Értékelőfüggvény közelítése a megerősítéses tanulásban

Készítette: Kaczúr Flórián
Témavezető: Csáji Balázs Csanád

1. Bevezetés

A megerősítéses tanulás a gépi tanulás azon ágazata, melynek során a döntéshozó (más néven ágens) egy adott környezetben döntéseket hoz, a környezet ezekre visszajelzést (jutalmat vagy büntetést) ad és átlép egy következő állapotba. Az ágens ezek alapján kísérletezéssel próbálja kialakítani az optimális politikáját. Az optimális politika megtalálása helyett már egy rögzített politika kiértékelése sem könnyű feladat. Erre egy jól ismert módszercsalád a projektált Bellman-egyenletek: a félév során ennek különféle változataival ismerkedtem meg. Ezt a második fejezetben vezetem be, előtte azonban definiálom a megerősítéses gépi tanuláshoz szükséges matematikai keretrendszert és felvázolom a főbb nehézségeket.

A fent körülírt problémakör formálisan az alábbiak összességével írható le:

- X állapotok halmaza
- A akciók halmaza
- $\mathcal{A} : X \to \mathcal{P}(A)$ egy adott állapotban lehetséges akciók
- $p : X \times A \to \Delta(X)$: egy $x \in X$-ből az $a \in A$ akciót választva $p_{xy}(a)$ valószűséggel jutunk el az y állapotba
- $g : X \times A \to \Delta(\mathbb{R})$: egy $x \in X$ állapotban $a \in A$ döntést hozva, $g(a, x)$ büntetést/jutalmat kapunk.

A fentiekben $\mathcal{P}(Y)$ az Y halmaz hatványhalmazát, $\Delta(Y)$ pedig az Y-on való valószínűségeloszlások halmazát jelöli. Ezek összességét *Markov-döntési folyamat*nak nevezzük, mely a matematikai keretrendszerét szolgálja a megerősítéses gépi tanulásnak.

Politika alatt egy

$$\mu : X \to \Delta(A)$$

függvényt értünk.

Minden μ politika indukál egy P *átmenetmátrix*ot, melynek az (i, j)-edik eleme

$$p_{ij} = \sum_{a \in \mathcal{A}(i)} p_{ij}(a) \cdot \mu(i, a), \quad i, j = 1, \ldots, n,$$

ahol $p_{ij}(a)$ annak a valószínűsége, hogy az i állapotban az a döntést hozva a j állapotba jutunk, $\mu(i, a)$ pedig annak a valószínűsége, hogy a μ politika az i állapotban az a akciót választja.

Jelölje ξ a P átmenetmátrix által meghatározott folyamat stacionárius eloszlását, vagyis amire $P\xi = \xi$ fennáll.

A továbbiakban csak véges állapotterekkel foglalkozunk, azaz feltesszük, hogy $X = \{1, 2, \ldots, n\}$. Feltesszük azt is, hogy az akciók halmaza is véges.

A fenti keretrendszerben több feladattípust szokás vizsgálni. Mi csak azzal az esettel foglalkozunk, amiben a folyamat sosem terminál, a költségek pedig diszkontálva vannak egy $0 < \alpha < 1$ faktorral. Belátható, hogy a többi feladattípus felírható ennek speciális eseteként.

Ahhoz, hogy egy μ politikának hatékonyságát mérni tudjuk, bevezetünk egy

$$J^\mu : X \to \mathbb{R}$$

*értékelőfüggvény*t, amit a feladattípusnak megfelelően

$$J^\mu(i) \doteq \limsup_{N\to\infty} \mathbb{E}\Big\{ \sum_{k=0}^{N-1} \alpha^k g\big(i_k, i_{k+1}\big) | i_0 = i \Big\}, \quad i = 1, \dots, n$$

definiálunk.

A továbbiakban legyen μ egy tetszőleges, rögzített politika. Célunk, hogy μ értékelőfüggvényére, J^μ-re közelítést adjunk. Az optimális politika megtalálására szintén számos technika ismert, melyek az itt bemutatott módszerek továbbfejlesztései, ezzel azonban a félév során nem foglalkoztam.

A témakör egy központi fogalma a *Bellman-operátor* (amit a továbbiakban T-vel jelölök), mely a $J : X \to \mathbb{R}$ függvényeken hat az alábbi módon:

$$(TJ)(i) \doteq \sum_{j=1}^{n} p_{ij} \cdot \big(g(i,j) + \alpha J(j)\big), \; i = 1, \dots, n.$$

A későbbiekben ezt $TJ = g + \alpha PJ$-vel rövidítjük, ahol $g(i) = \sum_{j=1}^{n} p_{ij} g(i,j)$ minden $i = 1, \dots, n$-re.

1.1. Állítás. *A T Bellman-operátor kontrakció, melynek egyetlen fixpontja J_μ.*

1.2. Következmény. *$T^k J_0 \to J_\mu$ tetszőleges kezdeti J_0 esetén.*

Egy adott politika értékelőfüggvényének kiszámítására számos módszer létezik, melyek a Markov-döntési folyamatok elméletébe tartoznak. A fő nehézséget azonban az okozza, hogy az átmenetvalószínűségeket (azaz P-t), illetve a költségeket (azaz g-t) a gyakorlatban nem ismerjük.

Számos olyan klasszikus megerősítéses tanulásos módszer ismert az értékelőfüggvény kiszámítására, melyek az átmenetvalószínűségek és a költségek explicit ismeretét nem tételezik fel. Ezek azonban nagyon számításigényesek tudnak lenni, hiszen n, vagyis az állapottér mérete a gyakorlatban nagy szokott lenni.

A továbbiakban egy olyan módszer-családdal foglalkozunk, amiben ezt az értékelőfüggvényt közelítjük, de egy n-nél kisebb dimenziós térben.

2. Projektált Bellman-módszerek

A μ politika értékelőfüggvényének, J_μ-nek közelítését egy rögzített m-dimenziós lineáris altérben keressük, ahol $m < n$. Legyenek $\Phi_1, \dots, \Phi_m \in \mathbb{R}^n$ rögzített vektorok, az általuk feszített altér pedig legyen

$$S \doteq span\{\Phi_1, \dots, \Phi_m\} = \{\Phi r | r \in \mathbb{R}^m\}.$$

Legyen továbbá

$$\Phi \doteq \begin{pmatrix} \Phi_1(1) & \dots & \Phi_m(1) \\ \vdots & \ddots & \vdots \\ \Phi_1(n) & \dots & \Phi_m(n) \end{pmatrix} = \begin{pmatrix} \Phi(1)^T \\ \vdots \\ \Phi(n)^T \end{pmatrix} .$$

Ekkor Φ egy $n \times m$-es (hosszú) mátrix, aminek az i-edik sora ($1 \le i \le n$) az i állapothoz tartozó ún. 'feature vector' lesz.

A továbbiakban végig két feltevéssel fogunk élni:

(a) A Φ mátrix teljes rangú, azaz $rang(\Phi) = m$.

(b) A μ politika által indukált Markov-lánc irreducibilis.

Adott $J, \zeta \in \mathbb{R}^n$-re, legyen $\|J\|_\zeta \doteq \sqrt{\sum_{i=1}^{n} \zeta_i \cdot (J(i))^2}$.

Legyen $\Pi J \doteq \arg\min_{\hat{J} \in S} \|J - \hat{J}\|_\xi^2$, vagyis Π a továbbiakban az S-re való merőleges vetítés mátrixát jelöli a $\|.\|_\xi$ norma szerint.

2.1. Állítás. *A T és a ΠT operátorok kontrakciók a $\|.\|_\xi$ normára.*

A Banach fixpont-tétel következtében igaz az alábbi:

2.2. Következmény. *A $\Phi r = \Pi T(\Phi r)$ egyenletnek létezik egyetlen fixpontja.*

A továbbiakban ezt a Φr-et jelöljük $\hat{J}_\mu$-vel: ez lesz közelítésünk J_μ-re. Mivel a Φ mátrixról feltettük a teljes rangúságot, ezért egyetlen olyan r létezik, ami kielégíti a fenti fixpont-egyenletet. A továbbiakban ezt jelöljük r^*-al, azaz $\hat{J}_\mu = \Phi r^*$. A célunk az r^* kiszámítása lesz, melyből egy mátrixszorzás után megkaphatjuk az értékelőfüggvény közelítését.

A fentiek következményeként a

$$\Phi r_{k+1} = \Pi T(\Phi r_k) \tag{1}$$

iteráció az optimális r^*-hoz konvergál.

A közelítés és a politika valódi értékelőfüggvényének eltéréséről az alábbi állítás áll fenn.

2.3. Állítás. *Legyen Φr^* a ΠT fixpontja, J_μ pedig a μ politika értékelőfüggvénye. Ekkor*

$$\|J_\mu - \Phi r^*\|_\xi \leq \left(\frac{1}{\sqrt{1-\alpha^2}}\right) \|J_\mu - \Pi J_\mu\|_\xi.$$

Most pedig rátérünk az r^* kiszámítására.

Mivel $\Phi r^* = \Pi T \Phi r^*$, így r^* kielégíti a

$$\Phi^T diag(\xi)\big(\Phi r^* - (g + \alpha P \Phi r^*)\big) = 0 \tag{2}$$

egyenletet, ahol $diag(\xi)$ azt a diagonális mátrixot jelenti, amiben a főátlóbeli elemek $\xi_1, \ldots, \xi_n$.

A fenti (2)-es összefüggés átfogalmazható egy $Ar^* = b$ lineáris egyenletrendszerré, ahol

$$A = \Phi^T \cdot diag(\xi) \cdot (I - \alpha P)\Phi = \sum_{i=1}^{n} \xi_i \Phi(i) \Big(\Phi(i) - \alpha p_{ij} \Phi(j)\Big)^T,$$

és

$$b = \Phi^T \cdot diag(\xi) \cdot g = \sum_{i=1}^{n} \xi_i \Phi(i) \sum_{j=1}^{n} p_{ij} g(i,j).$$

Az (1)-es egyenletben leírt iterációt ekvivalensen egy minimalizálási feladatként is megfogalmazhatjuk:

$$r_{k+1} = \arg\min_{r \in \mathbb{R}^m} \|\Phi r - (g_\mu + \alpha P r_k)\|_\xi^2.$$

A gradienset 0-ra állítva az alábbi iterációs egyenletet kapjuk:

$$r_{k+1} = r_k - (\Phi^T \cdot diag(\xi) \cdot \Phi)^{-1}(A r_k - b). \tag{3}$$

A gond az, hogy A-t és b-t a P átmenetmátrix és a g költségvektor ismeretének hiányában nem tudjuk kiszámolni. Ezeket empirikusan fogjuk közelíteni, melyhez egy trajektóriát futtatunk a μ politikát követve, aminek a k-adik körében lesz egy A_k mátrixunk és egy b_k vektorunk az alábbi módon definiálva:

$$A_k = \frac{1}{k+1} \sum_{t=0}^{k} \Phi(i_t)(\Phi(i_t) - \alpha \Phi(i_{t+1}))^T, \tag{4}$$

$$b_k = \frac{1}{k+1}\sum_{t=0}^{k}\Phi(i_t)g(i_t, i_{t+1}). \tag{5}$$

Ezekre a nagy számok törvényéből következően fennáll, hogy $A_k \to A$ és $b_k \to b$, ha $k \to \infty$.

Könnyen ellenőrízhető, hogy a fenti A_k mátrixot és b_k vektort az alábbi módon tudjuk rekurzívan frissíteni:

$$A_k = \Big(1 - \frac{1}{k+1}\Big)A_{k-1} + \frac{1}{k+1}\Phi(i_k)\big(\Phi(i_k) - \alpha\Phi(i_{k+1})\big),$$
$$b_k = \Big(1 - \frac{1}{k+1}\Big)b_{k-1} + \frac{1}{k+1}\Phi(i_k)g(i_k, i_{k+1}).$$

Az r^* approximálására három fő módszer létezik, az első az úgynevezett *least squares temporal differences* (a továbbiakban **LSTD**), melyben a k-adik iterációban kiszámított A_k és b_k segítségével az $A_k r = b_k$ lineáris egyenletrendszert oldjuk meg r-re, azaz a közelítő megoldásunk:

$$\hat{r}_k := A_k^{-1} b_k. \tag{6}$$

Ennek hátulütője az, hogy garantálni kell, hogy $\hat{r}_k$ közel legyen r^*-hoz, hogy kis hibájú legyen a becslés. Egy közel szinguláris A esetén nagyon nagy k-ig kell elmennünk, hogy pontos közelítést kapjunk. Ezt az alábbi módon tudjuk orvosolni: legyen $\overline{r}$ az r^* *a priori* becslése. Az $\hat{r}_k$-t az alábbinak választjuk:

$$\hat{r}_k = \min_r\{(b_k - A_k r)\Sigma^{-1}(b_k - A_k r) + \beta\|r - \overline{r}\|^2\},$$

ahol Σ pozitív definit szimmetrikus mátrix, β pozitív szám, ami azt szabályozza, hogy mennyire legyünk közel az *a priori* becsléshez.

A gradienset nullára állítva kiszámítható az $\hat{r}_k$ explicit alakja:

$$\hat{r}_k = (A_k^T\Sigma^{-1}A_k + \beta I)^{-1}(A_k^T\Sigma^{-1}b_k + \beta\overline{r}).$$

Ekkor a k-adik körben kapott becslés és az optimális r^* távolságára az alábbi hibabecslés adható:

2.4. Lemma. $\|\hat{r}_k - r^*\| \le \max_{i=1,\dots,m}\left(\frac{\lambda_i}{\lambda_i^2+\beta}\right)\|c_k\| + \max_{i=1,\dots,m}\left(\frac{\beta}{\lambda_i^2+\beta}\right)\|\overline{r} - r^*\|$,
ahol $c_k = \Sigma^{-1/2}(b_k - A_k r^*)$ *és* $\lambda_1, \dots, \lambda_m$ *a* $\Sigma^{-1/2}A_k$ *szinguláris értékei.*

Ha $\Sigma = I$ és $\beta = 0$ (vagyis nincs regularizáló tag), akkor az $\hat{r}_k$-ra kapott explicit egyenlet éppen a (6)-ban definiált alakot ölti. Ekkor a lemma által adott becslés az alábbira redukálódik:

$$\|\hat{r}_k - r^*\| \le \max_{i=1,\dots,m}\left(\frac{1}{\lambda_i}\right)\|c_k\|.$$

Vagyis ebből az látszik, hogy regularizálás nélkül, ha az A_k mátrix közel szinguláris, akkor nagyon nagy k-ig kell elmenni, hogy kis hibával közelítsük r^*-ot.

Ezt regularizálással ugyan kiküszöbölhetjük, de ekkor $\hat{r}_k$ az *a priori* $\overline{r}$-hez közel lesz. Ezt tovább javíthatjuk úgy, hogy a $k+1$. szimulációban az *a priori* becslést éppen az $\hat{r}_k$-nak választjuk, mellyel az alábbi iterációs egyenletet kapjuk:

$$\hat{r}_{k+1} = (A_k^T\Sigma^{-1}A_k + \beta I)^{-1}(A_k^T\Sigma^{-1}b_k + \beta\hat{r}_k). \tag{7}$$

Ez azonban – mint látni fogjuk – a következő módszernek egy speciális esetévé redukálódik.

A második közismert módszer az ún. *least squares policy evaluation* (a továbbiakban **LSPE**), ami a (3)-as iterációt szimulálja. Legyen γ egy pozitív konstans. Ekkor belátható, hogy az

$$r_{k+1} = r_k - \gamma G(Ar_k - b) \tag{8}$$

iteráció is az r^*-hoz konvergál, ha G egy $m \times m$-es szimmetrikus pozitív szemidefinit mátrix. Ennek speciális esete a $G = \left(\Phi^T diag(\xi)\Phi\right)^{-1}$ választás, mely éppen a (3) alakot ölti.

A (8)-as iteráció empirikus közelítése a

$$r_{k+1} = r_k - \gamma G_k(A_k r_k - b_k)$$

iteráció, ahol γ pozitív, G_k egy $m \times m$-es szimmetrikus pozitív szemidefinit mátrix, A_k és b_k pedig a (4)-(5)-ben definiáltak. Egy magától értetődő választás G_k-ra a $\left(\Phi^T diag(\xi)\Phi\right)^{-1}$ empirikus közelítése:

$$G_k = \left(\frac{1}{k+1}\sum_{t=0}^{k}\Phi(i_t)\Phi^T(i_t)\right)^{-1}.$$

2.5. Megjegyzés. *A $\gamma = 1$, $G_k = (A_k^T\Sigma A_k + \beta I)^{-1}A_k^T\Sigma^{-1}$ választással éppen a (7)-es egyenletet kapjuk vissza.*

Egy másik – szintén iterációs – módszer az úgynevezett **TD(0)**, amit az alábbi iterációs egyenlet ír le:

$$r_{k+1} = r_k - \gamma_k \Phi(i_k) q_{k,k},$$

ahol γ_k egy 0-hoz monoton csökkenő módon tartó sorozat, $q_{k,k}$ pedig az ún. *temporal difference* a $q_{k,k} \doteq \Phi(i_k)^T \cdot r_k - \alpha\Phi(i_{k+1})^T \cdot r_k - g(i_k, i_{k+1})$ definícióval.

A TD(0) módszer előnye az LSPE-vel szemben, hogy egyetlen iteráció sokkal kevesebb számítást igényel, hátránya viszont, hogy sokkal lassabban konvergál.

2.1. Többlépéses Bellman-egyenlet

Egy általánosabb megközelítés, ha nem a T Bellman-operátort, hanem annak egy kiterjesztését vesszük:

$$T^{(\lambda)} \doteq (1-\lambda)\sum_{\ell=0}^{\infty}\lambda^\ell T^{\ell+1}, \tag{9}$$

ahol $\lambda \in [0, 1)$.

2.6. Megjegyzés. *$T^{(0)} = T$, vagyis a $\lambda = 0$ választással a fenti esetet kapjuk vissza.*

Legyen $\alpha_\lambda = \dfrac{\alpha(1-\lambda)}{1-\alpha\lambda}$.

2.7. Állítás. *A fent definiált $T^{(\lambda)}$ és $\Pi T^{(\lambda)}$ kontrakciók az α_λ modulussal a $\|.\|_\xi$ normára nézve.*

Az előző állítás (és Φ teljes rangúsága) következtében létezik egyetlen r_λ^*, amire

$$\Pi T^{(\lambda)}\Phi r_\lambda^* = \Phi r_\lambda^*.$$

Ebben az esetben a J^μ becslése Φr_λ^* lesz, melyek eltérésére az alábbi hibabecslés áll fenn.

2.8. Állítás. $\|J_\mu - \Phi r_\lambda^*\|_\xi \leq \left(\dfrac{1}{\sqrt{1-\alpha_\lambda^2}}\right)\|J_\mu - \Pi J_\mu\|_\xi$.

2.9. Megjegyzés. *Minél közelebb van λ az 1-hez, annál jobb becslést kapunk.*

A $T^{(\lambda)}$ operátor hatása egy J értékelő függvényre a $\lambda = 0$ esethez hasonló formában felírható:

$$T^{(\lambda)}J = g^{(\lambda)} + \alpha P^{(\lambda)}J,$$

ahol

$$P^{(\lambda)} = (1-\lambda)\sum_{\ell=0}^{\infty}\alpha^{\ell}\lambda^{\ell}P^{\ell+1}, \qquad g^{(\lambda)} = \sum_{\ell=0}^{\infty}\alpha^{\ell}\lambda^{\ell}P^{\ell}g = (I-\alpha\lambda P)^{-1}g.$$

Továbbá az r_λ^* ez esetben is kifejezhető lineáris egyenletrendszer megoldásaként:

$$A^{(\lambda)}r_\lambda^* = b^{(\lambda)},$$

ahol

$$A^{(\lambda)} = \Phi^T diag(\xi)(I-\alpha P^{(\lambda)})\Phi,$$
$$b^{(\lambda)} = \Phi^T diag(\xi)g^{(\lambda)}.$$

Az $A^{(\lambda)}$-t és $b_k^{(\lambda)}$-át az alábbi módon közelíthetjük:

$$A_k^{(\lambda)} = \frac{1}{k+1}\sum_{t=0}^{k}\Phi(i_t)\cdot\sum_{h=t}^{k}\alpha^{h-t}\lambda^{h-t}\big(\Phi(i_h)-\alpha\Phi(i_{h+1})\big)^T,$$
$$b_k^{(\lambda)} = \frac{1}{k+1}\sum_{t=0}^{k}\Phi(i_t)\sum_{h=t}^{k}\alpha^{h-t}\lambda^{h-t}g(i_h,i_{h+1}).$$

A nagy számok törvényéből következik, hogy $A_k^{(\lambda)} \to A^{(\lambda)}$ és $b_k^{(\lambda)} \to b^{(\lambda)}$, ha $k\to\infty$.

A fenti három módszernek itt is bevezethetjük a megfelelőjét.

LSTD(λ):

$$\hat{r}_k = \big(A_k^{(\lambda)}\big)^{-1}b_k^{(\lambda)},$$

LSPE(λ):

$$r_{k+1} = r_k - \gamma G_k\big(A_k^{(\lambda)}r_k - b_k^{(\lambda)}\big),$$

ahol G_k szimmetrikus, pozitív szemidefinit mátrix minden k-ra.

TD(λ):

$$r_{k+1} = r_k - \gamma_k z_k q_{k,k},$$

ahol $z_k = \sum_{h=0}^{k}(\alpha\lambda)^{k-h}\Phi(i_h)$.

2.2. Felfedezést javító módszerek

A fenti módszerek mind úgy adtak közelítést, hogy a μ politikát követve generáltunk egy nagyon hosszú trajektóriát és az ezalatt gyűjtött visszajelzések alapján frissítettük/alkottuk meg a közelítő függvényünket. Feltettük azonban az átmenetvalószínűségek irreducibilitását, melyet nem mindig tudjuk garantálni, így ennek következtében előfordulhat, hogy olyan állapotba jut a rendszer, amiből nem tud kilépni vagy néhány állapotot nem látogatunk meg elégszer, így a közelítő értékelőfüggvényünk abban az állapotban nagyon eltérhet a politika valódi értékelőfüggvényének értékétől. Ennek áthidalására kettő közismert módszert mutatok be.

Az első módszer lényege, hogy a kezdőállapotot egy fix ζ eloszlásból sorsolja ki, a trajektóriát pedig – ugyanúgy mint a fentiekben – a μ politikát követve generálja annyi különbséggel, hogy minden lépésben egy fix valószínűség szerint abbamarad a folyamat és elölről indul, a kezdőállapotot pedig ismét a ζ eloszlás szerint sorsolja ki. Ha az s-edik trajektória állapotait $i_{0,s},\dots,i_{N_s-1,s}$-el jelöljük, akkor az s-edik trajektória ℓ-edik állapotához tartozó költség az iteráció aktuális r_k tagja szerint legyen

$$c_{\ell,s}(r_k) = \alpha^{N_s-\ell}\Phi(i_{N_s-1,s})^T r_k + \sum_{h=\ell}^{N_s-1} \alpha^{h-\ell} g\big(i_{h,s}, i_{h+1,s}\big).$$

Ha a k-adik iterációban az összes trajektória az összes $i_{\ell,s}$ állapotára kiszámoltuk a $c_{\ell,s}(r_k)$ költségeket ($s = 1, \ldots, T;\ \ell = 0, \ldots, N_s - 1$), akkor legyen az iteráció következő lépésében r_{k+1} a legkisebb négyzetes eltérés az alábbiak szerint:

$$r_{k+1} = \arg\min_{r\in\mathbb{R}^m} \sum_{s=0}^{T} \sum_{\ell=0}^{N_s-1} \big(\Phi(i_{\ell,s})^T r - c_{\ell,s}(r_k)\big)^2.$$

Belátható, hogy ha a trajektóriák száma, T a végtelenbe tart, akkor r_k a $\Phi r = \Pi_\zeta T^{(\lambda)}\Phi r$ egyenlet fixpontjához tart, ahol Π_ζ a ζ eloszlásvektor szerinti súlyozott projekciót jelenti.

A második módszer azon az ötleten alapul, hogy magukat az átmenetvalószínűségeket módosítjuk olyan módon, hogy ne legyenek elnyelő állapotaink és a szimulációk során minden állapotot elégszer látogassunk meg. Az új átmenetvalószínűségeket az alábbi módon vezetjük be:

$$\overline{P} := (1 - B)P + BQ,$$

ahol B diagonális mátrix β_i komponensekkel, ahol $\beta_i \in [0, 1],\ i = 1, \ldots, n$ és Q pedig egy másik átmenetmátrix. Szemléletesen ez azt jelenti, hogy az i állapotban a következő állapotot $1 - \beta_i$ valószínűséggel választjuk a p_{ij} szerint, és β_i valószínűséggel a q_{ij} szerint.

A $\overline{P}$ átmenetmátrix stacionárius eloszlását jelölje $\overline{\xi}$.

A fenti - új átmenetvalószínűségekhez - vezessük be az alábbi jelöléseket.

$$\overline{g}_i = \sum_{j=1}^{n} \overline{p}_{i,j} g(i, j), \quad i = 1, \ldots, n$$

$$\overline{T}(J) = \overline{g} + \alpha\overline{P}J$$

$$\overline{T}^{(\lambda)}(J) = (1 - \lambda)\sum_{t=0}^{\infty} \lambda^t \overline{T}^{t+1}(J)$$

$$\overline{T}^{(\lambda)}(J) = \overline{g}^{(\lambda)} + \alpha\overline{P}^{(\lambda)}J$$

Ekkor azt az r-t keressük, ami kielégíti a

$$\Phi r = \overline{\Pi}\, T^{(\lambda)}(\Phi r) \tag{10}$$

fixpont-egyenletet.

Ezt a módszert csak a $\lambda = 0$ esetre nézzük végig, nagyobb λ-ra hasonlóan kiterjeszthető, de bonyolultabb felírni.

Mivel $\lambda = 0$, ezért a (10)-es egyenlet szerint a $\Phi r = \overline{\Pi}T(\Phi r)$ fixpont-egyenletet szeretnénk megoldani. Ennek az r^* pontosan akkor megoldása, ha az

$$Ar^* = b$$

egyenletrendszer teljesül, ahol

$$A = \Phi^T diag(\overline{\xi})(I - \alpha P)\Phi,$$
$$b = \Phi^T diag(\overline{\xi})g.$$

Ezt empirikusan úgy tudjuk szimulálni, hogy a $\overline{P}$ szerint léptetjük a trajektóriát, de mindig teszünk egy elágazó lépést a P szerint is, azaz a trajektóriánk $\{(i_0, j_0), (i_1, j_1), (i_2, j_2), \ldots)\}$, ahol az $i_k \to j_k$ lépést P szerint, az $i_k \to i_{k+1}$ lépést $\overline{P}$ szerint tettük meg.

Legyenek

$$A_k = \frac{1}{k+1}\sum_{h=0}^{k}\Phi(i_h)\big(\Phi(i_h)-\alpha\Phi(j_h)\big)^T$$

$$b_k = \frac{1}{k+1}\sum_{h=0}^{k}\Phi(i_h)g(i_h,j_h).$$

A nagy számok törvénye alapján teljesül az $A_k \to A$, $b_k \to b$ konvergencia.

A fenti három közelítő módszernek a megfelelője itt is definiálható:
Az *LSTD(0)* eszerinti verziójának az $A_k^{-1}b_k$ által adott eredményt tekintjük.

Az *LSPE(0)*-át ezesetben az $r_{k+1} = r_k - \frac{\gamma}{k+1}G_k\sum_{h=0}^{k}\Phi(i_h)\tilde{q}_{k,h}$ iterációval definiáljuk, ahol $\tilde{q}_{k,h} \doteq \Phi(i_h)^T r_k \cdot \alpha \cdot \Phi(j_h)^T r_k - g(i_h,j_h)$.

A *TD(0)* módszer az alábbi iterációval van definiálva: $r_{k+1} = r_k - \gamma_k\Phi(i_k)\tilde{q}_{k,k}$.

Felhasznált irodalom

[1] Bertsekas, Dimitri P., and John N. Tsitsiklis. "Neuro-dynamic programming: an overview." Proceedings of 1995 34th IEEE conference on decision and control. Vol. 1. IEEE, 1995.

[2] Bertsekas, Dimitri. Dynamic programming and optimal control: Volume I. Vol. 1. Athena scientific, 2012.

[3] Bertsekas, Dimitri P. "Dynamic programming and optimal control 3rd edition, volume ii." Belmont, MA: Athena Scientific (2011).

[4] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.