برنامه ریزی پویا تحت دی ای ۳ ان: نزدارس | یادگیری تقویتی ۱۱ سینزیک

موضوعات درس: ۱- برنامه ریزی پویا   ۲- ارزیابی سیاست   ۳- بهبود سیاست   ۴) تکرار سیاست   ۵- تکرار ارزش   ۶ تکرار سیاست تعمیم یافته

برنامه ریزی پویا: مجموعه‌ای از روش‌ها برای تعیین سیاست بهینه با داشتن مدل EMDP است.

در این بین فرض می‌کنیم $p(s', r | s, a)$ را داریم.

تبصره: فضای شرایط مسئله باید بسته و محدود باشد.

در عمل مدل اغلب مسائل یادگیری تقویتی نامشخص است. $p(s', r | s, a)$

محاسبات ریاضی برنامه ریزی پویا خیلی زیاد است.

اما برنامه ریزی پویا درک مفاهیم و روش‌های Model Free را ساده‌تر می‌کند.

معادله بهینگی بلمن برای تابع ارزش حالت:

$$v_*(s) = \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a]$$
$$= \max_a \sum_{s', r} p(s', r | s, a)[r + \gamma v_*(s')]$$

← حل تحلیلی غیرممکن

$$q_*(s,a) = E[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a]$$
$$= \sum_{s', r} p(s', r | s, a)[r + \gamma \max_{a'} q_*(s', a')]$$

← " " " " " "

ارزیابی سیاست:

در برنامه ریزی پویا با روش‌های عددی تکراری توابع ارزش و سیاست تخمین زده می‌شوند.

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

شرط وجود جواب یکتا: نرخ تخفیف (discount factor) کوچکتر از یک و رسیدن به حالت نهایی تحت سیاست ← $\gamma < 1$ باشد و همچنین تحت این سیاست π باید بتوانیم از هر حالت اولیه به هر حالت نهایی برسیم. هر وقت می‌خواهیم ارزش تمامی state‌ها را بدست آوریم.

چطور حل کنیم؟ از روش ژاکوبی یکی از تکنیک‌های رایج برای حل عددی دستگاه معادلات خطی است.

$$\begin{cases} E(1) = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ E(2) = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ E(n) = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

$$x^{(0)} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix}$$
مقدار اولیه

s.a.m

روش ژاکوبی

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad B=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad x, A^{-1}B, A^{-1}B$$

قدم اول روش ژاکوبی: در معادله iام متغیر iام برحسب سایر مجهولات تعیین شود.

$$x_1 = \frac{b_1 - [a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n]}{a_{11}}$$

$$x_2 = \frac{b_2 - [a_{21}x_1 + a_{23}x_3 + \cdots + a_{2n}x_n]}{a_{22}}$$

$$\vdots$$

$$x_n = \frac{b_n - [a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{n,n-1}x_{n-1}]}{a_{nn}}$$

ژاکوبی $\longrightarrow$ $x_i = \dfrac{b_i - \sum_{j\neq i} a_{ij}x_j}{a_{ii}}$

قدم دوم روش ژاکوبی: به همین ترتیب برحله اول

$$x_1^{(1)} = \frac{b_1 - [a_{12}x_2^{(0)} + a_{13}x_3^{(0)} + \cdots + a_{1n}x_n^{(0)}]}{a_{11}}$$

به همین ترتیب برای سایر x_i ها عمل می‌کنیم.

ژاکوبی: $x_i^{(k)} = \dfrac{b_i - \sum_{j\neq i} a_{ij}x_j^{(k-1)}}{a_{ii}} \qquad k=1,2,\cdots$

while $\|x^k - x^{k-1}\| < \varepsilon$ مثلا $\varepsilon = 0.001$

معادله بلمن برای تابع ارزش حل به روش ژاکوبی است.

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

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_k(s')] \qquad k=1,2,\cdots$$

تخمین جدید

$$\vdots$$

iterative policy evaluation → تکراری انجام میدهیم — الگوریتم ارزیابی سیاست به منظور تعیین تابع ارزش حالت

Input π, the policy to be evaluated

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation

دقت تعیین ارزش که برای توقف الگوریتم استفاده میشود در ژاکوبی

$$|v_{k+1}(s) - v_k(s)| < \theta$$

Initialize $V(s)$, for all $s \in S^+$, arbitrarily except that $V(terminal) = 0$

مقدار اولیه باید بدیم

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in S$:

$v \leftarrow V(s)$

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

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

روش های ژاکوبی:

$x_1^{n+1} = f(x_1^n, x_2^n)$

$x_2^{n+1} = g(x_1^n, x_2^n)$

روش گوس سایدل:

$x_1^{n+1} = f(x_1^n, x_2^n)$

$x_2^{n+1} = g(x_1^{n+1}, x_2^n)$

سیاست تصادفی: در هر حالت احتمال انجام همه عمل ها وجود دارد.

برای هر State برابر یا بزرگتر از صفر احتمال انتخاب همه اکشن ها وجود دارد.

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

سیاست قطعی: در هر حالت فقط یک عمل مشخص قابل انجام است.

$s_1 \rightarrow a_1$

$s_i \rightarrow a_i$

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

اکشنی که تحت سیاست قطعی انتخاب می شود.

* همواره می توان از روی یک سیاست یک سیاست بهتر تولید کرد.

سیاست بهتر = سیاستی که منجر به ارزش های بیشتری شود.

$q_\pi(s,a)$ — اکشن را تحت π / اکشن را انتخاب می کنیم (تغییر)

$q_\pi(s, \pi'(s)) \geq v_\pi(s)$ → policy improvement theorem

$\forall s$

$v_{\pi'}(s) \geq v_\pi(s)$

★ با تعیین ارزش حالت s نسبت به سیاست اولیه، می‌توان سیاست بهبود یافته را تعیین کرد.

اگر v_π یا q_π را داشته باشیم میشه به π' رسید.

تابع ارزش حالت عمل تحت سیاست اولیه:
$$q_\pi(s,a) \doteq E[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_\pi(s')]$$

چه اکشنی باید انتخاب بشود تا این عبارت max شود.

← سیاست جدید نسبت به تابع ارزش حالت عمل greedy می‌شود.

$$\pi'(s) = \underset{a}{\operatorname{argmax}}\ q_\pi(s,a)$$
$$= \underset{a}{\operatorname{argmax}}\ E[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \underset{a}{\operatorname{argmax}} \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_\pi(s')]$$

اگر فقط v_π را داشته باشیم به مدل MDP نیاز است.

به همین دلیل سیاست جدید بهتر از سیاست قبلی است

برای بهبود کافی است ارزش حالتی که در آن حالت بیشتر شود.

چون داریم به صورت تکراری ارزش‌ها را تخمین می‌زنیم، $\pi'(s)$ با بهینه سازی کوتاه مدت بر مبنای این

← بهبود greedy از طریق انتخاب عمل برای ماکزیمم کردن پاداش کوتاه مدت است.

اما چون دینامیک سیستم و Return در الگوریتم بهبود سیاست حضور دارند، در بلند مدت Greedy خواهد بود.

★ اگر تابع ارزش سیاست اولیه و سیاست بهبود یافته یکسان شوند به سیاست بهینه رسیده‌ایم.

$$v_{\pi'}(s) = \max_a E[R_{t+1} + \gamma v_{\pi'}(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_{\pi'}(s')]$$

$v_{\pi'} = v^*$ ↓ π^*

v_π

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$ و $$v^* = \max_a q^*(s,a)$$

تکرار سیاست: بعد از تعیین سیاست جدید π' باشه می‌توانیم تابع ارزش حالت آن $v_{\pi'}$ را هم محاسبه کنیم و سپس از روی آن مجددا به سیاست بهتری برسیم.

سیکل ارزیابی سیاست و بهبود سیاست تا رسیدن به سیاست بهینه می‌تواند تکرار شود.

s.a.m

$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$ بهینه

هر سیاست جدید اکیداً بهتر از سیاست قبلی است مگر اینکه به سیاست بهینه برسیم.

چون MDP دارای حالت ها و سیاست های محدودی است، تکرار سیاست به سیاست بهینه همگرا خواهد شد.

الگوریتم تکرار سیاست: ← تو نوشتن برای پیدا کردن

1. Initialization

$V(s) \in \mathbb{R}$ and $\pi(s) \in A(s)$ arbitrarily for all $s \in S$

2. Policy Evaluation

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in S$:

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) [r + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$ (a small positive number determining the accuracy of estimation.

S^+ ← تمام state های که در محیط داریم

S ↳ همه S^+ است جز Terminal state

3. Policy Improvement

policy stable ← true

For each $s \in S$:

old-action ← $\pi(s)$

$\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) [r + \gamma V(s')]$

if old-action $\neq \pi(s)$, then policy-stable ← false

If policy-stable, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

تضمین می‌شود برویم به سیاست ثابت برسیم.

تکرار ارزش:

الگوریتم نقطه ثابت Fixed point

برای حل معادله $f(x)=0$ آن را به صورت $x - g(x) = 0$ می‌نویسیم.

تکرار تا رسیدن به همگرایی: $x_{k+1} = g(x_k)$

این معادله بلمن را می‌یابیم سیستم تقریب می‌زنیم

شرایط یکتایی ریشه:

$f(x)$ در بازه $[a,b]$ پیوسته باشد.

$|f'(x)|$ در بازه (a,b) عددی و کوچکتر از یک باشد.

یکی از معایب **تکرار سیاست** ارزیابی سیاست در هر گام است.

ارزیابی سیاست شامل تکرار یک حلقه طولانی تا رسیدن به دقت مورد نظر است.

در الگوریتم **تکرار ارزش** معادلات بهینگی بلمن برای تابع ارزش به صورت تکراری حل می‌شوند و نیازی به سیاست اولیه نیست.

با تکرار تنها یک حلقه ارزش‌های بهینه محاسبه و سپس سیاست بهینه تعیین خواهد شد.

در Policy iteration از اول یک سیاست تقریبی مشخص داریم که این سیاست باعث می‌شد الگوریتم در ابتدا خیلی کند عمل کند.

$$v_*(s) = \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$

$$= \max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_*(s')] \longrightarrow v_* \longrightarrow \pi_*$$

بازنویسی

$$v_{k+1}(s) = \max_a E[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a]$$

$$= \max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_k(s')]$$

الگوریتم تکرار ارزش:

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation

Initialize $V(s)$ for all $s \in S^+$, arbitrarily except that $V(terminal) = 0$

Loop:

  $\Delta \leftarrow 0$

  Loop for each $s \in S$:

    $v \leftarrow V(s)$

    $V(s) \leftarrow \max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma V(s')]$

    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$ such that

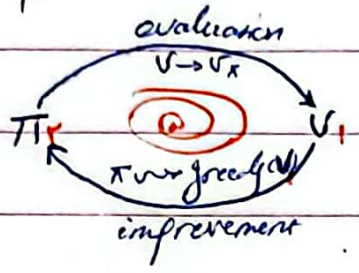
$\pi(s) = \arg\max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma V(s')]$

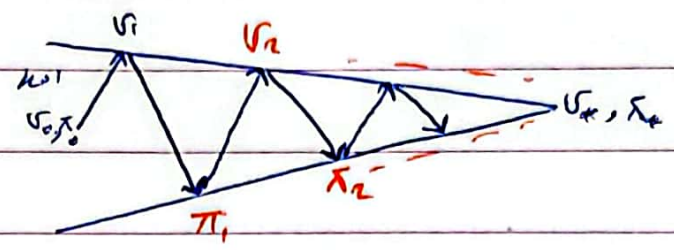
$\max_x f(x) \in R_f$ → برد تابع

$\arg\max_x f(x) \in D_f$ → دامنه

تکرار سیاست تعمیم یافته — Generalized policy iteration (GPI)

در تکرار سیاست باید مکرر ارزیابی تا همگرایی توابع ارزش حالت تکرار شوند.
در طول ارزیابی سیاست، خود سیاست ثابت می ماند.
در الگوریتم **تکرار سیاست تعمیم یافته (GPI)** در پایان هر مرحله ارزیابی چند سیاست هم بهبود داده خواهد شد. (مانند تعمیم روش زاویه به روش بابل)





s.a.m