روش تفاضل‌های زمانی

سیّد سینا ایروانیان

دانشگاه صنعتی شریف

۵ خرداد ۱۳۸۸

روش تفاضل‌های زمانی

روش تفاضل‌های زمانی یک روش یادگیری است

- یادگیری برای پیش‌بینی وضعیت یک سیستم پویا در دراز مدّت
- یادگیری برای کنترل $\Leftarrow$ کاربرد در یادگیری تقویتی

انواع روش‌های یادگیری:

- **یادگیری بانظارت:** عامل با یک سری ورودی و پاسخ صحیح از قبل تعیین شده آموزش می‌بیند
- **یادگیری بی‌نظارت:** عامل هیچ بازخوردی دریافت نمی‌کند؛ ورودی‌ها را بصورت خوشه‌ها، یا طبقه‌بندی‌ها بازسازی می‌کند
- **یادگیری تقویتی:** یادگیری از طریق تعامل با محیط، و سعی و خطا توسط خود عامل

یادگیری به روش تفاضل‌های زمانی از نوع یادگیری تقویتی به‌حساب می‌آید

روش تفاضل‌های زمانی

روش تفاضل‌های زمانی یک روش یادگیری است

- یادگیری برای پیش‌بینی وضعیت یک سیستم پویا در دراز مدّت
- یادگیری برای کنترل $\Leftarrow$ کاربرد در یادگیری تقویتی

انواع روش‌های یادگیری:

- **یادگیری بانظارت:** عامل با یک سری ورودی و پاسخ صحیح از قبل تعیین شده آموزش می‌بیند
- **یادگیری بی‌نظارت:** عامل هیچ بازخوردی دریافت نمی‌کند؛ ورودی‌ها را بصورت خوشه‌ها، یا طبقه‌بندی‌ها بازسازی می‌کند
- **یادگیری تقویتی:** یادگیری از طریق تعامل با محیط، و سعی و خطا توسط خود عامل

یادگیری به روش تفاضل‌های زمانی از نوع یادگیری تقویتی به‌حساب می‌آید

روش تفاضل‌های زمانی

روش تفاضل‌های زمانی یک روش یادگیری است

- یادگیری برای پیش‌بینی وضعیت یک سیستم پویا در دراز مدّت
- یادگیری برای کنترل $\Leftarrow$ کاربرد در یادگیری تقویتی

انواع روش‌های یادگیری:

- **یادگیری بانظارت:** عامل با یک سری ورودی و پاسخ صحیح از قبل تعیین شده آموزش می‌بیند
- **یادگیری بی‌نظارت:** عامل هیچ بازخوردی دریافت نمی‌کند؛ ورودی‌ها را بصورت خوشه‌ها، یا طبقه‌بندی‌ها بازسازی می‌کند
- **یادگیری تقویتی:** یادگیری از طریق تعامل با محیط، و سعی و خطا توسط خود عامل

یادگیری به روش تفاضل‌های زمانی از نوع یادگیری تقویتی به‌حساب می‌آید

پیش‌بینی چندمرحله‌ای

مسئله‌ی پیش‌بینی چند مرحله‌ای

- **ورودی:** دنباله‌ی مشاهدات–نتیجه: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, z$
- **خروجی:** $P_1, P_2, \ldots, P_m$
- $\mathbf{x}_t$ بردار مشاهده در مرحله‌ی t با درایه‌های حقیقی (برداری از اندازه‌گیری‌ها یا خصوصیات محیط) است
- z یک اسکالر حقیقی و نتیجه‌ی دنباله‌ی مشاهدات است
- P_t پیش‌بینی مقدار z که در مرحله‌ی t انجام شده، P تابعی از $\mathbf{x}_t$ و بردار وزن‌ها $\mathbf{w}$، (برداری از پارامترهای تغییر پذیر) است و آن را با $P(\mathbf{x}_t, \mathbf{w})$ نشان می‌دهند، به P تابع پیش‌بینی می‌گویند.

مثال

پیش‌بینی این که آیا روز جمعه باران می‌بارد یا خیر، از طریق مشاهده‌ی وضعیت هوای دوشنبه، و انجام پیش‌بینی برای روزهای متوالی تا جمعه

پیش‌بینی چندمرحله‌ای

مسئله‌ی پیش‌بینی چند مرحله‌ای

- **ورودی:** دنباله‌ی مشاهدات–نتیجه: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, z$
- **خروجی:** $P_1, P_2, \ldots, P_m$
- $\mathbf{x}_t$ بردار مشاهده در مرحله‌ی t با درایه‌های حقیقی (برداری از اندازه‌گیری‌ها یا خصوصیات محیط) است
- z یک اسکالر حقیقی و نتیجه‌ی دنباله‌ی مشاهدات است
- P_t پیش‌بینی مقدار z که در مرحله‌ی t انجام شده، P تابعی از $\mathbf{x}_t$ و بردار وزن‌ها $\mathbf{w}$، (برداری از پارامترهای تغییر پذیر) است و آن را با $P(\mathbf{x}_t, \mathbf{w})$ نشان می‌دهند، به P تابع پیش‌بینی می‌گویند.

مثال

پیش‌بینی این که آیا روز جمعه باران می‌بارد یا خیر، از طریق مشاهده‌ی وضعیت هوای دوشنبه، و انجام پیش‌بینی برای روزهای متوالی تا جمعه

پیش‌بینی تک‌مرحله‌ای

مسئله‌ی پیش‌بینی تک مرحله‌ای

- پیش‌بینی چندمرحله‌ای به ازای $m = ۱$
- دیگر در زمره‌ی مسائل تفاضل‌های زمانی قرار نمی‌گیرد
- در دسته‌ی یادگیری با نظارت قرار می‌گیرد (چون بلافاصله پس از پیش‌بینی، نتیجه نیز ارائه می‌شود)

مثال

پیش‌بینی این که آیا روز جمعه باران می‌بارد یا خیر، از طریق مشاهده‌ی وضعیت هوای پنج‌شنبه

پیش‌بینی تک‌مرحله‌ای

مسئله‌ی پیش‌بینی تک مرحله‌ای

- پیش‌بینی چندمرحله‌ای به ازای $m = ۱$
- دیگر در زمره‌ی مسائل تفاضل‌های زمانی قرار نمی‌گیرد
- در دسته‌ی یادگیری با نظارت قرار می‌گیرد (چون بلافاصله پس از پیش‌بینی، نتیجه نیز ارائه می‌شود)

مثال

پیش‌بینی این که آیا روز جمعه باران می‌بارد یا خیر، از طریق مشاهده‌ی وضعیت هوای پنج‌شنبه

یادگیری با نظارت

- منظور از یادگیری، تنظیم مقادیر بردار وزن‌ها $\mathbf{w}$ است، به‌طوریکه خطای تابع پیش‌بینی در هر مرحله کمینه شود.
- به‌ازای هرمرحله می‌توان تغییر در بردار وزن‌ها $\Delta\mathbf{w}_t$ را در نظر گرفت

$$\mathbf{w} \leftarrow \mathbf{w} + \sum_{t=1}^{m} \Delta\mathbf{w}_t \qquad (1)$$

- در یادگیری با نظارت تمام تغییرات در انتهای دنباله‌ی مشاهدات اعمال می‌شود؛ نه در هر مرحله

یادگیری با نظارت (ادامه)

- یادگیری با نظارت به هر دنباله‌ی مشاهدات–نتیجه، به صورت دنباله‌ای از زوج‌های مشاهده–نتیجه برخورد می‌کند

$$(\mathbf{x}_1, z), (\mathbf{x}_2, z), \ldots, (\mathbf{x}_m, z)$$

- تغییرات در بردار وزن‌ها به‌ازای مشاهده در لحظه‌ی t:

$$\Delta \mathbf{w}_t = \alpha(z - P_t)\nabla_w P_t \qquad (2)$$

- α: نرخ یادگیری
- $\nabla_w P_t$: بردار مشتقّات جزئی P، نسبت به $\mathbf{w}$؛ مشخص کننده‌ی جهتی که در آن تغییرات در $\mathbf{w}$ بیشترین تأثیر را در P دارد.

یادگیری با نظارت (ادامه)

حالت خاصّ خطی

- حالت خاص: P_t تابع خطّی از $\mathbf{x}_t$ و $\mathbf{w}$ است:

$$P_t = \mathbf{w}^T \mathbf{x}_t = \sum_i \mathbf{w}(i)\mathbf{x}(i)$$

- خواهیم داشت: $\nabla_w P_t = \mathbf{x}_t$

قانون Widrow-Hoff برای بروزرسانی وزن‌ها – قانون دلتا

$$\Delta \mathbf{w}_t = \alpha(z - \mathbf{w}^T \mathbf{x}_t)\mathbf{x}_t \qquad (۳)$$

- مقدار z در انتهای دنباله‌ی مشاهدات مشخص می‌شود، بنابراین تمامی $\mathbf{x}_t$ ها باید ذخیره شوند، و تمامی بروزرسانی‌ها در انتهای دنباله محاسبه شوند.

یادگیری با نظارت (ادامه)

حالت خاصّ خطی

- حالت خاص: P_t تابع خطّی از $\mathbf{x}_t$ و $\mathbf{w}$ است:

$$P_t = \mathbf{w}^T \mathbf{x}_t = \sum_i \mathbf{w}(i)\mathbf{x}(i)$$

- خواهیم داشت: $\nabla_w P_t = \mathbf{x}_t$

قانون Widrow-Hoff برای بروزرسانی وزن‌ها – قانون دلتا

$$\Delta \mathbf{w}_t = \alpha(z - \mathbf{w}^T \mathbf{x}_t)\mathbf{x}_t \qquad (۳)$$

- مقدار z در انتهای دنباله‌ی مشاهدات مشخص می‌شود، بنابراین تمامی $\mathbf{x}_t$ ها باید ذخیره شوند، و تمامی بروزرسانی‌ها در انتهای دنباله محاسبه شوند.

یادگیری با نظارت (ادامه)

حالت خاصّ خطی

- حالت خاص: P_t تابع خطّی از $\mathbf{x}_t$ و $\mathbf{w}$ است:

$$P_t = \mathbf{w}^T \mathbf{x}_t = \sum_i \mathbf{w}(i)\mathbf{x}(i)$$

- خواهیم داشت: $\nabla_w P_t = \mathbf{x}_t$

قانون Widrow-Hoff برای بروزرسانی وزن‌ها – قانون دلتا

$$\Delta \mathbf{w}_t = \alpha(z - \mathbf{w}^T \mathbf{x}_t)\mathbf{x}_t \qquad (۳)$$

- مقدار z در انتهای دنباله‌ی مشاهدات مشخص می‌شود، بنابراین تمامی $\mathbf{x}_t$ ها باید ذخیره شوند، و تمامی بروزرسانی‌ها در انتهای دنباله محاسبه شوند.

محاسبه‌ی افزایشی

- نمایش خطای $z - P_t$ به صورت مجموع تغییرات در پیش‌بینی‌های متوالی:

$$z - P_t = \sum_{k=t}^{m} (P_{k+1} - P_k) \quad \text{و} \quad P_{m+1} \overset{\text{تعریف}}{=} z$$

- با ترکیب با (۱) و (۲)، روابط زیر بدست می‌آیند:

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{w} + \sum_{t=1}^{m} \alpha (z - P_t) \nabla_w P_t \\ &= \mathbf{w} + \sum_{t=1}^{m} \alpha \sum_{k=t}^{m} (P_{k+1} - P_k) \nabla_w P_t \end{aligned}$$

محاسبه‌ی افزایشی

- نمایش خطای $z - P_t$ به صورت مجموع تغییرات در پیش‌بینی‌های متوالی:

$$z - P_t = \sum_{k=t}^{m} (P_{k+1} - P_k) \quad \text{و} \quad P_{m+1} \overset{\text{تعریف}}{=} z$$

- با ترکیب با (۱) و (۲)، روابط زیر بدست می‌آیند:

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{w} + \sum_{t=1}^{m} \alpha (z - P_t) \nabla_w P_t \\ &= \mathbf{w} + \sum_{t=1}^{m} \alpha \sum_{k=t}^{m} (P_{k+1} - P_k) \nabla_w P_t \end{aligned}$$

محاسبه‌ی افزایشی (ادامه)

- با جابجایی دو جمع و تبدیل حدود

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} + \sum_{k=1}^{m} \alpha \sum_{t=1}^{k} (P_{k+1} - P_k) \nabla_w P_t \\ &= \mathbf{w} + \sum_{t=1}^{m} \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_k\end{aligned}$$

رابطه‌ی بروز رسانی افزایشی وزن‌ها – TD(1)

$$\Delta \mathbf{w}_t = \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_k \qquad (۴)$$

محاسبه‌ی افزایشی (ادامه)

- با جابجایی دو جمع و تبدیل حدود

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} + \sum_{k=1}^{m} \alpha \sum_{t=1}^{k} (P_{k+1} - P_k) \nabla_w P_t \\ &= \mathbf{w} + \sum_{t=1}^{m} \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_k\end{aligned}$$

رابطه‌ی بروز رسانی افزایشی وزن‌ها – TD(1)

$$\Delta \mathbf{w}_t = \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_k \qquad (۴)$$

TD(1)

- اگر بیشترین طول دنباله‌ی مشاهدات M باشد، آن‌گاه الگوریتم TD(1) نیاز به $\frac{1}{M}$ حافظه و سرعت پردازشگر لازم برای یادگیری با نظارت دارد.
- اگر تابع پیش‌بینی را تابع خطّی در نظر بگیریم

رابطه‌ی بروز رسانی TD(1) خطّی

$$\Delta \mathbf{w}_t = \alpha(P_{t+1} - P_t) \sum_{k=1}^{t} \mathbf{x}_k \qquad P_t = \mathbf{w}^T \mathbf{x}_t$$

الگوریتم TD(1) خطّی

ورودی: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, z$ ▷ دنباله‌ی مشاهدات–نتیجه

خروجی: $\mathbf{w}$ ▷ بردار وزن‌ها

$\mathbf{w} \leftarrow \mathbf{w}_0$ ▷ بردار وزن‌ها را با مقادیر تصادفی مقداردهی اوّلیه کن

$P_t \leftarrow \mathbf{w}^T \mathbf{x}_1$ ▷ $\mathbf{x}_1$ مشاهده شد

$\mathbf{S}_t \leftarrow \mathbf{x}_1$ ▷ $\mathbf{S}_t$ همان جمع گرادیان‌هاست

for all $\mathbf{x}_t, t = 2, \ldots, m$ **do** ▷ برای مشاهدات $\mathbf{x}_2$ تا $\mathbf{x}_m$

  $P_{t-1} \leftarrow P_t$

  $P_t \leftarrow \mathbf{w}^T \mathbf{x}_t$

  $\Delta \mathbf{w}_{t-1} \leftarrow \alpha(P_t - P_{t-1})\mathbf{S}_t$ ▷ میزان افزایش بردار وزن‌ها برای مشاهده‌ی قبلی

  $\mathbf{S}_t \leftarrow \mathbf{S}_t + \mathbf{x}_t$

  $\mathbf{w} \leftarrow \mathbf{w} + \Delta \mathbf{w}_{t-1}$ ▷ بروزرسانی بردار وزن‌ها

end for

$\Delta \mathbf{w}_{t-1} \leftarrow \alpha(z - P_t)\mathbf{S}_t$

$\mathbf{w} \leftarrow \mathbf{w} + \Delta \mathbf{w}_{t-1}$ ▷ بروزرسانی نهایی بردار وزن‌ها

خانواده‌ی روش‌های یادگیری TD(λ)

- در TD(1)، به ازای هر مشاهده، بردار وزن‌ها طوری تغییر می‌کند، که تمام پیش‌بینی‌های گذشته را به یک میزان تغییر می‌دهد
- کلاس روش‌های TD(λ)، پیش‌بینی‌های جدیدتر را بیش‌تر از پیش‌بینی‌های گذشته تغییر می‌دهد
- با وزن‌دهی نمایی با تأخّر، تغییر در پیش‌بینی مشاهداتی که در k مرحله پیش انجام شدند متناسب است با λ^k برای $۰ \leq \lambda \leq ۱$

رابطه‌ی بروزرسانی وزن‌ها TD(λ)

$$\Delta \mathbf{w}_t = \alpha(P_{t+۱} - P_t) \sum_{k=۱}^{t} \lambda^{t-k} \nabla_w P_k \qquad (۵)$$

خانواده‌ی روش‌های یادگیری TD(λ)

- در TD(1)، به ازای هر مشاهده، بردار وزن‌ها طوری تغییر می‌کند، که تمام پیش‌بینی‌های گذشته را به یک میزان تغییر می‌دهد
- کلاس روش‌های TD(λ)، پیش‌بینی‌های جدیدتر را بیش‌تر از پیش‌بینی‌های گذشته تغییر می‌دهد
- با وزن‌دهی نمایی با تأخّر، تغییر در پیش‌بینی مشاهداتی که در k مرحله پیش انجام شدند متناسب است با λ^k برای $۰ \leq \lambda \leq ۱$

رابطه‌ی بروزرسانی وزن‌ها TD(λ)

$$\Delta \mathbf{w}_t = \alpha(P_{t+۱} - P_t) \sum_{k=۱}^{t} \lambda^{t-k} \nabla_w P_k \qquad (۵)$$

خانواده‌ی روش‌های یادگیری TD(λ)(ادامه)

- مزیّت وزن‌دهی نمایی در این است که رابطه را می‌توان به‌صورت افزایشی محاسبه کرد
- مثلاً اگر مقدار جمع در (۵) را برای مرحله‌ی t، با s_t نمایش دهیم، در این صورت مقدار s_{t+1} را می‌توان به صورت افزایشی محاسبه کرد

$$\begin{aligned} s_{t+1} &= \sum_{k=1}^{t+1} \lambda^{t+1-k} \nabla_w P_k \\ &= \nabla_w P_{t+1} + \sum_{k=1}^{t} \lambda^{t+1-k} \nabla_w P_k \\ &= \nabla_w P_{t+1} + \lambda\, s_t \end{aligned}$$

خانواده‌ی روش‌های یادگیری TD(λ) (ادامه)

- به ازای $\lambda < ۱$، خانواده‌ی روش‌های تفاضل‌های زمانی، بردار وزن‌ها را به شکلی کاملاً متمایز با تمامی روش‌های یادگیری با نظارت تغییر می‌دهد
- به ازای $\lambda = ۰$، این مطلب مشهودتر است. در TD(0)، میزان افزایش در بردار وزن‌ها متناسب است با تأثیر آن در آخرین پیش‌بینی (با فرض $\gamma = ۱$)

روش بروز رسانی وزن‌ها TD(0)

$$\Delta \mathbf{w}_t = \alpha(P_{t+۱} - P_t)\nabla_w P_t \qquad (۶)$$

الگوریتم TD(λ)

ورودی: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, z$ ▷ دنباله‌ی مشاهدات–نتیجه

خروجی: $\mathbf{w}$ ▷ بردار وزن‌ها

$\mathbf{w} \leftarrow \mathbf{w}_0$ ▷ بردار وزن‌ها را با مقادیر دلخواه مقداردهی اوّلیه کن

$P_t \leftarrow P(\mathbf{w}, \mathbf{x}_1)$ ▷ $\mathbf{x}_1$ مشاهده شد

$\mathbf{S} \leftarrow \nabla_w P_t$ ▷ $\mathbf{S}$ همان جمع وزن‌دار گرادیان‌هاست

for all $\mathbf{x}_t, t = 2, \ldots, m$ **do** ▷ برای مشاهدات $\mathbf{x}_2$ تا $\mathbf{x}_m$

  $P_{t-1} \leftarrow P_t$

  $P_t \leftarrow P(\mathbf{w}, \mathbf{x}_t)$

  $\Delta\mathbf{w}_{t-1} \leftarrow \alpha(P_t - P_{t-1})\mathbf{S}$ ▷ میزان افزایش بردار وزن‌ها برای مشاهده‌ی قبلی

  $\mathbf{S} \leftarrow \nabla_w P_t + \lambda\mathbf{S}$

  $\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{w}_{t-1}$ ▷ بروزرسانی بردار وزن‌ها

end for

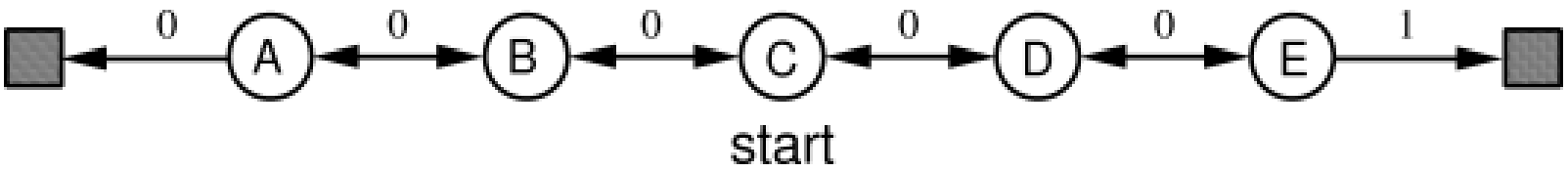
$\Delta\mathbf{w}_{t-1} \leftarrow \alpha(z - P_t)\mathbf{S}$

$\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{w}_{t-1}$ ▷ بروزرسانی نهایی بردار وزن‌ها

مثال قدم‌برداشتن تصادفی (Random Walk)

- تمامی اپیزودها از خانه‌ی مرکز، C، شروع می‌شوند. با احتمال‌های برابر حرکت به سمت راست، یا چپ انجام می‌شود. با رسیدن به هریک از خانه‌هایی که با مربّع نمایش داده شده‌اند اپیزود پایان می‌یابد.
- هدف یافتن احتمال این‌که از هرخانه به مربع انتهایی سمت راست برسیم
- این احتمال برای مربّع سمت راست مساوی ۱ ، و برای مربّع سمت چپ مساوی صفر است.
- دو نمونه از دنباله‌ی مشاهده–نتیجه: CDE1 و CDCBA0

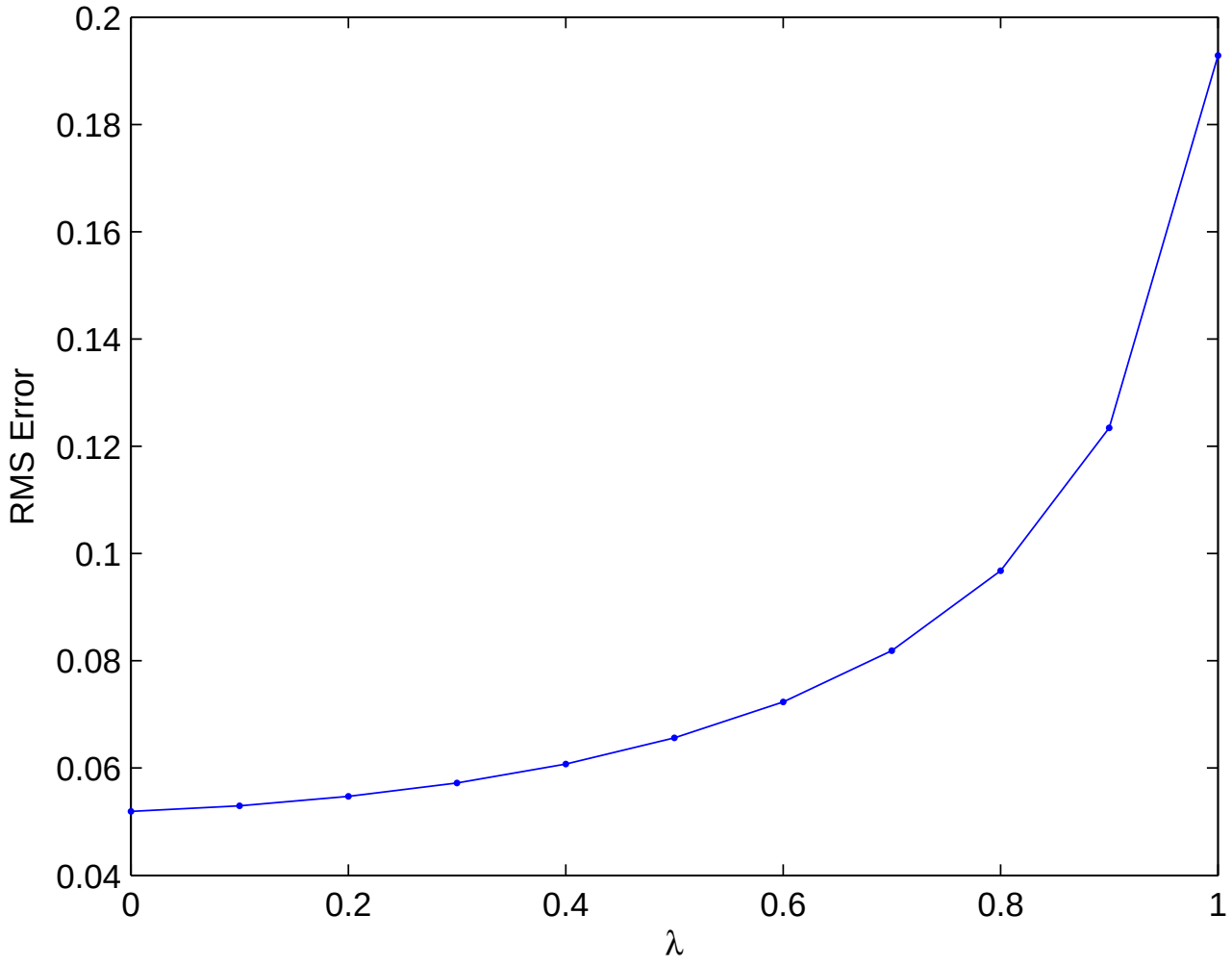
مثال قدم‌برداشتن تصادفی (Random Walk) (ادامه)



- برای پیاده‌سازی از $\mathrm{TD}(\lambda)$ خطّی استفاده شده؛ یعنی
  - $P_t = \mathbf{w}^T \mathbf{x}_t$
  - $\nabla_w P_t = \mathbf{x}_t$
- به‌جای بردارهای مشاهده از بردارهای پایه‌ی $\mathbb{R}^۵$ استفاده شده؛ یعنی
  - $\mathbf{x}_C = (۰, ۰, ۱, ۰, ۰)^T$
  - $\mathbf{x}_E = (۰, ۰, ۰, ۰, ۱)^T$

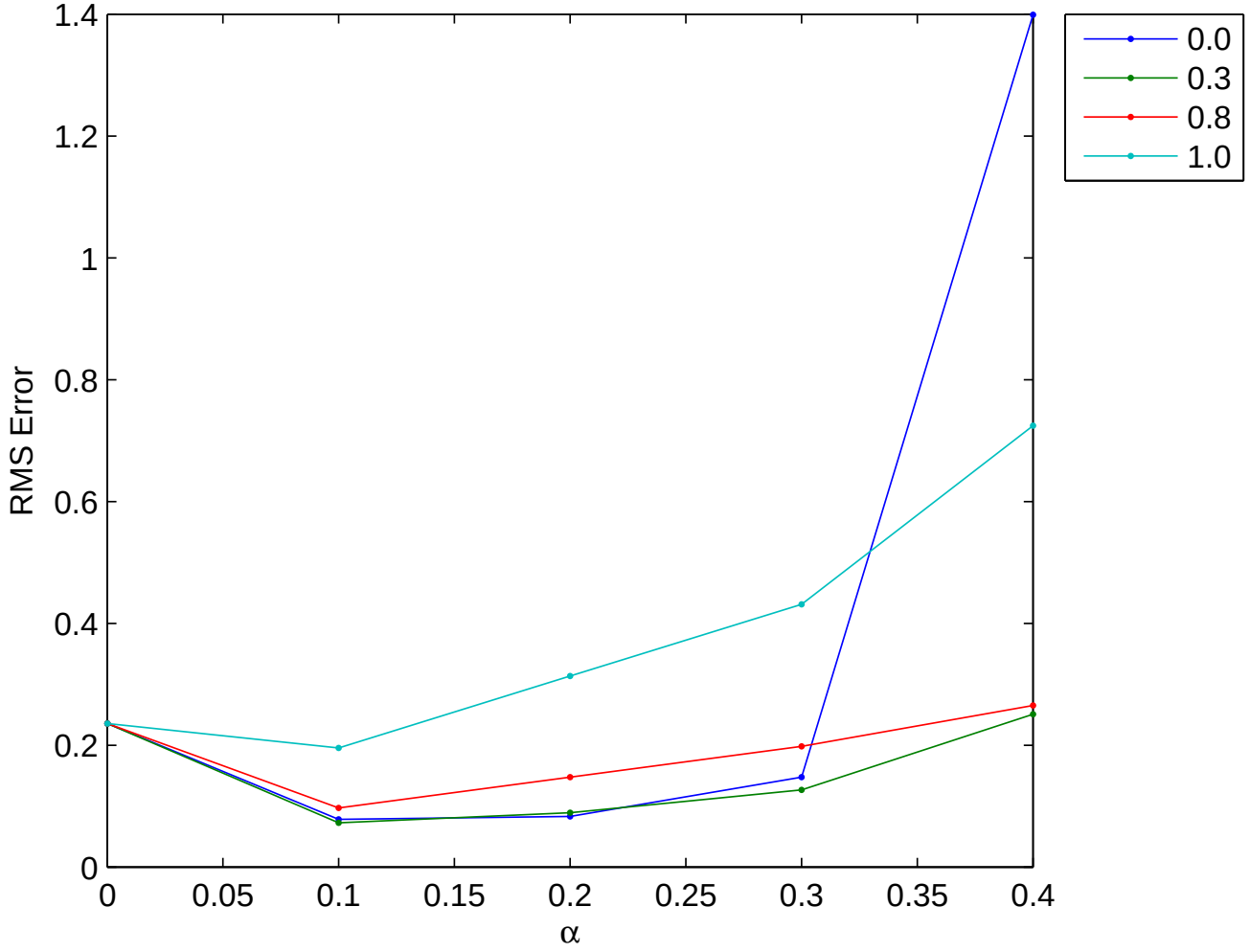
تأثیر λ بر دقّت یادگیری

PredictionRandomWalk.m را اجرا کنید

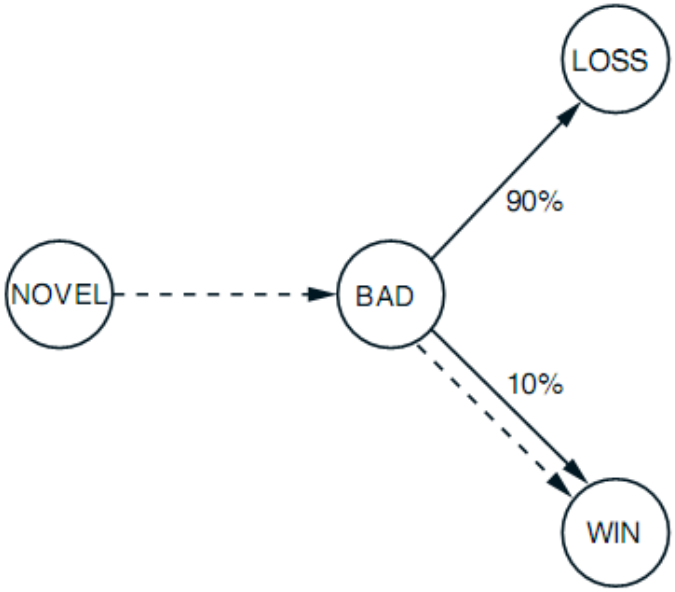


تأثیر α و λ بر دقّت یادگیری

PredictionRandomWalkAlphaEffect.m را اجرا کنید



مثال – محیط‌های مارکوف



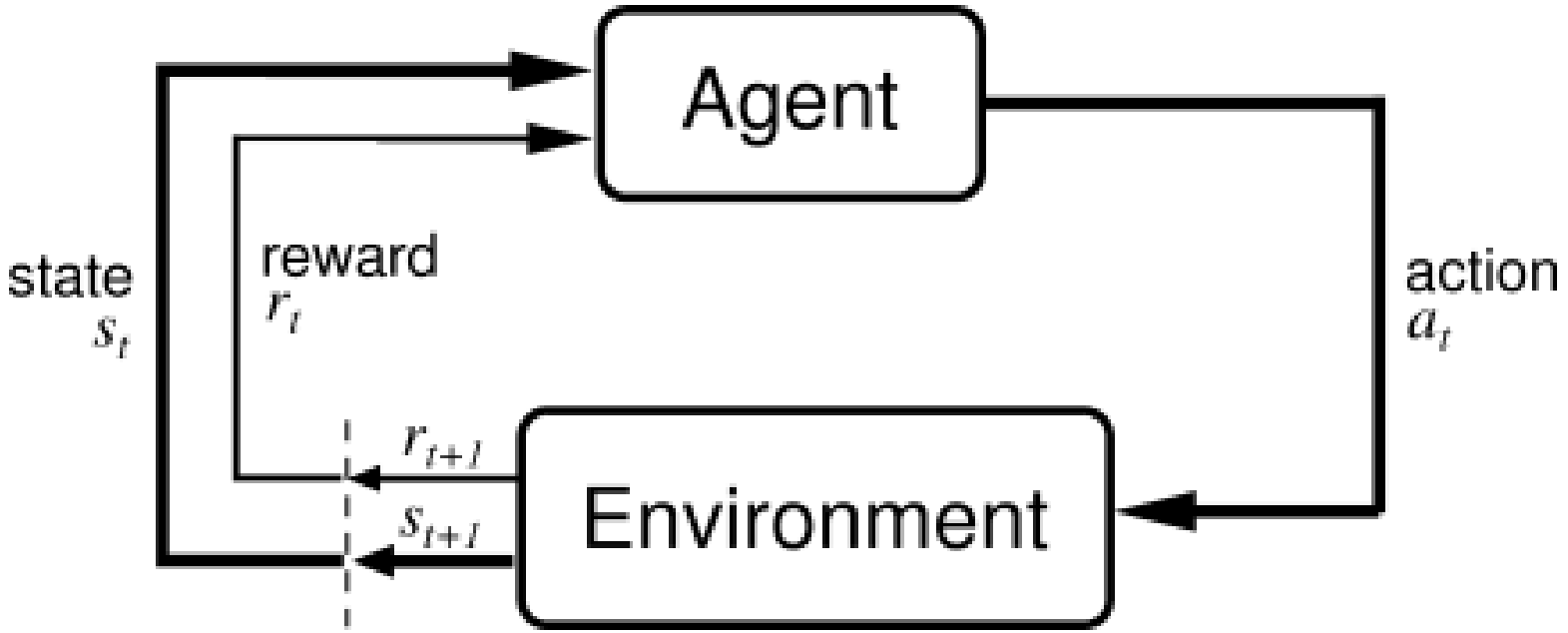
- مزیت روش‌های تفاضل‌های زمانی نسبت به یادگیری با نظارت تنها در سرعت نیست
- در برخی موقعیت‌ها روش‌های تفاضل زمانی جواب صحیح‌تری نسبت به روش‌های یادگیری با نظارت ارائه می‌دهند
- مثلاً در محیط‌های مارکوف

مقدّمه‌ای بر یادگیری تقویتی

- زیر شاخه‌ای از یادگیری ماشین، آغاز در نظریه‌ی کنترل
- یادگیری از طریق تعامل با محیط
- اجزای تشکیل دهنده‌ی مسئله‌ی یادگیری تقویتی:
  - عامل
  - محیط
  - کنش
  - پاداش (سیگنال تقویتی)
- هدف عامل: در هر وضعیت، طوری رفتار کند که مجموع پاداش‌های مورد انتظار دریافتی از محیط، در *دراز مدّت* بیشینه شود

مقدّمه‌ای بر یادگیری تقویتی (ادامه)

تعامل عامل با محیط در یادگیری تقویتی



مدل بیشینه‌کردن پاداش در دراز مدّت

مدل افق نامحدود تخفیف یافته

هدف بیشنه کردن امید ریاضی زیر است:

$$E(\sum_{t=0}^{\infty} \gamma^t r_t) \quad \text{و} \quad 0 \leq \gamma < 1$$

پاداش‌های آینده، با ضریب تخفیف γ به طور هندسی تخفیف داده می‌شوند

- ضریب γ موجب می‌شود که پاداش آنی، وزن بیشتری نسبت به پاداش‌های مورد انتظار در آینده داشته باشد، و در عین حال پاداش‌های آینده نیز نادیده گرفته نشوند

مدل بیشینه‌کردن پاداش در دراز مدّت (ادامه)

- هرچه γ به ۱ نزدیک‌تر باشد، الگوریتم یادگیری آینده‌نگرتر می‌شود و هر چه γ به صفر نزدیک‌تر باشد الگوریتم یادگیری حریص‌تر می‌شود
- کاربرد دیگر ضریب γ، کران‌دار کردن سری فوق است (کاربرد در اثبات قضایای همگرایی)
- انتخاب $\gamma = ۱$، غیرمجاز است مگر در سناریوهای اپیزودیک با تعداد مراحل محدود

استفاده از روش تفاضل‌های زمانی در یادگیری تقویتی

- یک عامل در هر لحظه‌ی t، زوج مرتب $(\mathbf{x}_t, r_t)$ را دریافت می‌کند
- $\mathbf{x}_t$ بردار مشاهدات عامل در لحظه‌ی t است
- r_t یک عدد حقیقی معرّف پاداش عامل در لحظه‌ی t
- بردار مشاهدات، $\mathbf{x}_t$ می‌تواند صرفاً حالت محیط باشد، یا علاوه بر آن شامل کنش عامل در لحظه‌ی t نیز باشد:

$$\mathbf{x}_t = \mathbf{s}_t \quad \text{یا} \quad \mathbf{x}_t = \langle \mathbf{s}_t, a_t \rangle$$

- هدف: با استفاده از روش تفاضل‌های زمانی، با دریافت دنباله‌های $(\mathbf{x}_t, r_t)$، برای $t = ۰, ۱, \ldots$، در هر لحظه‌ی t، پیش‌بینی P_t از کمیت زیر را انجام دهیم:

$$R_t = r_{t+۱} + \gamma r_{t+۲} + \gamma^۲ r_{t+۳} + \ldots = \sum_{i=۱}^{\infty} \gamma^{i-۱} r_{t+i}$$

استفاده از روش تفاضل‌های زمانی در یادگیری تقویتی

- یک عامل در هر لحظه‌ی t، زوج مرتب $(\mathbf{x}_t, r_t)$ را دریافت می‌کند
- $\mathbf{x}_t$ بردار مشاهدات عامل در لحظه‌ی t است
- r_t یک عدد حقیقی معرّف پاداش عامل در لحظه‌ی t
- بردار مشاهدات، $\mathbf{x}_t$ می‌تواند صرفاً حالت محیط باشد، یا علاوه بر آن شامل کنش عامل در لحظه‌ی t نیز باشد:

$$\mathbf{x}_t = \mathbf{s}_t \quad \text{یا} \quad \mathbf{x}_t = \langle \mathbf{s}_t, a_t \rangle$$

- هدف: با استفاده از روش تفاضل‌های زمانی، با دریافت دنباله‌های $(\mathbf{x}_t, r_t)$، برای $t = ۰, ۱, \ldots$، در هر لحظه‌ی t، پیش‌بینی P_t از کمیت زیر را انجام دهیم:

$$R_t = r_{t+۱} + \gamma r_{t+۲} + \gamma^۲ r_{t+۳} + \ldots = \sum_{i=۱}^{\infty} \gamma^{i-۱} r_{t+i}$$

استفاده از روش تفاضل‌های زمانی در یادگیری تقویتی (ادامه)

- در روش تفاضل‌های زمانی، مقادیر بردار وزن‌ها یادگرفته می‌شوند.
- در صورت استفاده از روش تفاضل‌های زمانی خطّی، طول بردار وزن‌ها برابر است با طول بردار مشاهدات.
- در یادگیری تقویتی در صورتی که بردار مشاهدات صرفاً شامل وضعیت محیط باشد، به بردار وزن‌ها می‌گویند *تابع ارزش* و آن را با V نمایش می‌دهند
- در صورتی که بردار مشاهدات علاوه بر وضعیت محیط، شامل کنش نیز باشد، آن را با Q نمایش می‌دهند

استفاده از روش تفاضل‌های زمانی در یادگیری تقویتی (ادامه)

- در هر لحظه محیط دارای وضعیت s است که s یک عضو از فضای حالات $\mathcal{S}$ می‌باشد
- با فرض گسسته و محدود بودن فضای حالات، به‌ازای هر حالت در فضای حالات، یک درایه در بردار وزن‌ها در نظر بگیریم. در این صورت بدست آوردن ارزش هر حالت به سادگی جستجو در یک جدول خواهد بود.

$$|V| = n(\mathcal{S})$$

- بردار مشاهده را طوری در نظر می‌گیریم که فقط مشخّص کند، کدام حالت در فضای حالات مشاهده شده است، بنابراین هر بردار مشاهده، یک پایه برای فضای $\mathbb{R}^{n(\mathcal{S})}$ است

- با استفاده از TD(0) خطّی:

$$\Delta V_t = \alpha(P_{t+1} - P_t)\nabla_V P_t$$
$$P_t = V^T \mathbf{x}_t \Rightarrow \nabla_V P_t = \mathbf{x}_t$$

- اگر وضعیت محیط در لحظه‌ی t، همان وضعیت شماره‌ی sام باشد در فضای حالت $\mathcal{S}$، تنها درایه‌ی sام بردار $\mathbf{x}_t$، مساوی یک است. بنابراین:

$$V_{t+1}(s) \leftarrow V_t(s) + \alpha(P_{t+1} - P_t)$$

- که در آن منظور از $V(s)$، درایه‌ی sام بردار V است؛ و منظور از V_t، نسخه‌ای از بردار V است که در زمان t، در دسترس بود.

- اگر $\gamma = ۰$؛ آن‌گاه $P_{t+۱} = r_{t+۱}$، و همچنین داریم:

$$P_t = P_t(\mathbf{x}_t) = V^T \mathbf{x}_t = V(s)$$

- تابع پیش‌بینی همان بردار ارزش‌ها است، و با یادگیری آن به طور خودبه‌خود تابع پیش‌بینی را نیز یاد گرفته‌ایم

$$V_{t+۱}(s) \leftarrow V_t(s) + \alpha(r_{t+۱} - V_t(s)) \qquad (۷)$$

- s وضعیت محیط در لحظه‌ی t است
- این الگوریتم در هر لحظه‌ی t، مقادیر بردار ارزش‌ها در گام زمانی قبلی را بروز می‌کند

- اگر $\gamma \neq ۰$؛ آن‌گاه

$$\begin{aligned} P_{t+۱} = R_t &= r_{t+۱} + \gamma r_{t+۲} + \gamma^۲ r_{t+۳} + \dots \\ &= r_{t+۱} + \gamma \left(r_{t+۲} + \gamma r_{t+۳} + \dots \right) \\ &= r_{t+۱} + \gamma R_{t+۱} \end{aligned}$$

- اگر وضعیت بعد از s را با s' نشان دهیم، آن‌گاه می‌دانیم که $V_t(s')$ تخمینی برای $R_{t+۱}$ است در لحظه‌ی t
- بنابراین می‌توانیم R_t را با $r_{t+۱} + \gamma V_t(s')$ تخمین بزنیم

$$V_{t+۱}(s) \leftarrow V_t(s) + \alpha(r_{t+۱} + \gamma V_t(s') - V_t(s)) \qquad (۸)$$

آشنایی با چند تعریف

سیاست عامل

نگاشتی که مشخّص می‌کند، عامل در هر وضعیت، ممکن است چه کنشی را انجام دهد را سیاست[1] عامل می‌گویند، و آن را با π نشان می‌دهند

[1] Policy

سیاست اپسیلون-حریصانه

سیاستی که در آن عامل به احتمال ϵ کنش تصادفی را انجام می‌دهد، و در باقی مواقع کنشی را انجام می‌دهد که در تابع ارزش بیشترین ارزش را بخود اختصاص داده.
هرچه اپسیلون بیشتر باشد عامل زمان بیشتری را صرف تجربه کردن محیط ناشناخته اطراف خود می‌کند.[2]

[2] ϵ-greedy

آشنایی با چند تعریف

سیاست عامل

نگاشتی که مشخّص می‌کند، عامل در هر وضعیت، ممکن است چه کنشی را انجام دهد را سیاست[1] عامل می‌گویند، و آن را با π نشان می‌دهند

[1] Policy

سیاست اپسیلون-حریصانه

سیاستی که در آن عامل به احتمال ϵ کنش تصادفی را انجام می‌دهد، و در باقی مواقع کنشی را انجام می‌دهد که در تابع ارزش بیشترین ارزش را بخود اختصاص داده.
هرچه اپسیلون بیشتر باشد عامل زمان بیشتری را صرف تجربه کردن محیط ناشناخته اطراف خود می‌کند.[2]

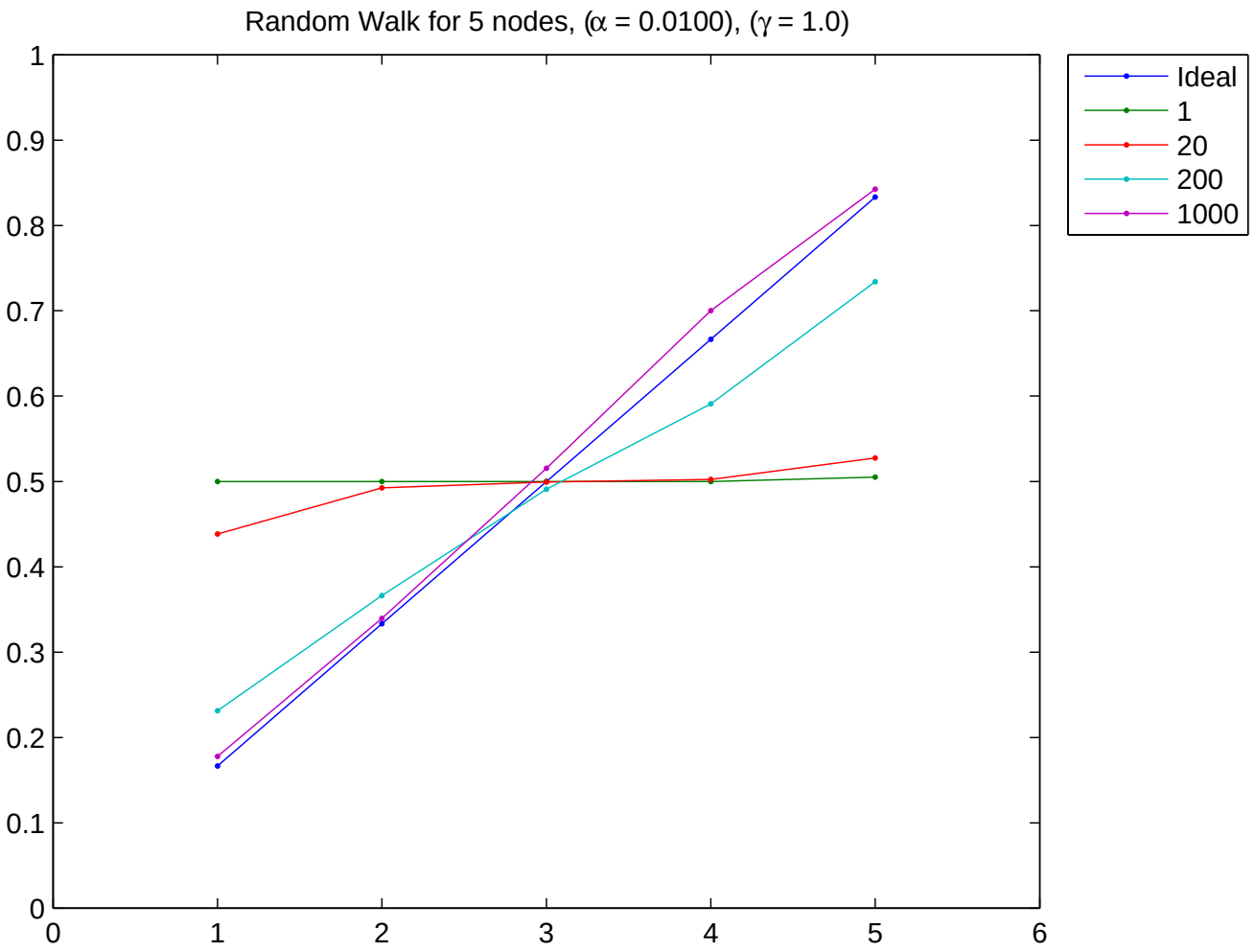
[2] ϵ-greedy

استفاده از روش TD(0) خطّی در یادگیری تقویتی

```
Initialize V arbitrarily                    ▷ بردار ارزش‌ها را به‌طور دلخواه مقداردهی اوّلیه کن.
for all episode do                          ▷ برای هر اپیزود
    Initialize s
    for all step of episode do
        a ← action given by π for s
        Take action a;                      ▷ کنش بدست آمده از طریق π را انجام بده
        observe r, and next state s′        ▷ پاداش و وضعیت بعدی محیط را مشاهده کن
        V(s) ← V(s) + α [r + γ V(s′) − V(s)]
        s ← s′
    end for
end for
```

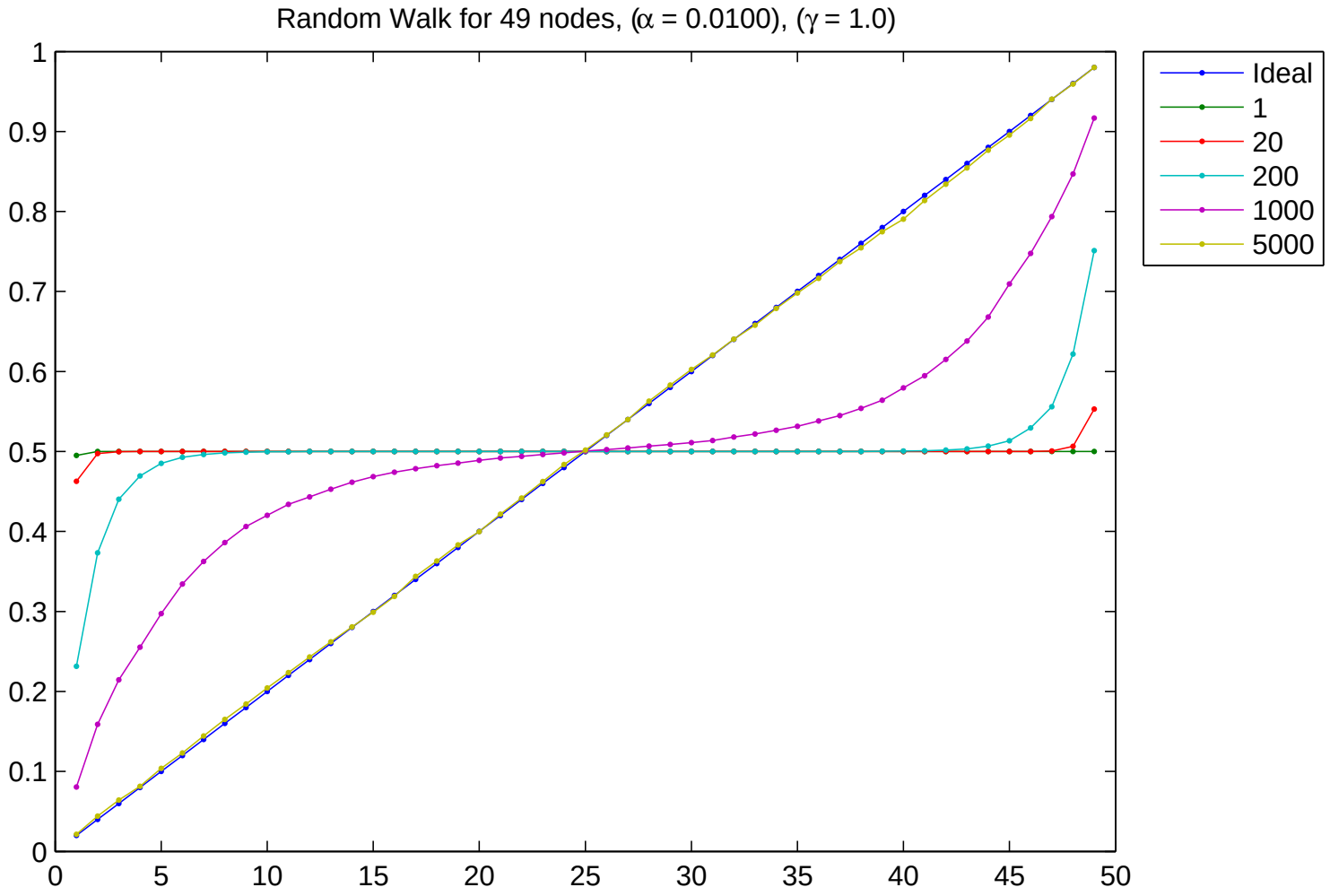
مثال: قدم برداشتن تصادفی Random Walk

RLRandomWalk.m را اجرا کنید



مثال: قدم برداشتن تصادفی Random Walk

RLRandomWalk.m را اجرا کنید



الگوریتم‌های یادگیری مبتنی بر سیاست، و مستقل از سیاست

الگوریتم‌های مبتنی بر سیاست

الگوریتم‌هایی که بهترین پاسخ را برای سیاستی که عامل در پیش گرفته ارائه می‌دهند. اگر عامل سیاست خود را تغییر دهد، پاسخ الگوریتم نیز ممکن است تغییر کند.[۳] مانند الگوریتم یادگیری SARSA.

---

on-policy

الگوریتم‌های مستقل از سیاست

الگوریتم‌هایی که بهترین پاسخ ممکن را برای محیط ارائه می‌دهند. اگر عامل سیاست خود را تغییر دهد، پاسخ الگوریتم در نهایت تغییر نمی‌کند.[۴] مانند الگوریتم یادگیری Q-Learning

---

off-policy

الگوریتم‌های یادگیری مبتنی بر سیاست، و مستقل از سیاست

الگوریتم‌های مبتنی بر سیاست

الگوریتم‌هایی که بهترین پاسخ را برای سیاستی که عامل در پیش گرفته ارائه می‌دهند. اگر عامل سیاست خود را تغییر دهد، پاسخ الگوریتم نیز ممکن است تغییر کند.[۳] مانند الگوریتم یادگیری SARSA.

---

on-policy

الگوریتم‌های مستقل از سیاست

الگوریتم‌هایی که بهترین پاسخ ممکن را برای محیط ارائه می‌دهند. اگر عامل سیاست خود را تغییر دهد، پاسخ الگوریتم در نهایت تغییر نمی‌کند.[۴] مانند الگوریتم یادگیری Q-Learning

---

off-policy

الگوریتم یادگیری مبتنی بر سیاست SARSA

Initialize $Q(s, a)$ arbitrarily
for all episode **do**
    Initialize s
    Choose a from s using policy derived from Q (e.g., ϵ-greedy)
    for all step of episode **do**
        Take action a, observe r, s'
        Choose a' from s' using policy derived from Q (e.g., ϵ-greedy)
        $Q(s, a) \leftarrow Q(s, a) + \alpha\,[r + \gamma Q(s', a') - Q(s, a)]$
        $s \leftarrow s'$; $a \leftarrow a'$
    end for
end for

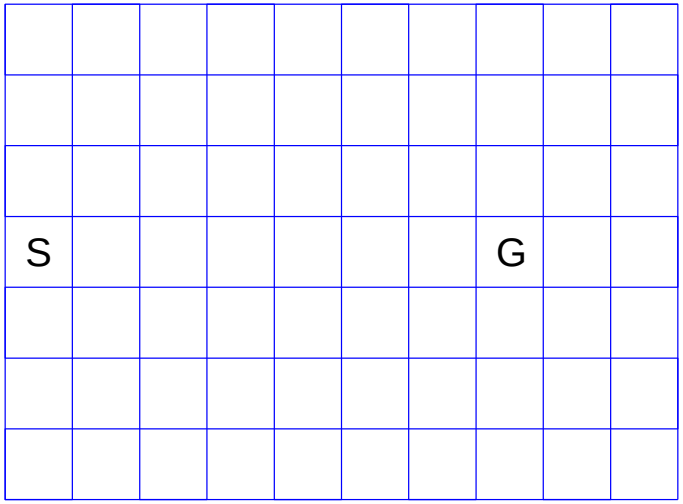
الگوریتم یادگیری مستقل از سیاست Q-Learning

```
Initialize Q(s, a) arbitrarily
for all episode do
    Initialize s
    for all step of episode do
        Choose a from s using policy derived from Q (e.g., ϵ-greedy)
        Take action a, observe r, s'
        Q(s, a) ← Q(s, a) + α [r + γ max_a' Q(s', a') − Q(s, a)]
        s ← s';
    end for
end for
```

مثال: Grid World

شمای کلّی محیط

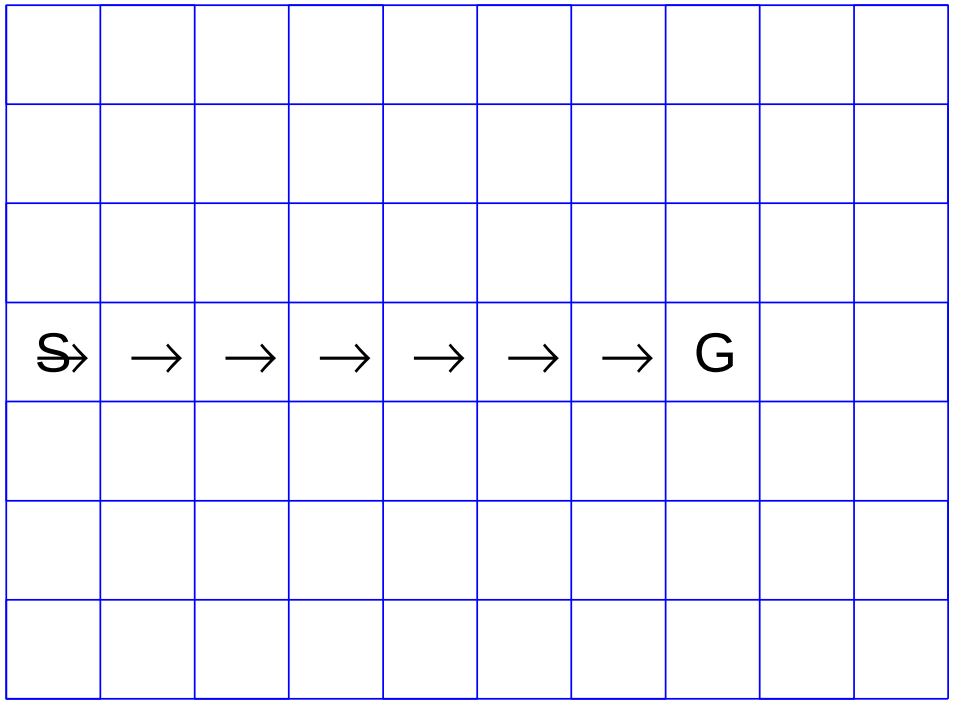
- هدف: آغاز حرکت از S و رسیدن به G
- اعمال ممکن: حرکت در چهار جهت اصلی
- الگوریتم‌های SARSA و Q-Learning یک پاسخ را ارائه می‌دهند



مثال: Grid World

GridWorldsDemo('sg-small-sarsa') را اجرا کنید

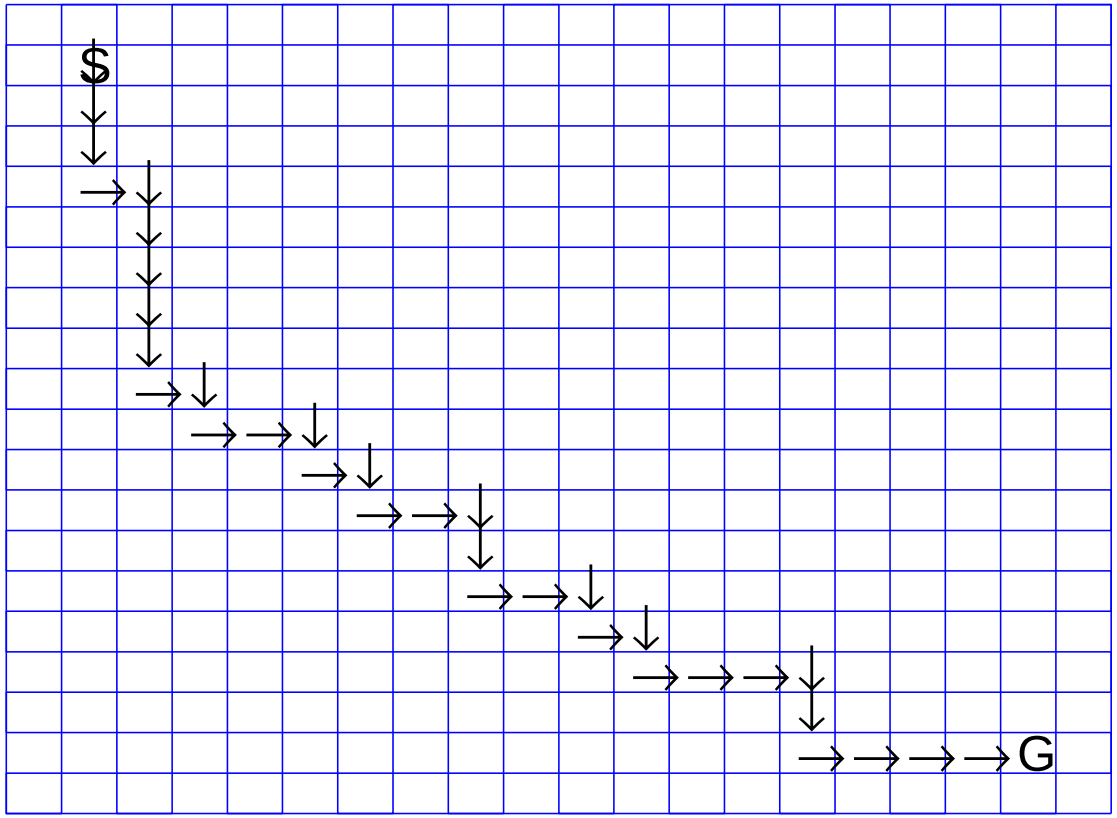
Simple grid−world SARSA − episode 700 − (ε: 0.100), (α = 0.1000), (γ = 0.9)



مثال: Grid World

GridWorldsDemo('sg-big-sarsa') را اجرا کنید

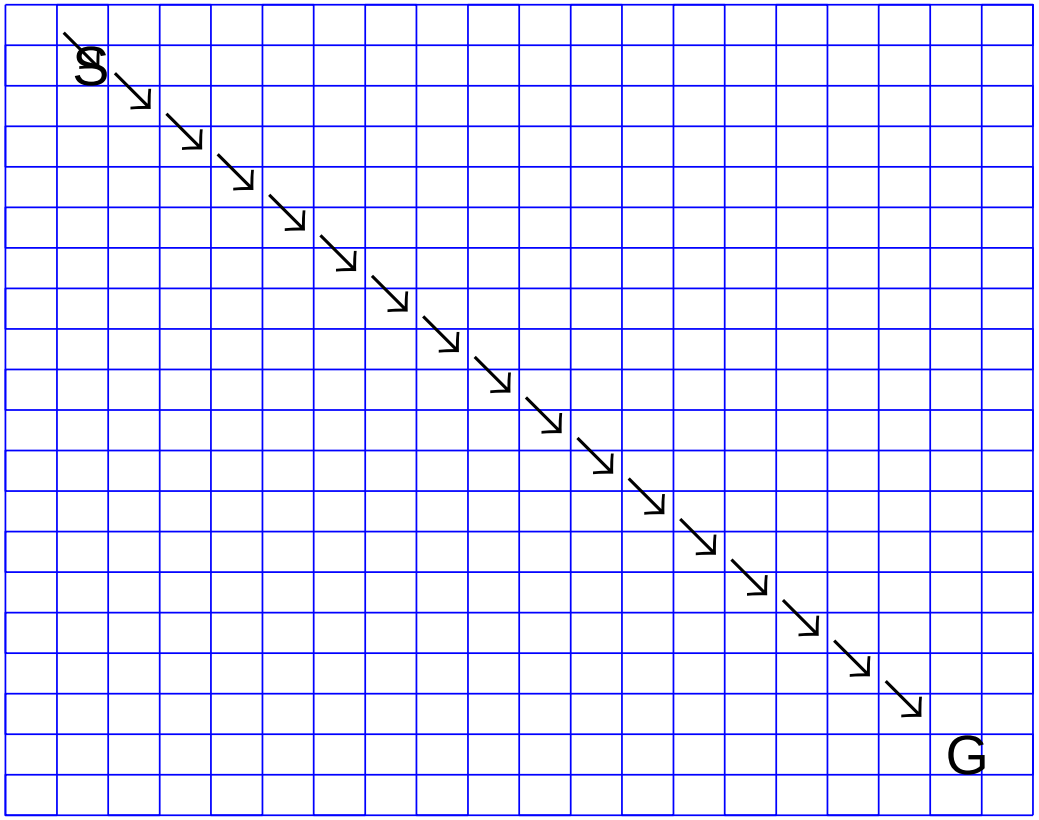
Simple grid−world SARSA − episode 4000 − (ε: 0.100), (α = 0.3000), (γ = 0.9)



مثال: Grid World

GridWorldsDemo('sg-big-king-sarsa') را اجرا کنید

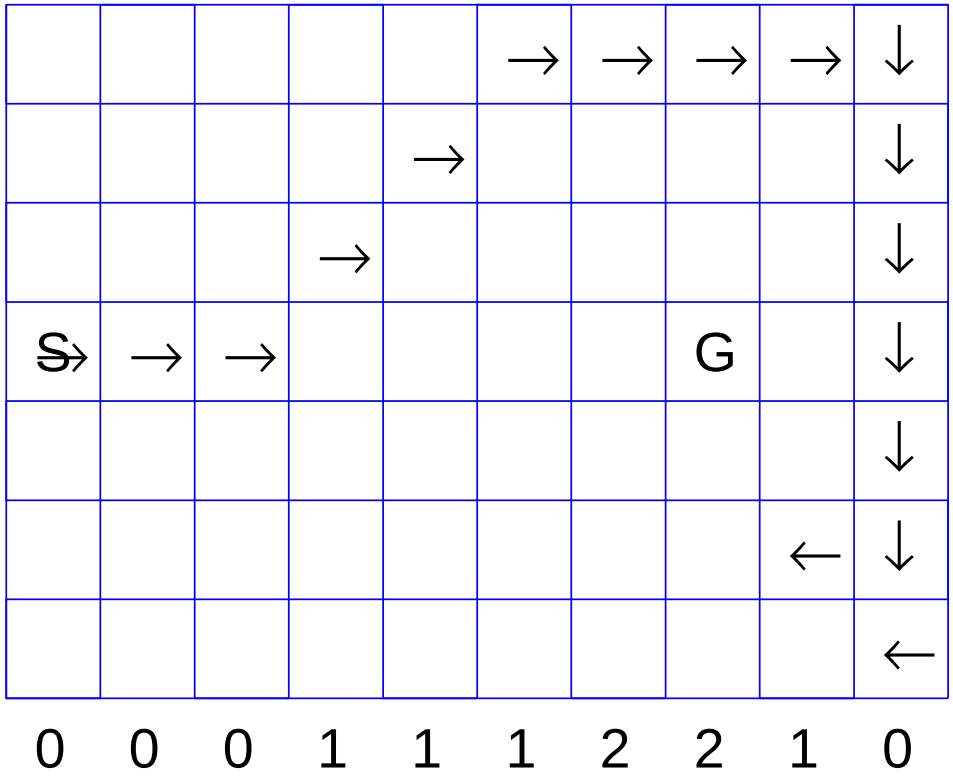
Simple grid−world SARSA − episode 4000 − (ε: 0.100), (α = 0.1000), (γ = 0.9)



مثال: Windy Grid World

GridWorldsDemo('wg-small-sarsa') را اجرا کنید

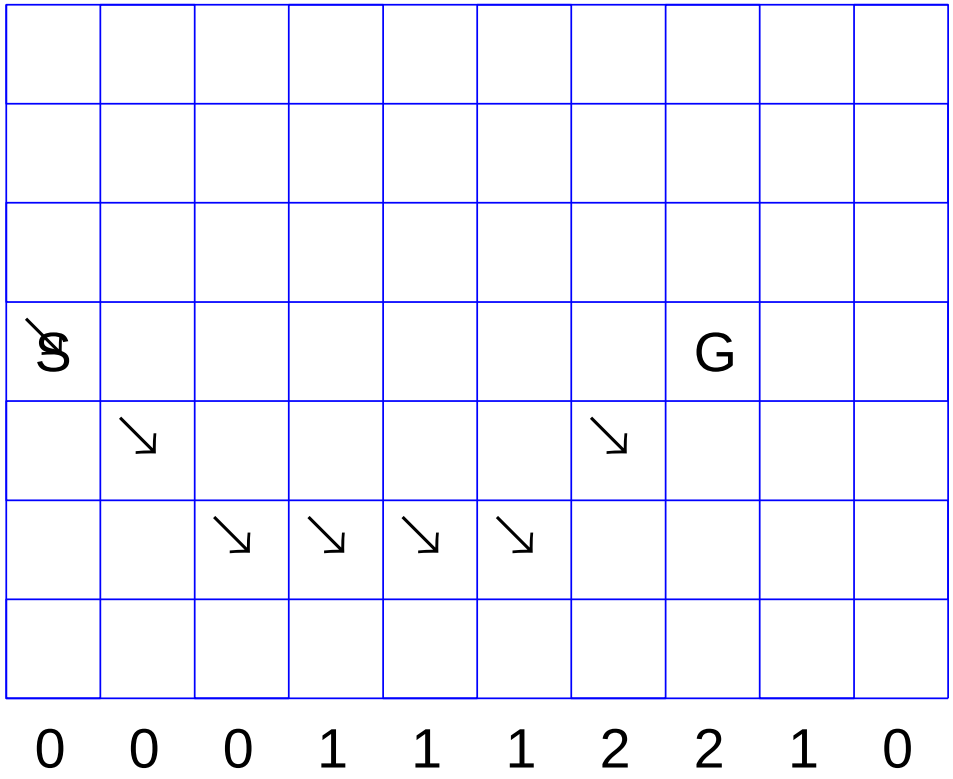
Windy grid−world SARSA − episode 1000 − (ε: 0.100), (α = 0.1000), (γ = 0.9)



مثال: Windy Grid World

GridWorldsDemo('wg-small-king-sarsa') را اجرا کنید

Windy grid−world SARSA − episode 2000 − (ε: 0.100), (α = 0.1000), (γ = 0.9)



مثال: Cliff Walking

GridWorldsDemo('cliff-small-sarsa') را اجرا کنید

Cliff Walking SARSA − episode 15000 − (ε: 0.100), (α = 0.0100), (γ = 0.9)

→	→	→	→	→	→	→	→	→	↓
↑									↓
$	C	C	C	C	C	C	C	C	G

مثال: Cliff Walking

GridWorldsDemo('cliff-small-qlearning') را اجرا کنید

Cliff Walking Q−Learning − episode 1000 − (ε: 0.100), (α = 0.1000), (γ = 0.9)

→	→	→	→	→	→	→	→	→	↓
$	C	C	C	C	C	C	C	C	G

با تشکر از شما

؟