מקליד: עידן אריה
מתרגל: דוד יצחקי
תאריך: 2026-05-07

בינה מלאכותית
89-5570-10

תכנון סטוכסטי

הגדרה: Markov Decision Process הוא $\langle S, A, T, R\rangle$ כאשר:

S אוסף סופי של מצבים $|S| = n$. מקבלים אותו קומפקטי.

A אוסף סופי של פעולות.

T פונקציית מעברים $T : S \times A \times S \to [0, 1]$:

$$T(s, a, s') = P(s'|s, a)$$

R פונקציית reward $R : S \to [0, r\max]$

הנחות:

- דינמיקה מרקובית (Markovian dynamics - history independence):

$$P\left(S^{t+1}\middle|S^t, A^t, S^{t-1}, A^{t-1}, \ldots, S^0\right) = P\left(S^{t+1}\middle|S^t, A^t\right)$$

כלומר אין משמעות להיסטוריה אלא רק למצב הנוכחי והפעולה שמבצעים בו.

- סטצינוריות (Stationary dynamics) - אין תלות בזמן:

$$\forall_{k\neq t} P\left(S^{t+1}\middle|S^t, A^t\right) = P\left(S^{k+1}\middle|S^k, A^k\right)$$

- תגמול מרקובי (Markovian reward) - ההסתברות לתגמול - גם אם היא סטוכסטית - תלויה אך ורק במצב:

$$P\left(R^t\middle|S^t, A^t, S^{t-1}, A^{t-1}, \ldots, S^0\right) = P\left(R^t\middle|S^t\right)$$

- נראות מלאה (Full observability). אנחנו רואים את כל העולם:
  - אנחנו יודעים את כל ההסתברויות.
  - אנחנו יודעים את כל התגמולים.

בתכנון קלאסי הפתרון היה תוכנית - סדרה של פעולות $(a_1, a_2, \ldots, a_k)$. כאן אי אפשר - אנחנו לא יודעים בוודעות איפה נהיה אחרי שנבצע a_1, ויכול להיות שבכלל לא יהיה אפשר לבצע את a_2!

הערה: בעולם האמיתי כמות המצבים גדולה מאוד, ונרצה למצוא ייצוג קומקפטי למודל.

מכיוון שסדרת פעולות לא ישימה כאן, נשתמש בפוליסה - פונקציה שאומרת איזה פעולה לבצע. יש שני סוגי פוליסות:

- סטציונרית: $\pi : S \to A$
- לא־סטציונרית: $\pi : S \times T \to A$. תלויה בזמן (כמות הצעדים שנשארו לנו)

מכיוון שהתוצאה (ולפעמים גם התגמול) של כל פעולה סטוכסטי, מסתכלים על תוחלת התגמול. בשביל זה אנחנו רוצים value function:

$$V_\pi(s) = \mathbb{E}\left[\text{following policy } \pi | \text{starting from state } s\right]$$

הפונקציה תלויה בתגמול המיידי שנקבל ע"י הפעלת $\pi(s)$ - אבל גם בתגמול העתידי שנקבל כאשר נמשיך להפעיל את הפונקציה.
יש לנו שני סוגי value function:

- תלוי בזמן - finite-horizon MDP - $V : S \times T \to \mathbb{R}$. מתחשב במספר הצעדים שנשאר לנו.
- לא תלוי בזמן - infinite-horizon MDP - $V : S \to \mathbb{R}$. מניח שאין הגבלת צעדים.

המקרה הסופי

המקרה הסופי תלוי בזמן. יש לנו אופק H שאומר כמה צעדים נשאר לנו. יש לנו שני אלגוריתמים חשובים:

- Policy Evaluation - בהינתן פוליסה לא־סטצינורית π, ומספר צעדים שנשארו באופק $h \in \{0, 1, \ldots, H\}$, מחשבים את $V_\pi^h(s)$.

  נחשב את זה באמצעות תכנות דינאמי:

$$V_\pi^0(s) = R(s)$$

$$V_\pi^h(s) = R(s) + \sum_{s' \in S} P(s'|s, \pi(s,h)) \cdot V_\pi^{h-1}(s')$$

  זמן החישוב של זה הוא $O\left(H \cdot |S|^2\right)$. הבעיה היא שיש יותר מדי פוליסות - כמות אקספוננציאלית, $|A|^{H \cdot |S|}$ - וצריך לחשב את זה לכל פוליסה.

- Policy Optimization בהינתן MDP ואופק H, מחשבים את הפוליסה האופטימלית π^*.

  בשביל לחשב את זה, נלך מהסוף. עבור $h = 0$ כל פוליסה היא אופטימלית, ובהינתן הפוליסה על h אפשר לחשב את הפוליסה האופטימלית על $h+1$:

$$\pi^*(s,t) = \arg\max_{a \in A} \left[R(s) + \sum_{s' \in S} P(s'|s,a) \cdot V_{\pi^*}^{t-1}(s')\right]$$

  הסיבוכיות של זה היא $O\left(H \cdot |A| \cdot |S|^2\right)$ - אבל צריך לחשב את זה רק פעם אחת ולא על כל פוליסה.

מקליד: עידן אריה
מתרגל: דוד יצחקי
תאריך: 2026-05-07

בינה מלאכותית
89-5570-10

המקרה האינסופי

אנחנו מחפשים כאן פוליסה סטציונרית, כי תמיד נשארו לנו אינסוף צעדים. אבל אם האופק אינסופי, ויש לנו אפשרות לתגמול חיובי, אז הערך יהיה אינסופי. איך מתמודדים עם זה? נשתמש בdiscount future rewards - $0 \leq \gamma < 1$ כדי ליצור טור הנדסי מתכנס[1].

- עבור γ קטן פחות איכפת לנו מהעתיד ויותר מshort term rewards.
- עבור γ גדול איכפת לנו יותר מהעתיד.

לפי הγ אפשר לחשב ערך:

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \cdot R\left(S^t\right) \middle| S^0 = s, \pi\right]$$

בשביל לחשב את זה נשתמש במשוואת בלמן:

$$V_\pi(s) = R(s) + \gamma \sum_{s' \in S} P\left(s' | s, \pi(s)\right) V_\pi\left(s'\right)$$

הפעם, בגלל שהאופק אינסופי, אנחנו לא יכולים להשתמש בתכנות דינמי, ולכן נצטרך לחשב:

$$\left(I - \gamma P_\pi\right) \cdot V_\pi = R$$

גם כאן אפשר לעשות value iteartion או policy iteration.

value iteration

הערך האופטימלי הוא נקודת שבת של אופרטור בלמן, שכן:

$$\left\|B[V] - B\left[V'\right]\right\|_\infty \leq \gamma \cdot \left\|V - V'\right\|_\infty$$

לכן נתחיל מנקודה שרירותית $V^{(0)}$ ונפעיל את אופרטור בלמן עד שהשיפור יהיה קטן מϵ.

policy iteration

מתחילים עם פוליסה שרירותית, ובלולאה מחשבים:

$$V_\pi(s) = R(s) + \gamma \sum_{s'} P\left(s' | s, \pi(s)\right) V_\pi\left(s'\right)$$

לפי זה מחשבים פוליסה חדשה - לכל מצב מחשבים איזו פעולה נותנת הכי הרבה ערך בהנחה שממשיכים עם פוליסה שהערך שלה הוא V_π.

$$\pi(s) \leftarrow \arg\max_a \left[R(s) + \gamma \sum_{s'} P\left(s' | s, a\right) V_\pi\left(s'\right)\right]$$

בלמן מבטיח שזה יתן לנו פונקציה יותר טובה. מכיוון שכמות הפוליסות היא סופית, בסופו של דבר נגיע לאיטרציה שבה אין שינוי - ואז נדע שהגענו לפוליסה האופטימלית.

[1] לא לחשוב על זה בתור ההסתברות להמשיך לשחק. רק בתור "כמה איכפת לנו מהעתיד". משחקים תמיד עד אינסוף.

מה עדיף?

תלוי בבעיה. אם יש הרבה מצבים אבל מעט פוליסות אפשריות, אז עדיף policy iteration.