8$

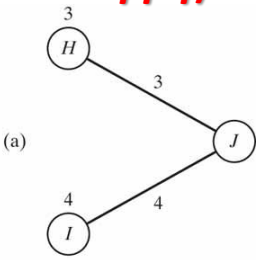
$Q(F, up) = 6 + 3 = 9, \qquad Q(F, down) = 3 + 4 = 7$

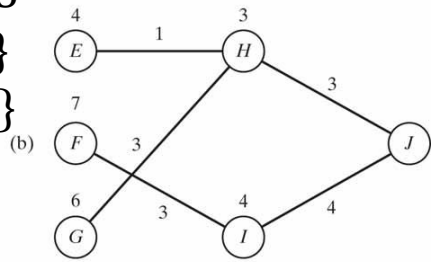
.....

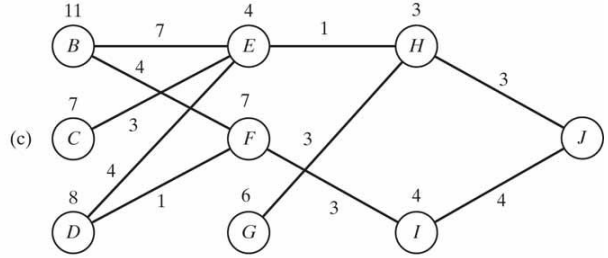
Κατεύθυνση Γραμμών

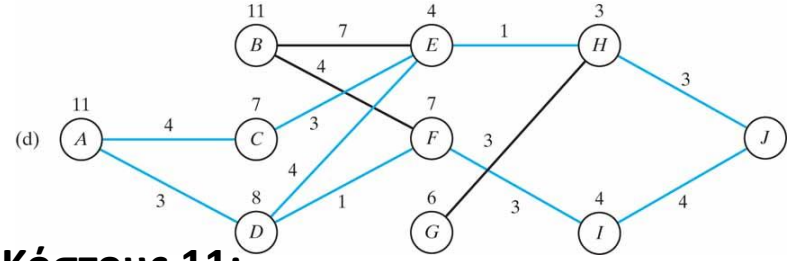
Α (Αριστερά) ⟶ ***Δ*** (Δεξιά)











Βέλτιστοι Δρόμοι Κόστους 11:

$\{A, C, E, H, J\}, \{A, D, E, H, J\}, \{A, D, F, I, J\}$

Αλγόριθμοι Δυναμικού Προγραμματισμού ***Bellman-Ford*** στηρίζουν την δρομολόγηση ***Border Gateway Protocols*** (***BGP***) ανάμεσα στα ~66,000 Αυτόνομα Συστήματα (***Autonomous Systems, AS***) στο ***Internet*** (~830,000 γνωστά δίκτυα)

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Απευθείας Προσεγγιστικές Μέθοδοι Δυναμικού Προγραμματισμού (1/2)

Οι δύο αλγόριθμοι Δυναμικού Προγραμματισμού (***Value Iteration*** & ***Policy Iteration***) προαπαιτούν γνώση των **πιθανοτήτων μεταβάσεων** $p_{ij}(a)$ και του ***άμεσα αναμενόμενου κόστους κατάστασης*** $c(i,a) = \sum_{j=1}^{N} p_{ij}(a) g(i,a,j)$ **εκτιμώμενου** με βάση τα **γνωστά** $g(i,\mu(i),j) = g(i,a,j)$ (***observed κόστη μετάβασης*** $i \to j$ με απόφαση a)

Οι απευθείας προσεγγιστικές μέθοδοι (***Direct Approximate Dynamic Programming Methods***) εκτιμούν τις πιθανότητες μετάβασης και τα αναμενόμενα κόστη μεταβάσεων - αποφάσεων μακροπρόθεσμων πολιτικών με προσομοιώσεις ***Monte Carlo***. Ενσωματώνονται στους δύο βασικούς αλγόριθμους Δυναμικού Προγραμματισμού με τις εξής παραλλαγές:

- **Value Iteration** → **Temporal-Difference *TD*(0) Learning**
- **Policy Iteration** → ***Q*-Learning**

Γενική Μεθοδολογία - Απαιτήσεις

- Οι προσομοιώσεις ***Monte Carlo*** δημιουργούν σενάρια πολλαπλών πιθανών τροχιών (***system trajectories***) της εξέλιξης του ***Markov Decision Process*** σε κάθε **επεισόδιο** (***episode***) από μια αρχική κατάσταση i_0 μέχρι κάποια τελική $i_T = TERMINAL$ (T είναι το βήμα n που η κατάσταση $i_n \to i_T$ και τερματίζεται το **επεισόδιο**). Η διαδικασία μάθησης συνήθως περιλαμβάνει πολλά ανεξάρτητα **επεισόδια** με διαφορετικές ***trajectories***
- Οι τιμές συναρτήσεων ***cost-to-go*** $J(i)$ ανανεώνονται σε κάθε προσομοίωση με προσθήκη του (**γνωστού**) ***άμεσου*** (***observed***) ***κόστους μετάβασης*** $g(i,j)$ σε επισκέψεις προσομοιωμένης τροχιάς μεταβάσεων από κατάσταση i προς κατάσταση j
- Οι μέθοδοι ***Monte Carlo*** απαιτούν γνώση της δομής του περιβάλλοντος από εμπειρία (όχι από πρότερη γνώση πιθανοτήτων), διαχειρίσιμο αριθμό παρατηρήσιμων (***observable***) καταστάσεων και σημαντικό αριθμό από ***trajectories*** για καλές εκτιμήσεις

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Απευθείας Προσεγγιστικές Μέθοδοι Δυναμικού Προγραμματισμού (2/2)

Σύνοψη Εννοιών Δυναμικού Προγραμματισμού

Ορισμός ***Άμεσου*** (***Observed***) ***Κόστους***: $g(i,a,j)$ για μετάβαση $i \to j$ με απόφαση a

Ορισμός ***Άμεσου Αναμενόμενου Κόστους***: $c(i,a) \triangleq \sum_{j=1}^{N} p_{ij} g(i,a,j)$ για $\forall i$ και $a \in \mathcal{A}_i$

Ορισμός ***Cost-to-Go*** : $J^{\mu}(i) = c(i,\mu(i)) + \gamma \sum_{j=1}^{N} p_{ij}(\mu(i)) J^{\mu}(j)$ για $\forall i$ και πολιτική $\mu(i)$

Βέλτιστα ***Cost-to-Go*** (***Bellman***): $J^{*}(i) = \min_{a \in \mathcal{A}_i} \left(c(i,a) + \gamma \sum_{j=1}^{N} p_{ij} J^{*}(j) \right), i = 1,2,\dots,N$

Ορισμός ***Q-Factors***: $Q^{\mu}(i,a) \triangleq c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^{\mu}(j)$ για $\forall i$ και $a \in \mathcal{A}_i$

Ορισμός ***Βέλτιστων Q-Factors***: $Q^{*}(i,a) = \sum_{j=1}^{N} p_{ij}(a) \left(g(i,a,j) + \gamma \min_{b \in \mathcal{A}_j} Q^{*}(j,b) \right)$

Ορισμοί *on-policy*, *off-policy*

- Η ***on-policy*** σε κάθε βήμα εκτιμά με προσομοιώσεις ***Monte Carlo*** το κόστος $J^{\mu}(i)$ των καταστάσεων i μιας τροχιάς (***trajectory***) όταν ακολουθείται η υπό αξιολόγηση **συνολική** πολιτική μ. Με επαναλήψεις που περιλαμβάνουν διορθωτικές αποφάσεις $i \to \alpha$ οδηγούνται τα $J^{\mu}(i)$ σε διαδοχικές μειώσεις (**Value Iteration → *TD*(0)-Learning**)
- Η ***off-policy*** συγκρίνει εναλλακτικές αποφάσεις σε καταστάσεις του περιβάλλοντος i μιας τροχιάς (***trajectory***) και σε κάθε βήμα **επιλέγει** με απληστία αποφάσεις a με το ελάχιστο $Q(i,a)$ στην παρούσα κατάσταση i. Τα ***Cost-to-Go*** $J^{\mu}(j)$ μιας προσωρινής πολιτικής μ εκτιμώνται από προσομοιώσεις ***Monte Carlo*** των ***trajectories*** χωρίς να συμπεριλαμβάνουν βελτιώσεις $i \to \alpha$ που ίσως προκύψουν από τα ***Q-Factors*** (**Policy Iteration → *Q*-Learning**)

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *TD*(0) Learning (1/2)

Value Iteration → Temporal-Difference *TD*(0) Learning

Εξισώσεις ***Bellman*** υπολογισμού ***costs-to-go*** από i_n στο βήμα $n < T$ με τελική κατάσταση i_T:

$$J^{\mu}(i_n) = \mathrm{E}[g(i_n, i_{n+1}) + \gamma J^{\mu}(i_{n+1})] = \mathrm{E}\left[\sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1})\right], n = 0,1, \dots, T-1$$

Με επανειλημμένες προσομοιώσεις ***Monte Carlo*** δημιουργούμε ***trajectories*** του συστήματος και μαθαίνουμε τα $J^{\mu}(i_n)$ μέσω ***Robbins-Monro Successive Approximations*** που διορθώνουν εκτιμήσεις τιμών τους (***updates***) κατά την επίσκεψη της κατάστασης i_n με συντελεστή μάθησης (***learning rate***) η_n:

$$J^{\mu}(i_n) \coloneqq J^{\mu}(i_n) + \eta_n[g(i_n, i_{n+1}) + \gamma J^{\mu}(i_{n+1}) - J^{\mu}(i_n)] = J^{\mu}(i_n) + \eta_n d_n$$

Το σφάλμα $d_n \triangleq g(i_n, i_{n+1}) + \gamma J^{\mu}(i_{n+1}) - J^{\mu}(i_n), n = 0,1, \dots, T-1$ ονομάζεται χρονική διαφορά (***Temporal Difference***, ***TD***) στο βήμα n και οδηγεί τα $J^{\mu}(i_n)$ προς τη ***σύγκλιση***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *TD*(0) Learning (2/2)

Value Iteration → Temporal-Difference *TD*(0) Learning

Εναλλακτικός αλγόριθμος ***update*** προκύπτει από την μακρόχρονη επαναληπτική σχέση:

$$J^{\mu}(i_n) := J^{\mu}(i_n) + \eta_n \left(\sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1}) - J^{\mu}(i_n) \right) = J^{\mu}(i_n) + \eta_n \sum_{k=0}^{T-n-1} \gamma^k d_{n+k}$$

Τα ***costs-to-go*** εκτιμώνται σαν μέσοι όροι σε μεγάλο αριθμό M επαναλήψεων προσομοιώσεων με πολλαπλές επισκέψεις καταστάσεων i_n στο βήμα n κάποιου ***trajectory*** :

$$J^{\mu}(i_n) = \mathrm{E}\left[\sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1}) \right] \cong \frac{1}{M} \sum_{M} c(i_n)$$

όπου

$c(i_n) \triangleq \sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1})$ και $J^{\mu}(i_n) = \mathrm{E}[c(i_n)]$

$$J^{\mu}(i_n) := J^{\mu}(i_n) + \eta_n (c(i_n) - J^{\mu}(i_n))$$

με αρχικές συνθήκες $J^{\mu}(i_n) = 0$, και ***learning rate*** $\eta_n = {}^{1}/_{n}$, $n = 1,2, \dots, T$

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *Q*-Learning (1/2)

Policy Iteration → *Q*-Learning

- Προσδιορισμός πολιτικής βέλτιστης συμπεριφοράς (***off-policy behavior generation***) μέσω δημιουργίας πολλαπλών ***trajectories*** (τροχιών) για δυνατά σενάρια αποφάσεων: ***Q*-Learning**

- Ορίζουμε $s_n \triangleq (i_n, a_n, j_n, g_n)$ στο βήμα n μιας ***trajectory*** όταν η κατάσταση του περιβάλλοντος οδηγείται σε μετάβαση $i_n \rightarrow i_{n+1} = j_n$ με απόφαση του agent a_n και observed κόστος μετάβασης $g_n = g(i_n, a_n, j_n)$

- Με βάση την καταγραφή των s_n σε προσομοιωμένες ***trajectories*** ο αλγόριθμος ***Q*-Learning** οδηγεί το σύστημα στη μάθηση βέλτιστης πολιτικής κατά προσέγγιση του ***policy iteration***

- **Προϋπόθεση**: Η i_n που προκύπτει σε μια ***trajectory*** πρέπει να είναι ***fully observable***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *Q*-Learning (2/2)

Policy Iteration → *Q*-Learning

Αλγόριθμος Υπολογισμού $Q^*(i,a)$ με Successive Approximations (*Robins-Monro*)

$$Q(i,a) := (1-\eta)Q(i,a) + \eta \sum_{j=1}^{N} p_{ij}(a)\left[g(i,a,j) + \gamma \min_{b\in\mathcal{A}_j} Q(j,b)\right] \text{ για } \forall(i,a)$$

Από τα $Q^*(i,a)$ προσδιορίζεται ο πίνακας βέλτιστης πολιτικής π με αντιστοίχηση

$$\mu^*(i) = \arg\min_{a\in\mathcal{A}_i} Q^*(i,a) \text{ για } i = 1,2,\dots,N$$

Στοχαστική Παραλλαγή

Έστω ότι η προσομοίωση ***Monte Carlo*** ορίζει τροχιά (***trajectory***) από αρχική κατάσταση i_0 μέχρι την i_n στο παρόν βήμα n με άπληστες επιλογές (i_n, a_n) που προσδιορίζουν τα ***costs-to-go*** $J_n(j_n)$ για $i_n \to j_n$. Ο επαναληπτικός αλγόριθμος ανανεώνει τους ***Q-factors*** από $Q_n(i,a)$ σε $Q_{n+1}(i,a)$ για προσομοίωση **επεισοδίου** $n = 0,1,\dots,T$ ως εξής:

- $Q_{n+1}(i,a) = (1-\eta_n)Q_n(i,a) + \eta_n[g(i,a,j) + \gamma J_n(j)]$ για $(i,a) = (i_n, a_n)$

  όπου $J_n(j) = \min_{b\in\mathcal{A}_j} Q_n(j,b)$ και j επόμενη κατάσταση της $i = i_n$ με τα παρόντα ***Q-factors***
- $Q_{n+1}(i,a) = Q_n(i,a)$ για όλα τα υπόλοιπα ζεύγη $(i,a) \neq (i_n, a_n)$
- Με την πρόοδο των επαναλήψεων $Q_n(i,a) \to Q^*(i,a)$, τις βέλτιστες τιμές των ***Q-factors***
- Η ***learning parameter*** η_n είναι φθίνουσα ως προς n, π.χ. $\eta_n = \alpha/(\beta+n)$ με α, β θετικά

Επειδή μια τροχιά με ***greedy*** αποφάσεις (***exploitation***) μπορεί να αγνοήσει άλλες επιλογές λόγω εκκίνησης από μια κατάσταση, μπορεί να απαιτείται προσομοίωση πολλαπλών **επεισοδίων** για επισκέψεις σε ευρύ φάσμα καταστάσεων (***exploration***). Το εύρος της αναζήτησης ενισχύεται με απόφαση ***greedy*** με πιθανότητα $(\mathbf{1}-\boldsymbol{\epsilon})$ ή άλλης με πιθανότητα $\boldsymbol{\epsilon}$